SLEZSKÁ UNIVERZITA
FILOZOFICKO-PŘÍRODOVĚDECKÁ FAKULTA V OPAVĚ

Šárka Vavrečková

Skripta do předmětu

Architektura počítačových systémů

Technické vybavení osobních počítačů

Ústav informatiky
Filozoficko-přírodovědecká fakulta v Opavě
Slezská univerzita v Opavě

Opava
5. prosince 2024

Anotace: Tato skripta jsou určena pro studenty předmětů *Architektura počítačových systémů* a *Technické vybavení osobních počítačů* na Ústavu informatiky Slezské univerzity v Opavě. V předmětu se zabýváme především hardwarem – principem fungování komponent, diagnostikou, údržbou. Jedná se sice o přednášky, ale vzhledem k charakteru probírané látky je součástí i praktická výuka.

Architektura počítačových systémů, Technické vybavení osobních počítačů

RNDr. Šárka Vavrečková, Ph.D.

Dostupné na: http://vavreckova.zam.slu.cz/hw.html

Ústav informatiky
Filozoficko-přírodovědecká fakulta v Opavě
Slezská univerzita v Opavě
Bezručovo nám. 13, Opava

Sázeno v systému LaTeX

Předmluva

Co najdeme v těchto skriptech

Rychlý náhled: Tato skripta jsou určena pro studenty Ústavu informatiky Slezské univerzity v Opavě. Obsahují látku vyučovanou na přednáškách předmětů *Architektura počítačových systémů* a *Technické vybavení osobních počítačů* (tentýž předmět, jen změna názvu v nových akreditacích), ve kterém se zabýváme hardwarem. Zajímá nás architektura výpočetních systémů (nejen počítačů), hardwarová komunikační rozhraní, jednotlivé komponenty počítače či jiného výpočetních systému (procesor, základní deska, paměti, napájení apod.), jejich funkčnost a údržba.

Pozor, důležitou součástí výuky jsou samotné přednášky, u zkoušky si pouze s informacemi ze skript nevystačíte (kromě těch, kdo s hardwarem již nějakou dobu pracují). Na přednáškách (i v kombinované formě studia) se studenti „ručně“ seznamují s konkrétními kusy hardwaru a předpokládá se, že na zkoušce se v nich orientují. Pokud z vážného důvodu nemůžete navštěvovat přednášky, využijte možnost konzultací.

Některé oblasti jsou „navíc“ (jsou označeny ikonami fialové barvy), ty nejsou probírány a ani se neobjeví na zkoušce – jejich úkolem je motivovat k dalšímu samostatnému studiu či pokusům nebo pomáhat v budoucnu při získávání dalších informací. Pokud je fialová ikona před názvem kapitoly (sekce), platí pro vše, co se v dané kapitole či sekci nachází.

Značení

Ve skriptech se používají následující barevné ikony:

- *Rychlý náhled* (skript, kapitoly), ve kterém se dozvíme, o čem to bude.
- *Klíčová slova* kapitoly.
- *Cíle studia* pro kapitolu nám řeknou, co nového se v dané kapitole naučíme.
- Nové *pojmy*, značení apod. jsou označeny modrým symbolem, který vidíme zde vlevo. Tuto ikonu (stejně jako následující) najdeme na začátku odstavce, ve kterém je nový pojem zaváděn.
- Konkrétní *postupy* a nástroje, způsoby řešení různých situací, do kterých se může správce počítačového vybavení dostat, atd. jsou značeny také modrou ikonou.

- Některé části textu jsou označeny fialovou ikonou, což znamená, že jde o *nepovinné úseky*, které nejsou probírány (většinou; studenti si je mohou podle zájmu vyžádat nebo sami prostudovat). Jejich účelem je dobrovolné rozšíření znalostí studentů o pokročilá témata, na která obvykle při výuce nezbývá moc času.
- Žlutou ikonou jsou označeny odkazy, na kterých lze získat *další informace* o tématu. Nejčastěji u této ikony najdeme webové odkazy na stránky, kde se dané tématice jejich autoři věnují podrobněji.
- Červená je ikona pro *upozornění* a poznámky.

Pokud je množství textu patřícího k určité ikoně větší, je celý blok ohraničen prostředím s ikonami na začátku i konci, například pro definování nového pojmu:

Definice

V takovém prostředí definujeme pojem či vysvětlujeme sice relativně známý, ale komplexní pojem s více významy či vlastnostmi.

Podobně může vypadat prostředí pro delší postup nebo delší poznámku či více odkazů na další informace. Mohou být použita také jiná prostředí:

Příklad

Takto vypadá prostředí s příkladem, obvykle nějakého postupu. Příklady jsou obvykle komentovány, aby byl jasný postup jejich řešení.

Úkol

Otázky a úkoly, náměty na vyzkoušení, které se doporučuje při procvičování učiva provádět, jsou uzavřeny v tomto prostředí. Pokud je v prostředí více úkolů, jsou číslovány.

Obsah

Předmluva **iii**

1 Historie **1**
1.1 Pravěk . . . 1
1.1.1 Učíme se počítat . . . 1
1.1.2 Mechanické pomůcky . . . 2
1.1.3 První stroje . . . 2
1.2 Výpočetní technika . . . 3
1.2.1 Nultá generace . . . 3
1.2.2 První generace . . . 5
1.2.3 Druhá generace . . . 6
1.2.4 Třetí generace . . . 7
1.2.5 Čtvrtá generace . . . 7
1.3 Analogové počítače . . . 8

2 Struktura počítače a BIOS **10**
2.1 Základní typy počítačových architektur . . . 10
2.1.1 Von Neumannovo schéma . . . 10
2.1.2 Harvardská architektura . . . 12
2.2 Instrukční a datové proudy – Flynnova taxonomie . . . 12
2.3 BIOS . . . 13
2.4 Zapnutí počítače . . . 14
2.5 EFI a UEFI . . . 16
2.5.1 Zařízení . . . 17
2.5.2 Ovládání UEFI a pre-boot aplikace . . . 17
2.5.3 Secure Boot . . . 18
2.6 Upgrade firmwaru (BIOSu) . . . 19
2.7 Čas v počítači . . . 21

3 Rozhraní **22**
3.1 Pár pojmů na začátek . . . 22
3.2 USB . . . 24
3.2.1 Vlastnosti a verze USB . . . 25
3.2.2 Hardwarové rozhraní . . . 26
3.2.3 Napájení . . . 30
3.2.4 Srovnání verzí . . . 31

3.2.5 Související technologie . . . 31
3.2.6 Protokoly vyšší úrovně . . . 34
3.3 Další univerzální rozhraní . . . 34
3.3.1 FireWire . . . 34
3.3.2 Thunderbolt . . . 35
3.3.3 Srovnání univerzálních rozhraní . . . 36
3.4 Paralelní a sériové porty, PS/2 . . . 37
3.5 Datové rozhraní paměťových médií . . . 39
3.5.1 PATA a SATA . . . 39
3.5.2 SCSI a SAS . . . 42
3.6 Konektory na grafických kartách . . . 44
3.6.1 VGA . . . 44
3.6.2 DVI . . . 45
3.6.3 HDMI, DisplayPort . . . 46
3.6.4 MHL . . . 48
3.6.5 Další rozhraní pro přenos obrazu . . . 48
3.7 Síť a možnosti bezdrátového propojení . . . 49

4 Case a základní deska **52**
4.1 Skříň počítače . . . 52
4.2 Co je to základní deska . . . 54
4.2.1 PCB . . . 54
4.2.2 Form factor . . . 54
4.2.3 Interfaces . . . 56
4.3 Sběrnice . . . 57
4.3.1 ISA, EISA . . . 58
4.3.2 PCI a PCI-X . . . 58
4.3.3 PCI Express . . . 59
4.3.4 AGP . . . 60
4.4 Čipset . . . 61
4.4.1 North-South Bridge Design . . . 61
4.4.2 Čipset a sběrnice . . . 64
4.4.3 One Chip Design . . . 65
4.4.4 SoC čipy . . . 66
4.5 Přepínače, propojky . . . 67
4.6 Rozvržení komponent na základní desce . . . 68

5 Procesory **70**
5.1 Co je to procesor . . . 70
5.1.1 Technologie výroby . . . 71
5.1.2 Tranzistory . . . 72
5.1.3 Matematický koprocesor . . . 74
5.1.4 Mooreův zákon . . . 74
5.2 Struktura procesoru . . . 76
5.2.1 Logická struktura . . . 76
5.2.2 Instrukční sada . . . 78
5.2.3 Cache paměť . . . 79
5.2.4 Úrovně hardwarové ochrany prostředků . . . 81

5.2.5 Registry . . . 82
5.2.6 Registry procesorů Intel a AMD . . . 83
5.2.7 Registry procesorů ARM . . . 86
5.3 Procesory podle instrukčních sad . . . 88
5.4 Vlastnosti procesorů . . . 90
5.5 Techniky zvyšování propustnosti a výkonu procesorů . . . 91
5.5.1 Pipelining a . . . skalární architektury . . . 91
5.5.2 HyperThreading, více jader, více procesorů . . . 93
5.5.3 Řešení datových závislostí . . . 94
5.5.4 Nepodmíněné a podmíněné skoky . . . 97
5.5.5 Branch Prediction . . . 99
5.5.6 Mechanismy optimalizace superskalárních procesorů . . . 101
5.5.7 Zneužití mechanismů superskalárních architektur . . . 103
5.6 Další vlastnosti procesorů . . . 104
5.6.1 Podpora virtualizace . . . 104
5.6.2 Energetická náročnost a správa energie . . . 104
5.6.3 Modularita procesorů . . . 106
5.6.4 Little a Big Endian . . . 108
5.6.5 Vícevláknové aplikace . . . 108
5.7 Patice procesorů . . . 109
5.8 Přehled procesorů . . . 111
5.8.1 Architektura a mikroarchitektura u Intelu . . . 111
5.8.2 Typy procesorů Intel a systém Tick-Tock . . . 111
5.8.3 Přehled procesorů Intelu . . . 113
5.8.4 Intel vs. AMD . . . 114
5.8.5 Typy procesorů AMD . . . 115
5.8.6 Přehled procesorů AMD . . . 116
5.8.7 Procesory ARM . . . 117
5.9 Komunikace se zařízeními . . . 118
5.9.1 IRQ . . . 119
5.9.2 DMA . . . 122
5.9.3 I/O adresy a adresy paměti zařízení . . . 123

6 Vnitřní paměti **125**
6.1 Nejdřív obecně o pamětech . . . 125
6.2 Vlastnosti vnitřních pamětí . . . 126
6.2.1 Členění . . . 126
6.2.2 Obvyklá struktura . . . 126
6.2.3 ROM paměti . . . 127
6.2.4 RAM paměti . . . 128
6.3 Paměťové moduly . . . 131
6.4 Jak fungují moduly DIMM . . . 134
6.4.1 Komunikační kanály . . . 135
6.4.2 Operace a časování DIMM . . . 135
6.4.3 Přetaktování pamětí . . . 137
6.5 Testování pamětí . . . 137

7 Vnější paměti **139**

7.1 Základní pojmy . . . 139
7.2 Magnetické disky . . . 140
7.2.1 Diskové paměti . . . 140
7.2.2 Fyzická struktura a geometrie disku . . . 140
7.2.3 Čtení a zápis . . . 143
7.2.4 Technologie . . . 145
7.2.5 Největší a nejmenší . . . 147
7.2.6 Sledování pevných disků . . . 147
7.2.7 Vybíráme pevný disk . . . 149
7.3 RAID . . . 150
7.3.1 Typy RAID . . . 150
7.3.2 Nastavení RAID pole . . . 152
7.4 NAS . . . 154
7.5 Optické paměti . . . 155
7.5.1 CD . . . 155
7.5.2 DVD . . . 157
7.5.3 Struktura přepisovatelného disku . . . 159
7.5.4 Blu-Ray . . . 160
7.5.5 Životnost a údržba optických médií . . . 161
7.6 Flash paměti . . . 162
7.6.1 USB flash disk . . . 162
7.6.2 SSD . . . 163
7.6.3 Hybridní disky . . . 166
7.6.4 Paměťové karty . . . 166
7.7 Zálohování a archivace . . . 168
7.7.1 Magnetická páska . . . 168
7.7.2 Pásky momentálně používané pro zálohování . . . 169

8 Rozšiřující karty **171**
8.1 Co je to rozšiřující karta . . . 171
8.2 Grafická karta . . . 172
8.2.1 Základní typy zobrazení a obsah videopaměti . . . 172
8.2.2 Vývoj grafických karet . . . 174
8.2.3 Struktura grafické karty . . . 175
8.2.4 Napájení grafické karty . . . 177
8.3 Zvuková karta . . . 178
8.3.1 Princip . . . 178
8.3.2 Vzorkování a syntéza zvuku . . . 178
8.3.3 Rozhraní . . . 179
8.4 Síťová karta . . . 180
8.4.1 Princip . . . 180
8.4.2 Související technologie . . . 180

9 Vstupní a výstupní zařízení **182**
9.1 Vstupy a výstupy . . . 182
9.2 Ukazovací zařízení . . . 183
9.2.1 Trackball, Trackpoint, Touchpad, Tablet . . . 183
9.2.2 Dotykové obrazovky . . . 184

9.3 Barvy v grafice . . . 184
9.3.1 Počítač zpracovává barvy . . . 184
9.3.2 CIE . . . 185
9.3.3 Pojmy z grafiky . . . 187
9.4 Scannery . . . 187
9.4.1 Snímací technologie . . . 187
9.4.2 Optické rozpoznávání znaků . . . 188
9.5 Monitory . . . 189
9.5.1 Parametry monitoru . . . 189
9.5.2 CRT monitory . . . 190
9.5.3 LCD panely . . . 191
9.5.4 Druhy TFT panelů . . . 193
9.5.5 OLED technologie . . . 196
9.6 Tisk . . . 196

10 Napájení a chlazení **200**
10.1 Napájení . . . 200
10.1.1 Jak zdroj funguje . . . 200
10.1.2 Standardy . . . 201
10.1.3 Vlastnosti zdroje . . . 202
10.1.4 Akumulátor . . . 203
10.2 Chlazení . . . 205
10.2.1 Princip chlazení počítače . . . 205
10.2.2 Typy chladičů . . . 206
10.2.3 Sledování teploty . . . 207
10.3 Power Management . . . 207

Literatura **209**

Kapitola 1

Historie

Rychlý náhled: Tato kapitola je věnována historii výpočetní techniky, v celé škále od pravěkých mechanických pomůcek přes první počítací stroje, jednotlivé generace výpočetní techniky až po novověk. Poslední sekce této kapitoly pojednává o rozdílu mezi digitálními a analogovými počítači.

Klíčová slova: Abakus, logaritmus, Antikythérský mechanismus, Pascalina, děrný štítek, generace, Charles Babbage, Diferenciální stroj, Analytický stroj, Augusta Ada Byron, elektromagnetické relé, MARK I, SAPO, elektronka, Colossus, Epos, polovodičový efekt, tranzistor, PDP-1, MSP, integrovaný obvod, IBM System 360, Intel 4004, analogový počítač, digitální počítač.

Cíle studia: Po prostudování této kapitoly získáte základní přehled ve vývoji výpočetní techniky a stavebních kamenech, na kterých stojí i současná výpočetní technika.

1.1 Pravěk

1.1.1 Učíme se počítat

Základem výpočetní techniky jsou samotné výpočty. Lidé začali počítat takto:

- desítková soustava – prsty na rukou,
- dvanáctková soustava – používaly se všechny prsty kromě palce, počítaly se klouby na prstech (tři klouby na každém ze čtyř prstů = 12), na druhé ruce počítali tucty, známé například ze starého Sumeru, podobný systém používali také Egypťané,
- 3000 př. n. l. – v Číně objevena dvojková soustava.

V Řecku vznikl systém kombinací pozic prstů, každá kombinace znamenala určitý počet. Počítali v pětkové soustavě.

Do dřevěné hůlky či kosti se dělaly zářezy, které mohly být sdružené do skupin.

- Před 37 tisíci lety v jižní Africe někdo do kosti paviána vyryl 29 pravidelných značek, zaznamenal si množství, které nedokázal slovně pojmenovat.
- Před 30 tisíci lety v oblasti Dolních Věstonic někdo do vlčí kosti vyryl dvě dlouhé podélné čáry a od nich na obě strany hodně menších zářezů – na jedné straně 25, na druhé 30. Kost pravděpodobně sloužila jako pomůcka při výměnném obchodu.

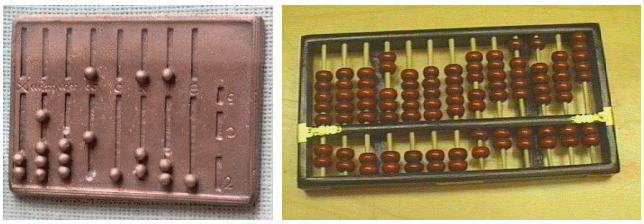
- Ještě nedávno se v horských oblastech světa používala tzv. vrubovka, tj. kost s vroubky, na které se zaznamenávaly počty čehokoliv.

Používaly se také desky kamenné, později dřevěné, pokryté pískem nebo prachem.

1.1.2 Mechanické pomůcky

Mechanická počitadla se používala v různých částech světa. Nejznámější jsou

- abakus (Evropa),
- suan-pchan (Čína, 500 př. n. l.),
- sčot (Rusko), používá se dodnes,
- saroban (Japonsko),
- současné jednoduché počitadlo.



Obrázek 1.1: Římský abakus a japonský saroban[1]

První zmínky o podobných počitadlech jsou staré 3000 let z Číny a Indie.

✎ Logaritmy a logaritmická pravítka znamenaly velké zrychlení výpočtů. Co je logaritmus?

- použitím logaritmu se násobení převádí na sčítání a odčítání,
- zjednodušily násobení a dělení velmi velkých čísel a čísel s dlouhým desetinným rozvojem.

John Napier objevil počítání s logaritmy, *William Oughtred* vynalezl logaritmické pravítko (1622).

1.1.3 První stroje

✎ *Antikythérský mechanismus* je 2000 let starý početní stroj, který byl nalezen v potopené starořecké lodi. Mechanika je řešena ozubenými kolečky, je vyroben z bronzu. Není jisté, k jakému účelu se používal, pravděpodobně k astronomickým výpočtům.

Obrázek 1.2: Antikythérský mechanismus a Pascalina[2]

✎ *Mechanické kalkulačky* byly ručně poháněné. Z nejznámějších:

- *Pascalina* (1642) – *Blaise Pascal* sestrojil funkční mechanickou kalkulačku z ozubených koleček pro sčítání a odčítání (pro svého otce, daňového úředníka),
- *Kroková kalkulačka* – *Gottfried Wilhelm Leibnitz* sestrojil mechanickou kalkulačku založenou na dvojkové soustavě, uměla navíc násobit, dělit a počítat druhou mocninu.

[1] Zdroje: http://abakus.navajo.cz/, http://www.fi.muni.cz/usr/jkucera/pv109/xdavidov.html.
[2] Zdroje: http://wbdno.wordpress.com/2010/02/06/the-antikythera-mechanism-research-project/, http://michele.nireblog.com/file/237571.

Později vznikaly kalkulačky na elektřinu, a to elektromechanické (ozubená kolečka poháněná elektřinou) nebo elektronické (zde už se používaly elektronické obvody, existovaly dokonce i programovatelné modely).

Programovatelné přístroje jsou další generací výpočetních strojů. Ovšem programovatelnost byla jen na omezené úrovni, vpodstatě šlo o možnost práce s variabilními daty uloženými na něčem, co se v principu podobalo pozdějším děrným štítkům. Z nejznámějších:

- *Joseph Marie Jacquard* sestrojil tkalcovský stav s tkaným vzorem uloženým na děrném štítku (je po něm pojmenován vzor „žakár“ (jacquard),
- *Herman Hollerith* – stroj na zpracování údajů při sčítání obyvatelstva, taktéž pomocí děrných štítků.

1.2 Výpočetní technika

V tabulce 1.1 je stručný přehled vývoje výpočetní techniky.

Generace	Roky	Součástky
0	19.–20. století 40. léta	Ozubená kola Elektromagnetická relé
1	50. léta	Elektronky
2	50.–60. léta	Tranzistory
3	konec 60. let	Integrované obvody
$3\frac{1}{2}$	1972 (70. léta)	Integrované obvody (vysoká integrace)
4	od roku 1981 do současnosti	Integrované obvody (velmi vysoká integrace)

Tabulka 1.1: Souhrn historie výpočetní techniky

1.2.1 Nultá generace

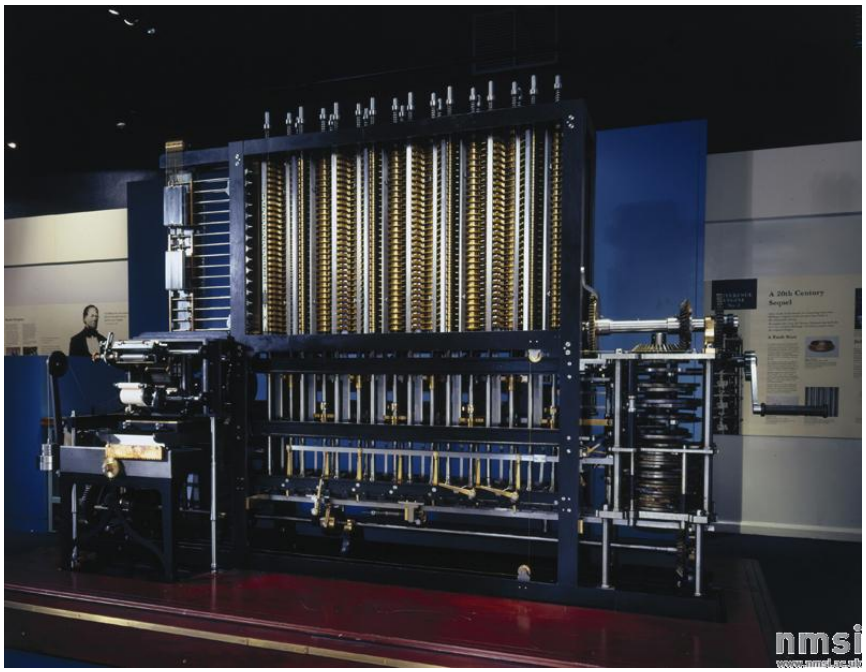
Charles Babbage, britský matematik, byl jedním z prvních geniálních vynálezců v oblasti výpočetní techniky. Jeho životním cílem bylo automatizovat výpočetní postupy tak, aby se ušetřil čas a především eliminovaly chyby (kterým se lidé při ručním počítání často nevyhnou).

Za svého života stihl sestrojit a zprovoznit několik strojů poháněných párou, čímž dokázal, že tento cíl je splnitelný. Nicméně jeho nejzajímavější stroje bohužel zůstaly jen „na papíře“ – nemohl je sestrojit z finančních důvodů, ale také proto, že vyžadovaly velmi jemnou mechanickou práci, která v té době ani zdaleka nebyla běžná.

Diferenciální stroj (také Diferenční, Difference Engine, 1822) je právě jedním z těchto strojů, které Babbage nestihl reálně dokončit (stihl jen některé komponenty). Jeho účelem mělo být počítání polynomických funkcí až 10. stupně, přičemž si Babbage vystačil s ozubenými koly – šlo o čistě mechanický stroj poháněný klikou, a počítání bylo realizováno důmyslně pouze s použitím operace sčítání.

Diferenciální stroj byl sestrojen a plně zprovozněn až nedávno, dokonce několikrát. V roce 1991 byl podle Babbageových plánů dokončen tzv. Difference Engine No 2, který je dnes součástí sbírek v Science Museum v Londýně. V roce 2003 k němu přibyla i tiskárna, také dle originálních Babbigeových plánů. V roce 2002 byl Diferenciální stroj sestrojen pro změnu v USA, v současné době tento pětitunový stroj najdeme v Kalifornii a jeho kopii taktéž v Londýně.

[3] Zdroj: http://collectionsonline.nmsi.ac.uk/browser.php?m=objects&kv=62748&i=123977.



Obrázek 1.3: Diferenciální stroj Charlese Babbage[3]

Analytický stroj (Analytical Machine) je považován za první univerzální počítací stroj světa (univerzální v tom smyslu, že dokáže simulovat jakýkoliv jiný stroj bez nutnosti hardwarové přestavby, jen s úpravou softwaru), byl již mnohem dokonalejší než Diferenciální stroj.

Jednalo se o programově řízený mechanický číslicový počítač (poháněný párou, už ne klikou), měl vlastní procesor, dva registry, uměl i podmíněné skoky a cykly. Kromě aritmetických operací také dokázal řešit algebraické a numerické rovnice, dokázal zhodnotit výsledky a přizpůsobit jim průběh dalšího výpočtu. Používal obdobu děrných štítků, v čemž se Babbage inspiroval u dříve zmíněného Jcaquarda.

Analytický stroj zůstal pouze na papíře, ale už v historii se objevily pokusy o jeho sestrojení. Donedávna se dařilo sestrojit spíše jeho části, s čímž začal vynálezcův syn (Henry Babbage). Problémy byly jak finančního rázu, tak i technologické (ozubená kolečka praskala apod.).

Analytický stroj nese ještě jedno prvenství – právě pro něj byly napsány první programy na světě. Prvním programátorem byla žena – *Augusta Ada King, hraběnka z Lovelace* (také hraběnka Ada Lovelace) rozená *Augusta Ada Byron* (byla dcerou známého anglického básníka lorda G. G. Byrona). Prvním programem světa, který vytvořila právě ona, je program na výpočet Bernoulliho čísel.

Poznámka

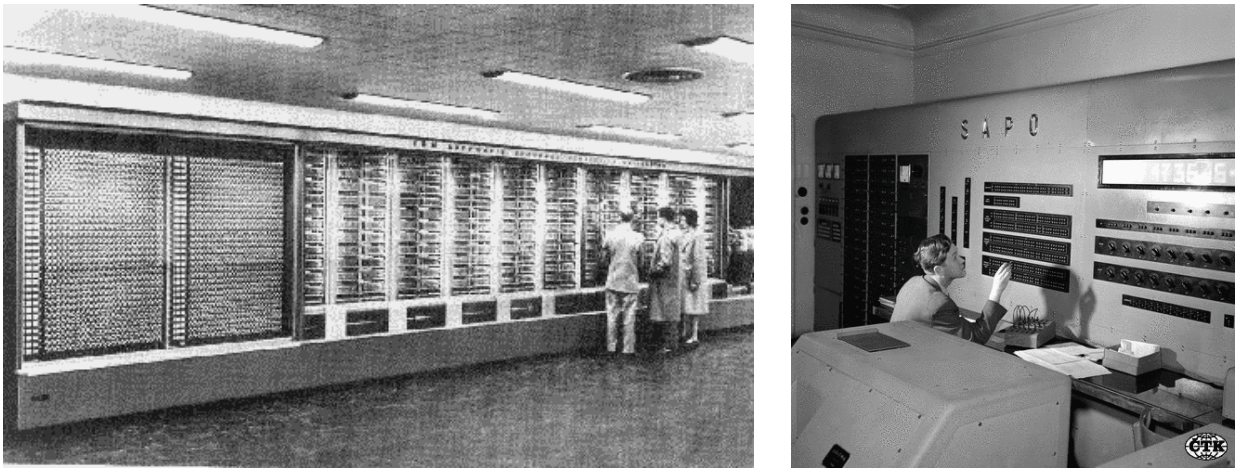
Po hraběnce Adě byl pojmenován *programovací jazyk Ada*, což byl jazyk trochu podobný Pascalu, ale s výraznými bezpečnostními mechanismy (tento jazyk vznikl za podpory americké armády).

Velkým krokem vpřed byl vynález *elektromagnetického relé*. K počítačům nulté generace sestrojeným za použití relé patřily například počítače německého vynálezce *Konráda Zuseho*, který vynalezl v 30.–40. letech 20. století postupně počítače Z1, Z2, Z3 a Z4, od Z2 používal také relé.

Z nich byl nejpřínosnější především Z3 z roku 1941, který jako první fungoval vcelku tak jak měl. Byl to první elektronický digitální programovatelný počítač složený z přibližně 2000 relé vážící jednu tunu. Jeho úkolem bylo pomáhat při projektování německých bomb, paradoxně jeho konec byl způsoben bombou (ale ne německou).

Z dalších jmenujme například americký počítač *MARK I* od Howarda Aikena (rok 1943, vážil 35 tun), anebo SAPO (SAmočinný POčítač), první československý počítač z roku 1957. Vidíme je na obrázku 1.4.

SAPO byl postaven z přibližně 7000 elektromagnetických relé a 400 elektronek (nebyl složen pouze z elektronek, proto ho neřadíme do první generace počítačů). Vznikl v laboratoři matematických strojů Ústředního ústavu matematického, do provozu byl uveden roku 1958 (uvedení do provozu nebylo zrovna okamžité, především z důvodů značné poruchovosti tohoto stroje). První funkční československý počítač by možná vznikl mnohem dřív, ale v padesátých letech (hlavně v jejich první polovině) se tomuto oboru moc nepřálo, kybernetika byla dokonce označována za „buržoazní pavědu“.



Obrázek 1.4: Počítače MARK 1 a SAPO[4]

1.2.2 První generace

Počítače první generace měly tyto vlastnosti:

- jako svůj základ používaly elektronky,
- žádný operační systém, žádné vyšší prog. jazyky, každý stroj měl svůj assembler,
- počítač zpracovával vždy jedinou úlohu, kterou zadával (živý) operátor.

Pořád šlo vpodstatě o velké soustavy skříní zabírající hodně místa. Potřebovaly sice celou místnost, ale na rozdíl od svých reléových předchůdců se do té místnosti vešlo taky něco jiného (stoly zaměstnanců apod.).

Elektronka (anglicky vacuum tube, vakuová trubice) se skládá z katody (žhavicí vlákno), anody (plíšek) a vodivých drátů, to vše je uzavřeno ve skleněné trubici s vyčerpaným vzduchem (tj. ve vakuu). Když je katoda nahřáta, dojde k emisi elektronů, které jsou přitahovány k anodě a tedy prochází proud.

Obvyklá velikost elektronky však byla několik centimetrů, a tedy sestavit celý počítač znamenalo mít dostatek místa, počítat s vysokou spotřebou elektřiny a důkladně chladit (elektronky hodně „topily“. Vzhledem k vysoké provozní teplotě a celkové složitosti byly počítače první generace velmi poruchové.

Mezi prvními elektronkovými počítači byly

- *Colossus* (1943) – britský počítač, vypomáhal v druhé světové válce (dešifroval zachycené německé depeše),
- *ENIAC* (1946) – první zcela elektronkový stroj (asi 17 500 elektronek), vážil 27 tun,
- *EDVAC* (1951) – na rozdíl od ENIACu byl binární, s univerzálnějším použitím.

[4] Zdroj: http://www.columbia.edu/acis/history/mark1.html, http://www.ceskenoviny.cz/tema/index_img.php?id=68769.

Prvním komerčně úspěšným počítačem první generace byl *UNIVAC* (UNIVersal Automatic Computer, 1951), stroj přímo navržený pro použití v administrativě a obchodu. Byl potomkem ENIACu (sestavili ho titíž lidé). UNIVAC původně nebyl vybaven zařízením pro práci s děrnými štítky, což byl ze začátku jeho velký handicap. Postupně byl tento problém vyřešen a také s UNIVACem bylo toto zařízení dodáváno.

Prvním zákazníkem byl Úřad pro sčítání lidu v USA, další zákazníci byli především z okruhu americké armády (letectvo, kartografie, námořnictvo apod.), ale také velké americké firmy (Remington Rand, General Electric, U.S. Steel, atd.). Cena byla pohyblivá (spíše stoupala), postupně překročila milion USD.

UNIVAC obsahoval přibližně 5200 elektronek, vážil 13 tun. Pracoval na frekvenci 2,25 MHz. Pracoval s čísly v dekadickém (desítkovém) zápisu, používal také znaménko. Paměť pracovala na principu *zpožďovací linky* na bázi rtuti (tj. značně neekologický výrobek).

V Československu byl roku 1963 dokončen elektronický počítač *Epos 1* (stejný ústav a téměř stejný tým jako u počítače SAPO). Skládal se z 8000 elektronek. Do sériové výroby se nedostal, protože byl velmi poruchový, ale jeho přínos spočíval především v ozkoušení některých originálních postupů.

1.2.3 Druhá generace

Druhá generace je datována do 50. a částečně 60. let 20. století. Ve výpočetní technice je to přelomové období, protože právě tehdy se začaly objevovat první opravdu stolní počítače, které se dostávaly i k „obyčejným“ lidem (i když v žádném případě ještě nešlo o domácí počítače dnes běžných rozměrů). Většinu známých počítačů té doby bychom dnes označili spíše jako „skříňové“.

Počítače druhé generace byly založeny na tranzistorech. *Polovodič* je pevná látka, jejíž elektrická vodivost závisí na vnitřních nebo vnějších podmínkách, tj. není stálá a dá se ovlivňovat.

Tranzistor: tranzistorový efekt byl objeven roku 1947 v Bellových laboratořích. Za tento objev jim byla roku 1956 udělena Nobelova cena za fyziku. Tranzistor je polovodičová součástka, používají se polovodiče typu P a N

- P: do čtyřmocného křemíku je přidán některý pětimocný prvek,
- N: do čtyřmocného křemíku je přidán některý trojmocný prvek.

Má (nejméně) tři elektrody – kolektor, bázi a emitor (u bipolárních tranzistorů). podle uspořádání použitých polovodičů rozlišujeme tranzistory PNP a NPN (prostřední je báze), na bázi je připojen proud, jeho velikost ovládá proud mezi emitorem a kolektorem.

Kromě používání tranzistorů měly počítače druhé generace také tyto vlastnosti:

- *dávkový systém práce* – programy s daty pro zpracování jsou operátorem naskládány za sebe, když je dokončen jeden program, automaticky se začne zpracovávat další program z dávky,
- kromě assembleru další *programovací jazyky* (například FORTRAN, COBOL).

Jedním z prvních tranzistorových počítačů druhé generace byl *PDP-1* firmy DEC. Měl přibližně rozměry skříně a byl určen pro větší firmy (prodalo se celkem 55 kusů, byl to komerčně velmi úspěšný stroj). Právě pro tento počítač vznikl první UNIX.

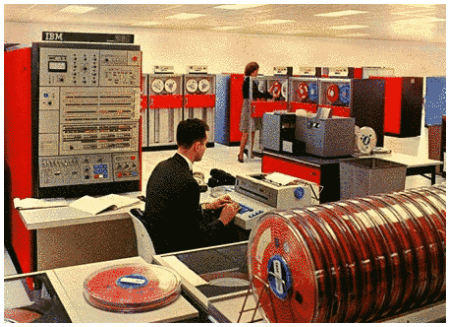
Prvním komerčně dostupným československým tranzistorovým počítačem byl *MSP* (Malý Stolní Počítač, 1965), dále pro hromadné zpracování dat byl vyvinut počítač DP 100, taktéž komerčně úspěšný. Další čs. počítač, *Epos 2*, pokračovatel počítače *Epos 1* z první generace, byl také složen z polovodičových součástek (diod a tranzistorů), jeho sériová výroba začala roku 1969.

1.2.4 Třetí generace

Počítače třetí generace jsou založeny na integrovaných obvodech.

Integrovaný obvod je elektronická součástka integrující drobnější součástky (tranzistory, rezistory, kondenzátory, apod.) na jedné polovodičové destičce (obvykle křemíkové), v plastovém pouzdře či jinak chráněná. Byl vynalezen ve společnosti Texas Instruments roku 1958. Vlastnosti počítačů třetí generace:

- první operační systémy (CP/M apod.),
- vyšší programovací jazyky (například ALGOL, LISP, Pascal, BASIC),
- disketová mechanika (8" od IBM, rok 1971),
- možnost paralelního zpracování dat.



Obrázek 1.5: Počítač IBM 360[5]

Hlavními zástupci této generace jsou IBM System 360, Siemens, Tesla 200 a 300.

Především z důvodu poměrně rychlého technického rozvoje rozlišujeme také *třiapůltou generaci*. Má tyto vlastnosti:

- používají se integrované obvody vysoké integrace,
- mikroprocesory, minipočítače, terminály, obrazovka,
- první mikroprocesory firem Intel a Motorola.

První mikroprocesor byl *Intel 4004* (vývoj začal roku 1971). Tento 4bitový mikroprocesor původně vznikl na zakázku japonské firmy Busicom a byl určen do kalkulátoru. Později si společnost Intel uvědomila možnosti tohoto svého výrobku a pokračovala v jeho vývoji.

Ryze československé počítače už v té době nevznikaly, bohužel byla státem uplatňována politika povinného dovozu počítačů a komponent z východního bloku, především ze Sovětského svazu (*JSEP* – Jednotný systém elektronických počítačů) a pokud se vyvíjelo něco nového, tak pod taktovkou Sovětského svazu. To znamenalo konec samostatného vývoje na dlouhou dobu. Přesto u nás byly počítače vyráběny (například *Tesla 200* z roku 1969).

U třiapůlté generace šlo o projekt *JSEP2*, u nás byly v rámci tohoto projektu vyvinuty počítače *EC1025*, *EC1026* a *EC1065*. Další společný projekt východního bloku států byl *SMEP* (Systém malých elektronických počítačů), kam lze zařadit první české klony počítačů IBM PC.

1.2.5 Čtvrtá generace

Čtvrtá generace trvá prakticky dodnes. Hlavním rysem je používání *integrovaných obvodů velmi vysoké integrace*. Objevila se první mechanika CD-ROM (rok 1984).

Dostupné osobní počítače jsou nejdřív 8bitové (Altair, IBM, Apple, Commodore, Atari, ZX Sinclair), později 16bitové (IBM, Apple, ...). Objevil se také první standard pro napájení počítače – AT (Advanced Technology). První osobní počítač, který tento standard splňoval, byl IBM PC AT.

Altair 8800 (rok 1975) byl prvním komerčně úspěšným počítačem určeným pro domácnosti. Neprodával se však „vcelku", zákazník dostal komponenty a návod a tento počítač si vlastnoručně poskládal. Další revoluci v domácích počítačích znamenal počítač *Apple II*. Pro následující vývoj je také důležitý příchod *IBM PC 5150* (rok 1981) dodávaný s operačním systémem MS-DOS. Roku 1982 se objevil předchůdce přenosných počítačů (notebooků), *GRiD Compass 1100* (vážil 5 kg, byl vybaven modemem a plochou obrazovkou, ale neměl disk).

[5] Zdroj: http://www.plyojump.com/classes/mainframe_era.html.

V Československu se do škol dostaly české stroje *PMD* (od roku 1985), *IQ-151* (rok 1984, měl procesor Tesla MHB 8080 a 32 KB operační paměti), *Didaktik* (kompatibilní s Sinclair ZX Spectrum) a *Tesla Ondra*. Především počítač IQ-151, tzv. „ikvéčko“, využívaný na školách při výuce programování, se vryl do paměti svých uživatelů, kromě jiného taky velmi špatnou klávesnicí (klávesy se obtížně mačkaly) a tím, jak moc „topil“. Z dovozových strojů byly oblíbené především *Atari ST* (16bitový počítač s velmi zajímavým a inovativním grafickým rozhraním a myší) a *Amiga 500* (taktéž grafické rozhraní s multitaskingem).

1.3 Analogové počítače

Počítače dělíme na *digitální* (také číslicové) a *analogové*. U osobních počítačů postupně převládly digitální, ale v historii tomu vždy tak nebylo a dodnes existují oblasti použití, kde je analogové zpracování výhodou.

Digitální počítač: digitální signál nabývá hodnot 0 a 1, nic mezi tím, digitální (číslicové) počítače zpracovávají digitální hodnoty, jsou univerzálnější.

Analogový počítač: analogový signál je spojitý, může probíhat celou škálu mezi 0 a 1, analogové počítače zpracovávají analogový signál (nejen elektrický), používají se především v souvislosti s fyzikálními pochody.

Analogový počítač není univerzální, je konstruován pro jeden konkrétní účel. Programování spočívá ve vhodném propojení jednotlivých částí tak, aby celek plnil daný účel – zpracovával zadané vstupy na očekávané výstupy.

Jaký je tedy rozdíl mezi digitálním a analogovým zpracováním dat?

- u digitálního počítače při převodu analogové hodnoty (třeba fyzikální veličiny) dochází k zaokrouhlovacím chybám, výpočet je nepřesný (omezení datového typu),
- u analogového počítače hodnota proudu nebo napětí určuje momentální hodnotu dané fyzikální veličiny, modelování fyzikálních dějů je mnohem přesnější,
- v jiných typech výpočtů je naopak přesnost analogového počítače horší, zejména pokud jde o abstraktní matematické výpočty přímo neodvoditelné z použitelných fyzikálních veličin,
- analogové počítače pracují v reálném čase, zpoždění je téměř nulové (informace „protéká“ systémem).

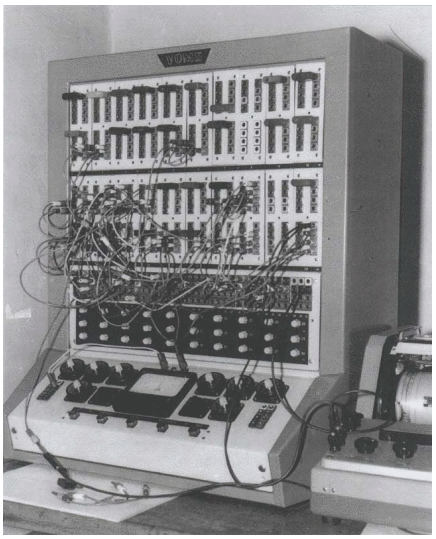
Tedy tam, kde nehrozí zaokrouhlovací chyby způsobené převodem z použitelné fyzikální veličiny a kde spíše provádíme abstraktní matematické výpočty, vítězí digitální počítače.

Známí zástupci analogových počítačů:

- mozek,
- Antikythérský mechanismus,
- logaritmické pravítko,
- vojenství, především námořnictvo, pro řízení zaměřování (Argo, HACS, Dreyerův zaměřovací tabulátor, Nordenův zaměřovač) – mechanické, elektro-mechanické,
- Direct Analog Computer (1949) – první elektrický počítač,
- MONIAC (1950) – hydraulický počítač simulující ekonomiku Spojeného království,
- ELWAT – Polský analogový počítač,
- MEDA – analogový počítač vyrobený ve Výzkumném ústavu matematických strojů.

Ve skutečnosti se analogové zpracování používá i dnes. Například chladiče procesoru a jiných komponent bývají řízeny některým vhodným mechanismem, třeba DC – změna napětí na vodičích určuje, jak rychle se má větrák otáčet.

[6] Zdroj: http://www.fi.muni.cz/usr/jkucera/pv109/vystavka/xprocha1_index.html.



Obrázek 1.6: Československý analogový počítač MEDA[6]

Kapitola 2

Struktura počítače a BIOS

Rychlý náhled: V této kapitole se budeme zabývat (abstraktní) strukturou počítače. Probereme postupně několik modelů – Von Neumannovo schéma, Harvardské schéma a Data-flow, včetně jejich vztahu k současným výpočetním architekturám. Dalším probíraným tématem je BIOS a jeho nástupce UEFI.

Klíčová slova: Von Neumannovo schéma, Harvardská architektura, Data-flow architektura, Flynnova taxonomie, SISD, SIMD, MISD, MIMD, firmware, BIOS, UEFI, pre-boot aplikace, secure boot, hodiny reálného času (RTC).

Cíle studia: Po prostudování této kapitoly si ujasníte základní strukturu počítače a tok dat v systému – na tuto znalost navazujeme v dalších kapitolách. Dále se seznámíte s BIOSem/UEFI, tedy firmwarem základní desky, jeho vlastnostmi a funkcemi.

2.1 Základní typy počítačových architektur

2.1.1 Von Neumannovo schéma

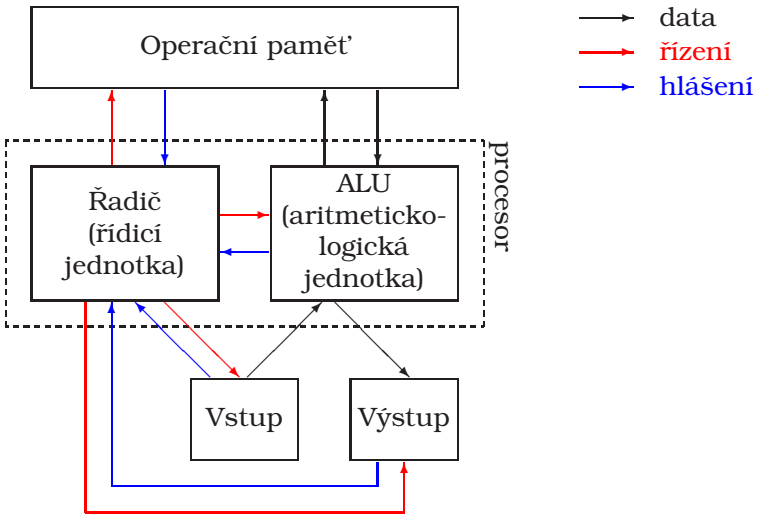
John von Neumann, celým jménem John Ludwig von Neumann, resp. János Neumann, byl americký matematik židovského původu narozený v Maďarsku. Od dětství projevoval známky geniality, měl výbornou paměť, ke které přidával velmi dobrou schopnost logického myšlení.

V 17 letech publikoval svou první vědeckou práci, ve 22 letech již byl asistujícím profesorem na Berlínské univerzitě (odpovídá titulu docenta).

Čím například byl von Neumann významný pro výpočetní techniku:

- spolutvůrce *Teorie her* (použití: ekonomika a jakékoliv další obory s projevy chaosu),
- podílel se na konstrukci několika nejznámějších počítačů (například ENIAC),
- tvůrce *operační teorie kvantové mechaniky* (Von Neumannova algebra),
- jaderná fyzika, spolupodílel se na konstrukci první jaderné bomby,
- r. 1949 vytvořena *von Neumannova matematická pravidla pro konstrukci robotů*, kteří se budou sami zdokonalovat a reprodukovat; možnosti využití nastávají až nyní, NASA je chce využít pro konstrukci robotů pro vesmír.

Roku 1945 navrhl schéma samočinného počítače, po něm nazvané *von Neumannovo schéma*, jehož nákres vidíme na obrázku 2.1.



Obrázek 2.1: Von Neumannovo schéma

K jednotlivým částem schématu:

1. *Operační paměť* – uchovávání programu, dat, mezivýsledků, výsledků výpočtu,
2. *Řadič* (řídicí jednotka) – řídí činnost všech částí počítače pomocí řídicích signálů,
3. *ALU* (aritmeticko-logická jednotka) – provádí veškeré aritmetické výpočty a logické operace,
4. *Vstup* (vstupní zařízení) – zařízení pro vstup programů a dat,
5. *Výstup* (výstupní zařízení) – zařízení pro výstup programů a dat.

Hlavní myšlenkou von Neumanna, která je do určité míry používána i dnes, je *umístění programu a dat do téže paměti* (tj. do operační paměti). Program spustíme, pak je podle tohoto programu vytvořen proces, který celý pracuje v operační paměti (načte se tam jeho programový kód, globální proměnné, případně dynamické knihovny, atd.). Když tento program (vlastně proces, abychom byli přesnější) otevře některý soubor, tento soubor se přenese (mapuje) opět do operační paměti.

U starších výpočetních zařízení (před tímto schématem) tato vlastnost nebyla splněna, obvykle jsme měli program na některém vnějším paměťovém médiu, ze kterého se také spouštěl, a data na jiném médiu (například děrných štítcích).

Další důležité vlastnosti:

- existuje instrukční sada počítače, ve které je psán program, tato instrukční sada je samozřejmě známa programátorům,
- protože je program uložen v paměti, je možné ho za běhu modifikovat nebo jinak využívat, s instrukcemi lze zacházet jako s daty (myšlenkou bylo umožnit „samoprogramování“ počítačů, běžně využíváno při ladění programů nebo v překladačích),
- instrukce jsou zpracovávány *sekvenčně*,
- je zpracováván pouze jeden program, nikoliv více zároveň.

Poslední dva body jsou dnes běžně porušovány, s čímž se seznámíme v kapitole o procesorech.

Poznámka

Na toto schéma a některé jeho součásti se budeme odvolávat v následujících kapitolách, zejména v kapitole o procesoru. Procesor má svůj řadič, který je v procesoru jakýmsi „dirigentem“, řídí činnost celého

procesoru. Ve výpočetních jádrech máme „dělníky“ – ALU – provádějící aritmetické a logické operace. V procesoru máme také pomocnou paměť (registry, cache paměti) a až přes ni procesor komunikuje s operační pamětí. Toto schéma vpodstatě zůstává zachováno i v současných výpočetních zařízeních, až na drobnosti jako je zavedení paralelního zpracování programů i instrukcí v rámci jednoho programu.

Použití v současnosti:

- Assembler jako programovací jazyk nízké úrovně funguje podle von Neumannovy architektury, data mohou být volně smíchána s instrukcemi, třebaže se to nedoporučuje,
- překladač (compiler) programovacího jazyka – jeho výstupní data jsou vlastně instrukcemi jiného programu,
- většina dnešních procesorů *téměř* odpovídá von Neumannově architektuře.

Odlišnosti současných procesorů od von Neumanna:

- v jednom počítači může být více než jeden procesor, případně jeden procesor může mít více jader,
- v jednom okamžiku může být spuštěno více programů, dokonce jich může více běžet najednou (multitasking, multiprocessing),
- existují I/O zařízení, která jsou vstupně-výstupní (dotykové obrazovky, multifunkční zařízení, atd.),
- není nutné mít program v operační paměti celý, je možné zavést do paměti jen potřebnou část,
- virtuální paměť.

2.1.2 Harvardská architektura

Další architekturou, která má vztah k dnes běžným počítačovým architekturám, je Harvardské schéma. Název odvozen od počítače *Harvard MARK 1*, celým názvem IBM Automatic Sequence Controlled Calculator (ASCC), reléového počítače nulté generace na Harvardské univerzitě. MARK 1 přijímal program na děrné pásce a data na elektromechanickém médiu.

Rozdíl oproti von Neumannově architektuře:

- oddělení instrukcí programu od dat (separované paměti), tj. v různých paměťových prostorech s oddělenou adresací a oddělenými cestami na sběrnici do CPU,
- nemožnost modifikovat instrukce.

Harvardský koncept přináší určité výhody. Kromě větší bezpečnosti to taky znamená, že procesor může zároveň načítat data i instrukce, protože pro ně používá různé komunikační kanály.

Od Harvardského schématu odvozujeme *Modifikovanou Harvardskou architekturu*. Oproti původní je přidána možnost zacházet s instrukcemi jako s údaji, tj. přenést část kódu do paměti určené pro data.

V současné době se Harvardská architektura (spíše v modifikované podobě) využívá například ve specializovaných čipech určených pro zpracování videa a audia.

V moderních procesorech je tento koncept použit jen uvnitř procesoru pro cache paměti, kdy je především cache první úrovně rozdělena na dvě části – část pro data (Data Cache) a část pro instrukce (Instruction Cache). Čímž poněkud předbíháme.

2.2 Instrukční a datové proudy – Flynnova taxonomie

Instrukční proud je proud instrukcí, který je přijímán výpočetní jednotkou (procesorem). *Datový proud* neobsahuje instrukce, ale data, jde tedy o proud dat.

V souvislosti s počítačovými architekturami můžeme počítače rozdělit podle počtu instrukčních a datových proudů do čtyř skupin. Původně byly takto klasifikovány (ne)paralelní systémy, dnes má smysl používat Flynnovu taxonomii především na instrukční sady (především první dva typy).

1. *SISD* (Single Instruction, Single Data) – plně odpovídá von Neumannově architektuře, jedna výpočetní jednotka zpracovává jeden proud instrukcí a jeden proud dat,

   ⇒ instrukční sada s jednoduchými instrukcemi typu sčítání dvou čísel apod., jedna instrukce zpracovává statický počet parametrů (např. sčítáme vždy dvě čísla, nikoliv různý počet),
2. *SIMD* (Single Instruction, Multiple Data) – tentýž proud instrukcí se používá na více proudů dat (může to znamenat několik výpočetních jednotek se stejným programem, ale různými daty),

   ⇒ instrukční sada s vektorovými instrukcemi, tedy tatáž instrukce je použita na celý vektor dat (vektor může mít různou velikost podle množství dat, která je takto nutno zpracovat), typickým příkladem jsou multimediální instrukce,
3. *MISD* (Multiple Instruction, Single Data) – více instrukcí na stejných datech, řetězené procesory (pipeline processors), v praxi se nepoužívá,
4. *MIMD* (Multiple Instruction, Multiple Data) – víceprocesorové systémy:
   - multiprocessors (multiprocesory) – procesory (nebo jádra) sdílejí paměť, nutno řešit současné přístupy do paměti,
   - multicomputers (multipočítače) – procesory nesdílejí paměť, spolupracují jiným způsobem (například zasíláním zpráv), typický příklad jsou distribuované systémy, serverové farmy, výpočetní clustery.

Poznámka

S těmito zkratkami se setkáme u procesorů, když budeme probírat instrukční sady.

Řešením problémů se sdílenou pamětí v multiprocesorových systémech je architektura *NUMA* (Non-Uniform Memory Access). Používá se u systémů s velkým množstvím procesorů, které by „jen tak“ bylo náročné spravovat, hlavně co se týče adresování. Celý systém je rozdělen do tzv. uzlů, přičemž každý uzel má přiděleny své procesory, paměť a další komponenty. Jednotlivé uzly v systému jsou spravovány částečně samostatně, což zjednodušuje práci na tomto systému.

2.3 BIOS

Firmware je základní software (téměř) každého zařízení (základní deska, rozšiřující karty, pevný disk, tiskárna, apod.). Zařízení je firmwarem vybaveno již od výrobce, nachází se většinou na paměťovém čipu (ROM, EPROM, flash nebo podobném typu). V některých typech zařízení se firmwaru říká BIOS.

Firmware slouží k identifikaci zařízení a umožňuje toto zařízení správně používat. Je nutný ke zjištění správných ovladačů, které je třeba instalovat, s firmwarem pracují také nástroje na diagnostiku hardwaru.

Chyby ve firmwaru mohou způsobit chybovost zařízení, nedostupnost některých funkcí nebo mohou například zařízení zpomalit či zhoršit jeho komunikaci s jinými zařízeními v počítači.

BIOS (Basic Input/Output System) základní desky (tj. firmware základní desky) je systém, který zajišťuje základní funkce počítače, obvykle ve vztahu ke komunikaci s periferiemi. Jedná se o nejdůležitější firmware v celém počítači, jednoduché nízkoúrovňové softwarové rozhraní k hardwaru počítače. Firmwaru jiných komponent se také někdy říká BIOS, můžeme například slyšet pojem „BIOS grafické karty“.

BIOS je uložen v paměti (na čipu) typu ROM, EEPROM nebo flash PROM. Pokud se jedná o BIOS základní desky (tj. její firmware), čip s ním najdeme na základní desce. Pracuje s konfigurací základní desky a k ní připojených komponent uloženou většinou v CMOS RAM paměti napájené z baterie (čip i baterie jsou na základní desce). S typy pamětí se podrobněji seznámíme v jedné z následujících kapitol.

Poznámka

Takže k BIOSu se vlastně vztahují dva čipy na základní desce – v jednom z nich (většinou typu flash PROM) je uložen spustitelný kód BIOSu, v druhém (většinou typu CMOS napájeném z baterie) najdeme konfiguraci základní desky a k ní připojených komponent.

Nejznámější výrobci BIOSu:

- AMI (American Megatrends) – AMI BIOS,
- Award-Phoenix – Award BIOS a Phoenix BIOS,
- další – Insyde, General Software,
- Open BIOS – volně šiřitelný včetně zdrojového kódu (pod GNU GPL), lze volně upravovat.

BIOS Setup je speciální program na nastavení voleb souvisejících s funkcemi BIOSu, který je možné spustit v krátkém časovém okamžiku mezi testováním hardwarových komponent po zapnutí počítače a spuštěním operačního systému. V tomto okamžiku se dole na obrazovce objeví výzva („Press DEL to enter Setup“, případně jiná klávesa – F2, F1, CTRL+ALT+ESC, CTRL+ESC, atd.). Pracujeme v textovém, pseudografickém režimu, používáme anglickou klávesnici.

Možnosti nastavení jsou různé pro různé BIOSy, především:

- nastavení data a času, zobrazení informací o připojeném hardwaru,
- nastavení posloupnosti bootování zařízení, nastavení některých vlastností zařízení včetně procesoru,
- režim ACPI (menu se nejčastěji jmenuje „Power Management Setup“),
- zapnutí komunikačních portů (COM, LPT, USB), podpora RAID, IR, nastavení hesla,
- BIOSy dražších desek mají různé volby pro přetaktování procesoru a pamětí.

SMBIOS (System Management BIOS) je specifikace pro datové struktury týkající se údajů v BIOSu a přístupové metody k těmto strukturám. Účelem SMBIOSu je umožnit operačnímu systému a aplikacím softwarově přistupovat k informacím souvisejícím s BIOSem.

Další informace

Seriál o BIOSu na SvětHardware
https://www.svethardware.cz/theme_doc-E0134E73EE970A4BC125747100209659.html

2.4 Zapnutí počítače

Popíšeme si, co se děje při zapnutí počítače. Nejdřív je spuštěn BIOS:

1. vynuluje registry procesoru,
2. nastaví registr programový čítač na adresu, na které se nachází BIOS (pro IBM kompatibilní je to obvykle adresa v segmentu začínajícím na adrese F000h),
3. takto se spustí BIOS.

BIOS spustí test *POST* (Power-On Self-Test), který postupně kontroluje fyzický stav a připojení jednotlivých hardwarových komponent k základní desce.

1. nejdřív je zkontrolován procesor, protože hraje velkou roli v dalších testech (tyto testy jsou složeny z instrukcí, které provádí procesor),
2. pak je prověřen samotný test POST,
3. přes sběrnici POST kontroluje veškeré obvody a zjišťuje, zda jsou provozuschopné,
4. kontrola časovače (timeru),
5. kontrola videopaměti, pak je videopaměť připojena do paměťového fondu a adresována, začíná fungovat obrazovka a na ni se vypisují hlášení o všech následujících testech,
6. kontrola paměťových modulů,
7. kontrola klávesnice, začínají fungovat klávesy a je možné se například dostat do BIOS Setup (klávesa se liší podle typu BIOSu),
8. kontrola dalších zařízení; pokud zařízení obsahuje vlastní BIOS (třeba řadičě disků), je tento BIOS začleněn do hlavního BIOSu.

Po testu POST je již zjištěna hardwarová konfigurace systému. V BIOSu (CMOS paměti) je jako součást konfigurace uložena posloupnost paměťových médií, ze kterých je povoleno zavést operační systém (pevný disk, CD, DVD, flash disk, apod.). První médium ze seznamu, na kterém je funkční operační systém, je použito pro zavedení operačního systému (ovšem nemusí jít zrovna o zavaděč operačního systému, ale třeba jen o *boot managera*, který umožní uživateli rozhodnout se mezi několika operačními systémy).

Postup je následující:

1. BIOS spustí základní zaváděcí program „bootstrap loader“,
2. bootstrap loader postupně prověřuje paměťová média z této posloupnosti a zvolí první, ze kterého lze zavést systém,
3. bootstrap loader najde *zaváděcí záznam* na tomto médiu a předá mu řízení (Windows BootMgr, GRUB, atd.),
4. dále již pracuje zaváděcí záznam příslušného operačního systému instalovaného na paměťovém zařízení.

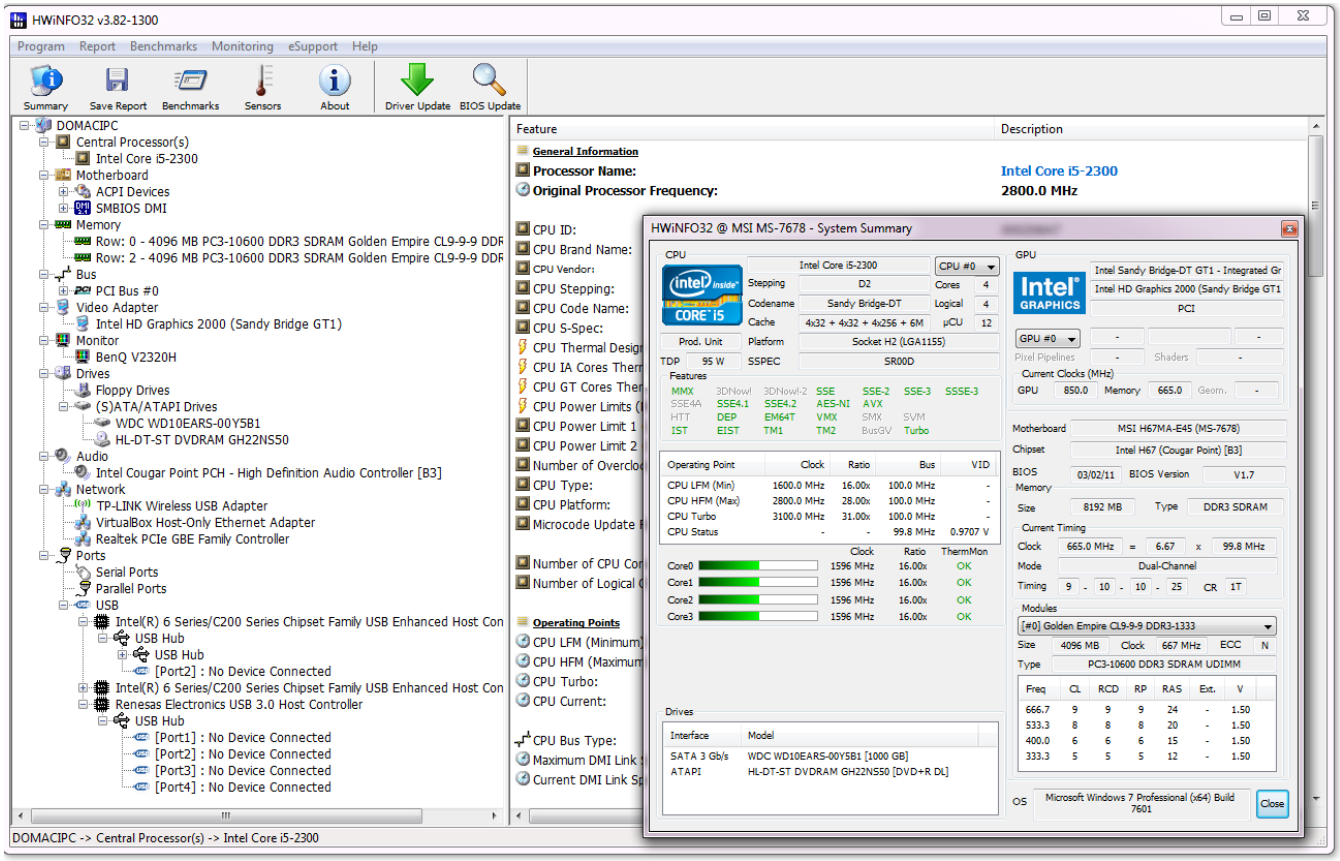
Systém může nastartovat dvěma způsoby:

1. *Studený start* – po připojení počítače k elektrické síti (resp. po přivedení napětí k počítači, zapneme počítač), platí vše výše uvedené,
2. *Teplý start* – po stisknutí CTRL+ALT+DEL a potvrzení restartu (správněji resetu); šetrnější ke komponentám počítače, POST neproběhne.

Další možnost je samozřejmě probuzení počítače z režimu spánku či hibernace, tomu se budeme věnovat spíše v předmětech zaměřených na operační systémy.

Další informace

To, které komponenty počítače byly rozpoznány při startu, je možné zjistit celkem jednoduše pomocí různých nástrojů. Přímo ve Windows máme *Systémové informace* (můžeme spustit příkazem `msinfo32`, a nebo se k nim doklikat přes tlačítko Start). Lepší jsou však nástroje třetích stran, například velmi oblíbený program HWInfo32 nebo HWInfo64 (podle naší hardwarové platformy) zdarma dostupný na webu http://www.hwinfo.com/.



Obrázek 2.2: Zjištění informací o komponentách v HWInfo32

2.5 EFI a UEFI

BIOS vznikl před několika desetiletími a byl projektován podle tehdy běžných požadavků na hardware, přičemž se od té doby už moc nezměnil. Sice postupně přibývaly nové položky k nastavení v BIOS Setup, ale limity pro ukládané datové struktury zůstaly.

EFI (Extensible Firmware Interface) je následovník BIOSu s otevřeným kódem. Účelem bylo vytvořit systém jednoduchý, snadno konfigurovatelný. Z nejdůležitějších vlastností:

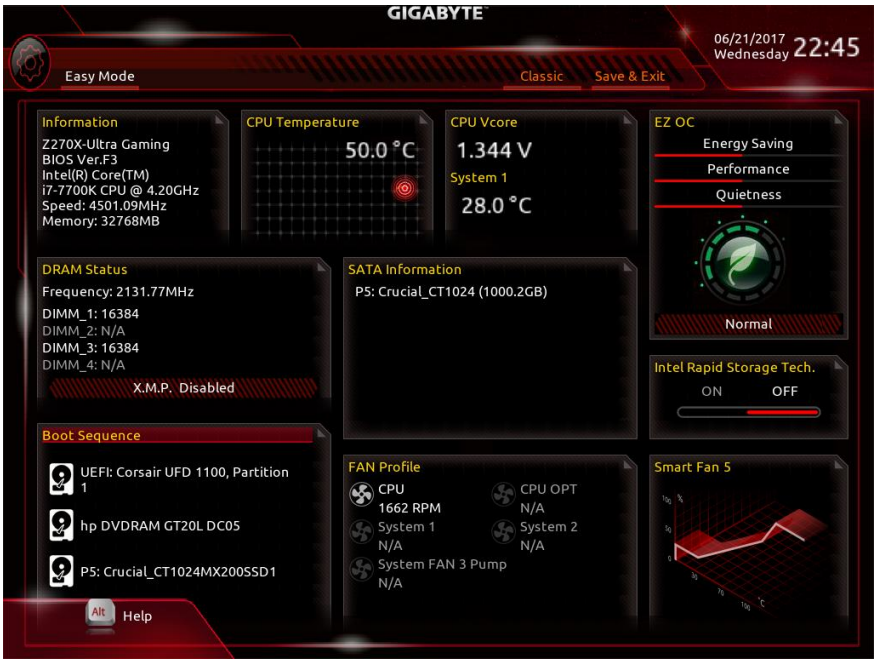
- podpora 64bitových systémů (délka adres, také se navýšily limity pro datové struktury),
- vlastní koncepce ovladačů (tj. už v rozhraní EFI je možné používat některá zařízení včetně myši a síťového rozhraní, není nutné kvůli tomu startovat operační systém),
- v novější specifikaci obsahuje podporu šifrování a autentizace po síti,
- většinou se konfiguruje v grafickém režimu, místo textového (pseudografického) prostředí BIOS Setup.

Je psán v jazyce C, proto je kód EFI mnohem delší než u BIOSu, na základní desce musí být flash čip o větší kapacitě.

Existuje více variant EFI, ale nejznámější a nejrozšířenější je *UEFI* (Unified Extensible Firmware Interface) – na vývoji spolupracuje konsorcium firem (Intel, AMD, Apple, ARM, Phoenix, AMI, atd.).

Následující text se bude vztahovat k UEFI, protože pojmy EFI a UEFI jsou v současné době víceméně zaměnitelné.

[1] Zdroj: https://www.anandtech.com/show/11618/asus-prime-z270-a-gigabyte-z270x-ultra-gaming-motherboards-review/5.



Obrázek 2.3: UEFI pro základní desky Gigabyte[1]

2.5.1 Zařízení

Ovladače zařízení existují už v UEFI, důsledkem je lepší přístup k zařízení bez nutnosti spuštění operačního systému (UEFI startuje rychleji než operační systém). Ovladače zařízení mohou být psány ve vyšších programovacích jazycích, například v C (nejen v Assembleru), což hodně ulehčuje práci programátorům ovladačů.

✎ Také je možné zabudovat do UEFI *vzdálenou správu zařízení* (kvůli zásahům do konfigurace zařízení, zálohování, aktualizacím a některým dalším aktivitám není nutné vzdáleně spouštět operační systém), to je užitečné pro správce výpočetní techniky ve firmách.

2.5.2 Ovládání UEFI a pre-boot aplikace

UEFI může (nemusí) mít grafické rozhraní podobné Windows, tlačítka, ovládání myší. Síťové rozhraní je dostupné už v rámci UEFI, proto je možné přímo z rozhraní provádět po síti aktualizace.

✎ Mnoho operací lze provádět v *UEFI shellu* – mj. práce se soubory a adresáři, přístup na síť, používání .NSH skriptů.

☎ **Další informace**

O UEFI shellu je pěkný článek na https://www.linuxexpres.cz/software/efi-shell. ☎

✎ *Pre-boot aplikace* jsou drobné aplikace spustitelné přímo z UEFI, nainstalované na základní desce nebo na skrytém oddílu pevného disku. Jde například o diagnostické programy, zálohování, programy pro konfiguraci firmwaru, *zavedení operačního systému*, u některých výrobců se počítá i s drobnějšími multimediálními programy (prohlížení obrázků, přehrávání hudby či videa, práce s e-maily, webový prohlížeč, apod.). Účelem většiny pre-boot aplikací je vyhnout se zdlouhavému startu operačního systému, když po počítači chceme jen nějakou „jednoduchou“ úlohu.

Zavaděče operačních systémů (jako je Windows, Linux, MacOS, Solaris apod.) jsou při použití UEFI

ve skutečnosti pre-boot aplikacemi, které se spouštějí uvnitř UEFI. Tím se zcela mění koncept startu operačních systémů, boot loader je již integrální součástí UEFI a jednotlivé zavaděče jsou do UEFI dodány vždy při instalaci operačního systému.

Poznámka

Na straně 14 je popsán postup zapnutí počítače s původním BIOSem. Bootování s UEFI je v mnohém podobné (taky se nejdřív provádějí kontroly všeho možného, zejména hardwaru, proběhne POST), bootstrap loader místo hledání zaváděcích záznamů operačních systémů spustí stanovenou pre-boot aplikaci, která zavede „svůj“ operační systém.

Aby na počítači mohl být používán UEFI, musí být podporován také operačním systémem – operační systém používá na počítači s UEFI jiný zavaděč než na počítači s BIOSem. Podpora UEFI je v nejběžnějších operačních systémech následující:

- Windows – pro desktop od Vista SP1, některé varianty Windows 2000 Server, novější verze Windows Server (2003, 2008), Windows XP 64bit,
- Linux – verze asi od roku 2000, obvykle není problém,
- Apple MacOS – od přechodu na procesory Intel,
- ostatní UNIXové systémy – různě, většinou podpora začíná v letech 2002–2005.

2.5.3 Secure Boot

Secure Boot je funkce, kterou najdeme ve většině implementací UEFI. Jejím účelem je ochránit počítač před neoprávněnými zásahy do jádra – především si vynucuje používání certifikátů u instalovaných a spouštěných operačních systémů a také u jakýchkoliv modulů, které jsou načítány do jádra operačního systému.

Instalovaný či spouštěný operační systém (zavaděč a jádro) a každý modul jádra musí být digitálně podepsán důvěryhodným certifikátem. Ne každému certifikátu se důvěřuje. Pokud systém či modul nemá důvěryhodný podpis, instalaci/spuštění/načtení je zabráněno.

V praxi to znamená, že na počítač s UEFI a zapnutým Secure Boot nelze nainstalovat operační systém, který není opatřen důvěryhodným certifikátem. Co se týče produktů společnosti Microsoft, certifikát mají Windows od verze 8, svůj certifikát také mají mnohé UNIXové systémy včetně běžných distribucí Linuxu.

Pokud chceme instalovat systém bez certifikátu, je třeba funkci Secure Boot vypnout. U některých základních desek to bohužel nejde, u jiných ano, záleží na výrobci. Je to podobné jako u BIOSu – výrobce určuje, jak bude rozhraní vypadat a jaké volby v něm budou dostupné (a pod jakými názvy).

Další informace

- http://www.uefi.org
- http://www.intel.com/technology/efi
- http://en.wikipedia.org/wiki/Extensible_Firmware_Interface
- http://www.virtualization.info/2007/10/bios-leader-phoenix-working-on-its-own.html

2.6 Upgrade firmwaru (BIOSu)

Může se stát, že BIOS (obecně firmware) je potřeba aktualizovat (provést upgrage). Důvody mohou být různé – v lepším případě chceme vyměnit procesor za novější, vylepšit komunikaci systému s příslušným kusem hardwaru (třeba přidat nové funkce tiskárně, přidat podporu pro jiný operační systém (nebo novou verzi) nebo zrychlit komunikaci se zařízením), v horším případě je třeba odstranit chyby firmwaru. Můžeme se setkat se zařízeními, která již fyzicky podporují některou novou funkci, ale protože si výrobce ještě není jist skutečným dopadem této funkce nebo ji ještě nemá zahrnutu ve firmwaru zařízení, uživatel se k dané funkci nedostane. Proto upgrade firmwaru může někdy znamenat zpřístupnění nových funkcí zařízení – tato funkce je v zařízení fyzicky implementována, upgrade k ní jen otevře přístup.

Některé chyby firmwaru mohou naštvat, jiné dokonce znamenají velké problémy. Notoricky se chyby objevují například ve firmwaru síťových zařízení (třeba VDSL routerů). Zapojíme a nakonfigurujeme zařízení a pak zjistíme, že zrovna s naší konfigurací operačního systému nekomunikuje, že firewall reaguje v některých případech naprosto nečekaně a nedá se to změnit, že router vydává velmi nepříjemný zvuk či se hodně zahřívá, že lze použít jen WEP šifrování a WPA či dokonce WPA2 není dostupné, atd. U tiskárny se dvěma zásobníky papíru můžeme například zjistit, že jeden z nich funguje jen tehdy, když druhý není prázdný, u tiskárny se taky může stát, že když dojde papír v zásobníku, tiskárna pořád odmítá tisknout i po doplnění papíru i po restartu tiskárny (tady obvykle pomůže restart celého počítače), apod.

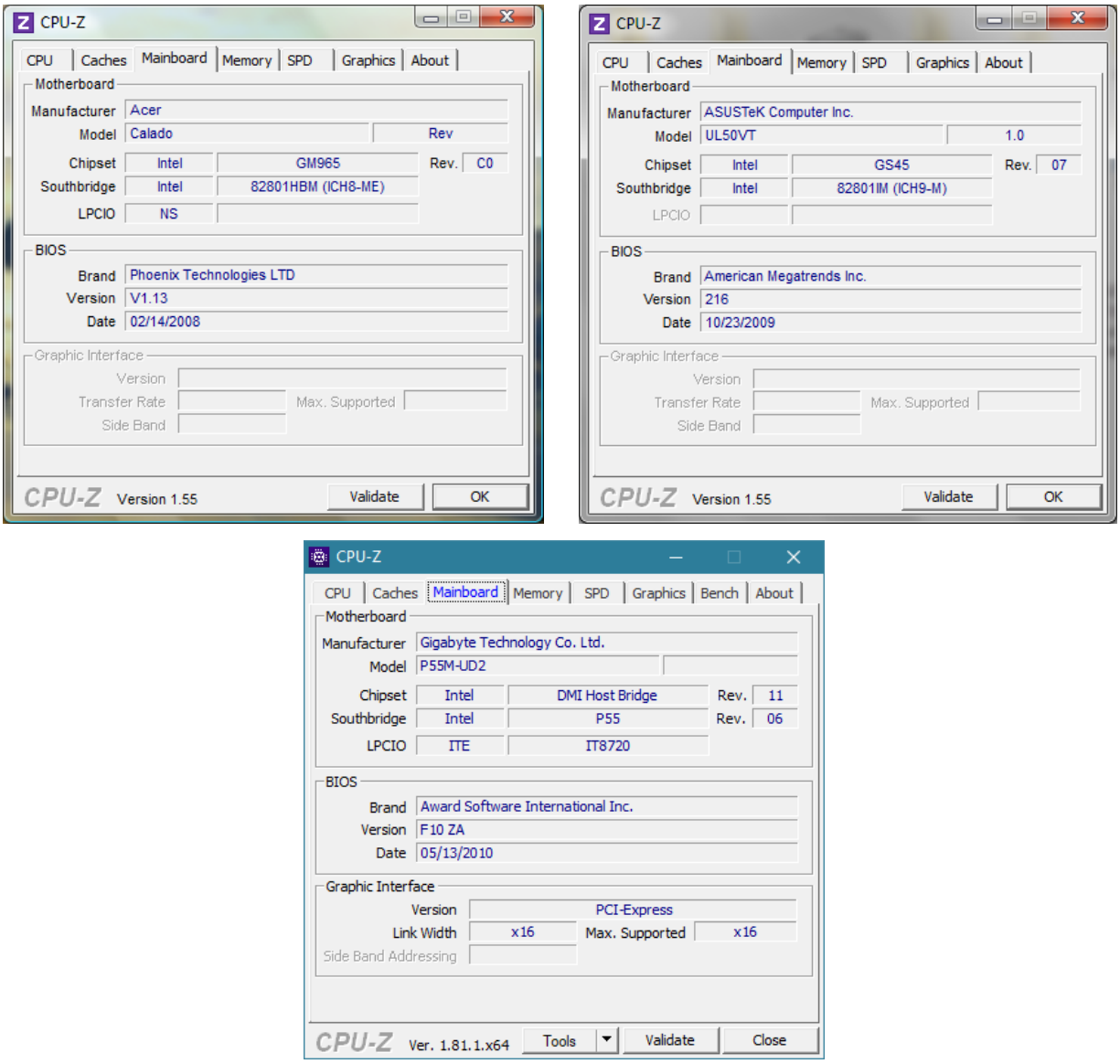
Nebezpečná chyba se objevila například ve firmwaru hybridních automobilů Toyota Prius (v letech 2004–05, v moderních automobilech totiž taky tepe počítačové srdce). Při běhu na benzinový motor se občas stalo, že se motor náhle zastavil při rychlosti něco mezi 50 a 100 km/h. U automobilu BMW 745i (u některých kusů) docházelo k „násilnému“ zastavení vozidla, pokud bylo v nádrži méně než třetina objemu pohonných hmot.

Na začátku roku 2009 doporučila firma Seagate svým zákazníkům (pevný disk Seagate Barracuda 7200.11), aby provedli upgrade firmwaru tohoto disku. Bohužel tento upgrade obsahoval chybu, která způsobila trvalé zablokování přístupu k disku. Pro firmu Seagate byla tato chyba poměrně drahá (všem postiženým uživatelům zaplatila odškodné).

Někdy je třeba upgradovat firmware (BIOS), ale měli bychom vědět, že tato operace s sebou nese jisté nebezpečí. Při upgradu může dojít k poškození dat, operačního systému, ale v nejhorším případě i k poškození zařízení (jak jsme četli o odstavec výše). Tedy postup je takový (předpokládáme upgrade BIOSu základní desky):

- zálohujeme vše, co je nám drahé (hlavně data),
- zjistíme přesný typ základní desky (ideálně některým nástrojem, který dokáže dobře zjistit podrobnosti o našem hardwaru, třeba CPU-Z od firmy CPUID[2] (vidíme na obrázku 2.4),
- najdeme stránky výrobce základní desky a tam zadáme takto zjištěný typ desky,
- na stránkách výrobce si najdeme přesný postup upgradu, tento postup se u různých výrobců (a taky desek) může lišit, postup je dobré si někam poznamenat nebo lépe vytisknout (tento postup taky obvykle najdeme v dokumentaci k základní desce),
- dále většinou následuje stáhnutí souboru s aktualizací (často mívá příponu .ROM), uložení na USB flash disk a pak dle pokynů výrobce jeho instalace.

[2]CPU-Z získáme například na http://www.cpuid.com/. Dejte pozor během instalace, budete přesvědčováni k přesměrování domovské stránky v prohlížeči a k instalaci další lišty do prohlížeče – obojí odmítněte (jsou to celkem 3 zaškrtávací políčka).



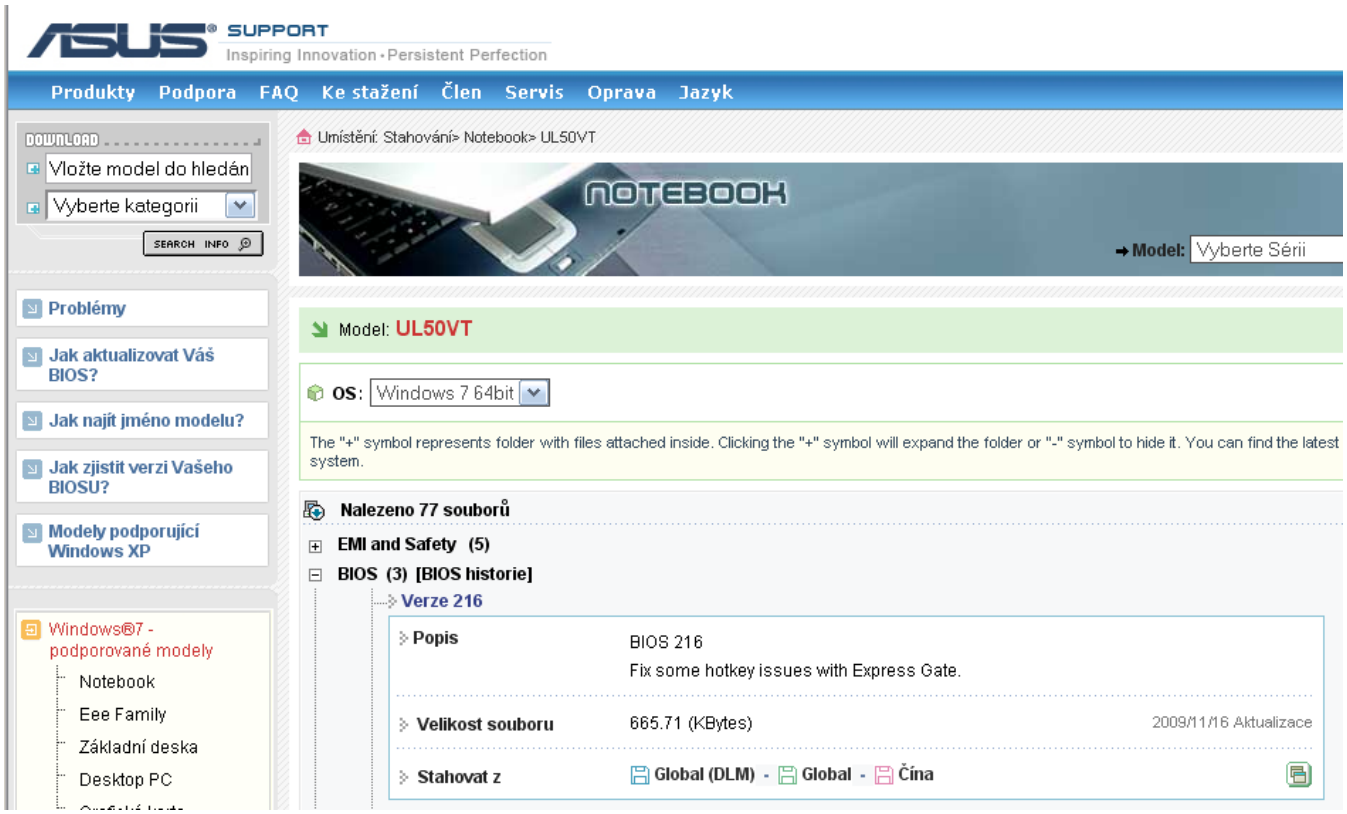
Obrázek 2.4: Zjistění typu základní desky v nástroji CPU-Z, na třech různých strojích

Příklad

Na obrázku 2.4 vidíme údaje o základní desce a BIOSu na třech různých počítačích s různými verzemi Windows. V prvním případě jde o Phoenix BIOS, v druhém případě o AMI BIOS, v třetím UEFI od Awardu.

V prvním případě jde o notebook od výrobce Acer, v druhém od výrobce Asustek (značka Asus), třetí je počítač se základní deskou od Gigabyte. Podíváme se na upgrade BIOSu pro notebook Asus. Na http://support.asus.com zadáme do vyhledávacího pole nalezený typ desky (UL50VT, ale taky můžeme postupovat výběrem v rozbalovacích menu). Jsme dotázáni na operační systém (zadáme například Windows 7 64bit), pak získáme seznam všech možných upgradů pro tento model základní desky. Vybereme to, co potřebujeme, stáhneme. Na obrázku 2.5 vidíme stránku s aktuálním upgradem.

Všimněte si položky *Jak aktualizovat váš BIOS?* – tam najdeme veškeré potřebné návody. Je možné, že mezi návody nenajdeme návod pro naši základní desku. Pak zbývá podívat se po dokumentaci, kterou jsme dostali zároveň se základní deskou (resp. s počítačem), tam postup většinou najdeme taky.



Obrázek 2.5: Stažení upgrade BIOSu na notebooku Asus

2.7 Čas v počítači

Hodiny reálného času (RTC) jsou obvod umístěný na základní desce nebo integrovaný do jižního mostu v chipsetu, který udržuje reálný čas. Na rozdíl od timeru hardwarového, který řídí časování hardwaru, hodiny reálného času počítají v „lidských“ jednotkách.

Hodiny reálného času najdeme nejen v klasických počítačích, ale prakticky v jakékoliv elektronice včetně té „obývákové“, některá zařízení bez RTC dokonce vůbec nemohou fungovat (GPS navigace). V některých architekturách naopak standardně nejsou a je třeba RTC čip dokoupit, pokud od těchto architektur chceme sledování reálného času, ať už kvůli jeho zobrazování na displeji nebo pro časová razítka do log souborů (například v Arduinu).

Pokud se jedná o samostatný čip, je to CMOS čip, který vyžaduje neustálé napájení, tedy musí být *připojen k baterii* na základní desce stejně jako CMOS čip s konfigurací BIOSu.

Jak bylo dříve uvedeno, výstup RTC se používá v pozičních systémech (včetně navigací), dále pro časová razítka do log souborů, také při komunikaci po síti (tam jsou také časová razítka, také je třeba odměřovat čekání na ohlášené doručování dat, atd.). A ovšem pro zobrazení momentálního času uživateli.

V počítači najdeme ještě několik obvodů, které souvisejí s odměřováním času. Pro nás je momentálně důležitý ještě další typ – časovač (timer, clock) pro procesor a různé sběrnice (komunikační cesty), který v pravidelných intervalech generuje signál a tím určuje rychlost procesoru nebo zařízení napojených na danou sběrnici. Digitální zařízení totiž kvůli synchronizaci potřebují takové „mávání praporkem“, nedokážou pracovat spojitě.

Kapitola 3

Rozhraní

Rychlý náhled: V této kapitole se podíváme na rozhraní, se kterými se můžeme v počítači setkat. Po vysvětlení základních pojmů se budeme věnovat rozhraním USB, Thunderbolt, FireWire, stručně starším rozhraním LPT, COM a PS/2, následují sekce o datových rozhraních pro paměťová média, o konektorech pro přenos grafiky, rozhraních pro síťovou komunikaci.

Klíčová slova: Konektor, koncovka, zástrčka, plug, zásuvka, socket, kabel, rozhraní, hardwarové rozhraní, signální rozhraní, sériová komunikace, paralelní komunikace, řadič, hot-plug, plug-and-play, USB, USB hub, OTG, Power Delivery, USB-AV, USB mass storage, mezipaměť, FireWire, Thunderbolt, LPT, COM, RS-232, PS/2, PATA, SATA, PIO, DMA, ATAPI, eSATA, mSATA, M.2, SATA Express (SATAe), SCSI, SAS, D-SUB, VGA, DVI, HDMI, DisplayPort, RJ-45 (8p8c), Bluetooth, UPnP, DLNA, Wireless HDMI.

Cíle studia: Po prostudování této kapitoly budete mít přehled v rozhraních, přes která se připojují komponenty či jiná zařízení, a to jak vně, tak i uvnitř počítače.

3.1 Pár pojmů na začátek

Abychom mohli propojovat různá zařízení nebo komponenty uvnitř zařízení, obvykle k tomu použijeme *konektor* (koncovku, zástrčku, plug) daného typu (například USB, HDMI, SATA,...). Konektor je obvykle na konci *kabelu*.

Konektor zasouváme do *socketu* (zásuvky, jack) stejného typu, tedy například USB konektor zasouváme do USB zásuvky. Socket je většinou na zařízení (například USB socket je dostupný na počítači, tabletu, apod.), ale může být taky na konci kabelu (například někdy je potřeba propojit navzájem víc kabelů).

Definice (Rozhraní)

Rozhraní (přesněji komunikační rozhraní) je obecně komponenta nebo funkčnost zajišťující komunikaci mezi zařízeními, komponentami, programy nebo čímkoliv jiným, co potřebuje komunikovat. Je to obecný pojem, který může být chápán dvěma způsoby:

- *hardwarové rozhraní* – určuje, jak vypadá konektor, zásuvka a kabel, například kolik má být v kabelu vodičů, jak mají být v konektoru a plugu uspořádány, jaký je celkový tvar a velikost konektoru apod.,

- *signální rozhraní (protokol)* – určuje, jak konkrétně se vodiče na komunikační cestě používají, například co se kterým vodičem přenáší, jak je reprezentovaná logická 0 a logická 1, atd.

Hardwarové a signální rozhraní jsou dvě různé věci a mohou být různě kombinovány. Například pro totéž signální rozhraní USB 3.0 existuje několik různě vypadajících druhů konektorů a plugů (tedy hardwarových rozhraní) – typ A, typ B, mini-USB, micro-USB, C. A naopak v konektoru/plugu USB typ C může být zavedeno signální rozhraní USB verze 3.1 (což většina uživatelů automaticky očekává), ale může to být i starší verze 3.0 či dokonce 2.0.

Rozhraní mohou být *sériová* nebo *paralelní* – sériové rozhraní posílá data patřící k sobě sekvenčně (jeden bit za druhým), paralelní rozhraní posílá vždy určitý blok dat o stanovené velikosti („slovo“) najednou (podle šířky komunikační cesty). Sériová rozhraní jsou univerzálnější, snadněji rozšiřitelná a dobře programovatelná, proto se v mnoha oblastech přechází od paralelních k sériovým rozhraním. U paralelních rozhraní je totiž právě kvůli nutnosti synchronizace a jasnému oddělení „dávek“ příliš komplikované přecházet na vyšší frekvence.

Poznámka

Ale pozor, to neznamená, že sériová rozhraní pracují s jediným vodičem – obvykle bývá uzemění (zem), případně po některém vodiči, nebo spíše dvojici vodičů, jsou přenášena přímo data, zatímco po jiných se přenášejí servisní informace, synchronizační signál apod., taky mohou být některé vodiče vyhrazeny pro napájení připojeného zařízení (například u USB).

Rozhraní mohou být buď *univerzální* nebo *specializovaná* – jsou určena pro některý konkrétní účel. Například rozhraní USB je univerzální (přes ně připojujeme různé typy zařízení – paměťová zařízení, monitory, fotoaparáty, mobilní telefony, také je používáme pro nabíjení zařízení, atd.).

Naproti tomu třeba HDMI je určeno pro přenos multimediálních dat a pro jiné účely není vhodné. Rozhraní SATA, přes které většinou připojujeme pevné disky, taky není univerzální, ale je určeno pro připojení paměťových zařízení.

Definice (Řadič)

Řadič (controller) je obecně komponenta řídící činnost jednoho nebo více zařízení. Obvykle to znamená, že když komunikujeme se zařízením, ve skutečnosti se „bavíme“ právě s jeho řadičem a ten pak zprostředkovává naše požadavky dále.

Řadič je většinou čip nebo součást čipu – integrovaný obvod, s nízkoúrovňovým programem (vpodstatě firmware). Svůj řadič má například procesor (procesorový řadič, je součástí čipu procesoru), pevný disk (většinou je na desce tištěných spojů přímo na pouzdře disku), operační paměti jsou řízeny paměťovým řadičem (dnes je obvykle součástí procesoru, u starších počítačů byl součástí severního mostu čipové sady základní desky), tiskárna, atd.

Poznámka

Řadiče mohou být jednoduché nebo složité, podle toho, jaké zařízení řídí. Například pokud srovnáme řadiče běžného pevného disku a SSD (v obou případech se jedná o úložné zařízení, s oběma se zachází víceméně stejně), řadič pevného disku patří spíše k jednodušším, kdežto řadič SSD je složitější a plní více funkcí,

přičemž kvalita řadiče má velký vliv na kvalitu SSD včetně rychlosti zápisu a životnosti. Jako řadiče SSD se dnes obvykle používají čipy s architekturou ARM, které známe jako procesory v mobilech.

Definice (Hot-plug a plug-and-play)

Rozhraní má vlastnost *hot-plug* (také hot swapping), pokud se k němu dají připojovat zařízení za chodu (tj. není nutné připojovat před zapnutím počítače). Například USB je typu hot-plug.

Rozhraní má vlastnost *plug-and-play* (Plug&Play), pokud podporuje zjednodušenou softwarovou instalaci zařízení, tj. při prvním připojení zařízení není nutné složitě konfigurovat hardwarové propojky, nastavovat komunikační adresy apod. Dnes už naprostá většina zařízení je typu plug-and-play.

Technologie Plug&Play (autoři: společnosti Intel, Microsoft, Compaq) představuje možnost zjednodušeného rozpoznání a konfigurace zařízení *při prvním zapojení* tohoto zařízení. Výrobce přidá k řadiči zařízení obvody, které provedou automatickou konfiguraci a ve spolupráci s operačním systémem umožní vybrání vhodných zdrojů (IRQ, DMA, I/O porty, adresy BIOSu, o nich později v této kapitole).

Účelem je usnadnit uživateli instalaci nového zařízení a redukovat fyzické zásahy do hardwaru (přepínače, propojky, přiřazování čísel IRQ apod.), případná uživatelská konfigurace (pokud je nutná), probíhá softwarově a co nejjednodušeji.

Hot-plug je něco trochu jiného – zařízení s podporou hot-plug je možné připojit a nechat správně rozpoznat *za běhu systému*. Tato funkce se využívá při každém, nejen prvním použití. V praxi to znamená, že například hot-plug monitor nemusíme zapínat ještě před zapnutím počítače, ale i kdykoliv později, podobně hot-plug USB flash disk lze připojit kdykoliv za běhu systému.

Takové zařízení, které není hot-plug, musí být připojováno pouze do vypnutého počítače nebo je nutné počítač po jeho připojení restartovat – to je typické například pro většinu rozšiřujících karet.

Poznámka

Mnohá zařízení mají obě tyto vlastnosti (například zmíněný USB flash disk), některá jen jednu z nich. U některých zařízení jsou tyto vlastnosti natolik praktické, až si často ani nedokážeme představit, že by nebyly podporovány, ať už na straně zařízení, nebo na straně počítače či operačního systému. Například Windows NT v prvních verzích (3.x) nepodporovaly vlastnost plug-and-play u žádného zařízení.

Rozšiřující karty určené do sběrnice PCI Express (pokud ještě nevíte, o co se jedná, dozvíte se v následující kapitole) jsou obvykle plug-and-play, ale obvykle nejsou hot-plug. Pokud například chceme připojit novou grafickou kartu do této sběrnice, musíme počítač vypnout, zasunout kartu a počítač znovu zapnout, za běhu to nejde. Ve skutečnosti existuje možnost, aby PCI Express fungovala hot-plug, ale musí to „umět“ hardware i software, typicky to potřebujeme na některých serverech, které není možné „jen tak“ vypnout. Tato problematika je diskutována například zde:

https://electronics.stackexchange.com/questions/208767/does-pcie-hotplug-actually-work-in-practice

3.2 USB

USB (Universal Serial Bus) je univerzální *sériové* rozhraní určené pro připojení externích komponent k základní desce. Má vlastnost hot-plug i plug-and-play.

V některém z čipů na základní desce obvykle máme hlavní řadič rozhraní USB a *topologie* (tj. způsob, jak je co propojeno či co s čím komunikuje) je založena na *USB rozbočovačích* (USB hub), které zároveň

pracují jako opakovače (repeater – zesiluje signál). Celkově tvoří jakousi *stromovou strukturu* – hlavní řadič si představme jako kořen stromu, ze kterého vycházejí větve, tyto větve se dále větví, atd. Každé větvení je zajištěno právě USB hubem – jednu původní větev rozděluje na dvě „podvětve“ (viz logo).

V počítači je dnes obvykle více USB hubů (zapojených do stromové struktury), aby bylo možné vyvést více USB zásuvek. USB huby jsou buď na základní desce, nebo slouží k vyvedení rozhraní ze skříně, nebo jsou například součástí monitorů, klávesnic či jiných periferních zařízení.

3.2.1 Vlastnosti a verze USB

Rozhraní USB má tyto vlastnosti (pozitivní i negativní):

+ velmi nízká cena řadičů,
- závislost přenosové rychlosti na rychlosti a vytížení procesoru (CPU),
+ vysoká kompatibilita zařízení (obvykle není problém s ovladači USB zařízení),
+ kapacita až 127 zařízení (ovšem včetně rozbočovačů).

Verze 1.0 z roku 1995 se pro svou chybovost neuchytila. První rozšířenou verzí byla 1.1 (1998). Přehled současných používaných verzí najdeme v tabulkách 3.1 a 3.2.

Rychlost je závislá nejen na limitech technologie, ale také na množství rozbočovačů na cestě, délce kabelu a také na konstrukci samotného zařízení včetně kvality řadiče. V tabulce 3.1 jsou ke každé verzi uvedeny jak teoretická propustnost, tak i reálná rychlost. Všimněte si jednotek – určitě každý ví, jaký je rozdíl mezi bitem (b) a Bytem (B) – že 1 B = 8 b.

Ozn.	**Verze 1.1**		**Verze 2.0**		**Verze 3.0**		**Verze 3.1**	
	propustnost	rychlost	propustnost	rychlost	propustnost	rychlost	propustnost	rychlost
Low Speed	1.5 Mbit/s (0.2 MB/s)	0.18 MB/s	1.5 Mbit/s (0.2 MB/s)	0.18 MB/s				
Full Speed	12 Mbit/s (1.5 MB/s)	1.1 MB/s	12 Mbit/s (1.5 MB/s)	1.1 MB/s				
High Speed			480 Mbit/s (60 MB/s)	30 MB/s				
Super Speed					4.8 Gbit/s (600 MB/s)	300 MB/s		
Super Speed+							10 Gbit/s (1200 MB/s)	800 MB/s

Tabulka 3.1: Starší verze USB – teoretická propustnost a reálná rychlost, původní označení

Rychlost je ovlivněna více faktory, například:

- přenos je řízen procesorem, a pokud je procesor zaneprázdněn jinými úlohami, rychlost přenosu klesá,
- čím víc zařízení je připojeno přes USB, tím nižší je rychlost,
- čím víc hubů je na cestě, tím nižší je rychlost,
- rychlost přenosu taky závisí na činnosti řadiče připojeného zařízení (jak rychle dokáže vysílat či přijímat).

Jinými slovy – údaje v sloupci s rychlostí jsou nejen nižší než údaje v sloupci s propustností, ale dokonce i tak je musíme brát jako orientační, ve skutečnosti může být ještě nižší.

	USB 3.0	USB 3.1	USB 3.2	USB 4	USB 4 v2
= verze po uvedení 3.1	3.1 Gen 1	3.1 Gen 2	–	–	–
= verze po uvedení 3.2	3.2 Gen 1	3.2 Gen 2	3.2 Gen 2×2	–	–
Označení po uvedení USB4		USB4 Gen 2×1 USB4 Gen 2×2	USB4 Gen 3×1 USB4 Gen 3×2	USB4 Gen 4	USB4 v2
Propustnost	5 Gb/s	10 Gb/s	20 Gb/s	40 Gb/s	80 Gb/s

Tabulka 3.2: Verze USB podle nejnovějšího označení

K tabulce 3.2: označení s křížkem – USB4 a×b: Pokud je „b“ jednička, v USB-C se používá jen jedna strana, pokud je to dvojka, tak obě strany (dual lane).

Poměrně velký rozdíl mezi teoretickou propustností a reálnou rychlostí je prakticky u všech rozhraní, netýká se jen USB. Je dán například nutností synchronizace přenosu (jedno ze zařízení, master, čas od času posílá synchronizační informaci, samotný přenos obsahuje přidané synchronizační prvky v podobě úpravy signálu kódováním do podoby, kdy se snižuje pravděpodobnost chyb, čímž se zvyšuje množství přenášených dat).

Kromě dat se přenášejí i metadata (data o datech), například řadiče se navzájem domlouvají, kdy který pracuje, že má být něco přeneseno, kam to má být přeneseno (pracuje se s adresami), občas se objevují chybová hlášení, atd.

3.2.2 Hardwarové rozhraní

Jak určitě každý ví, přes USB kabel lze nejen přenášet data, ale také *napájet* (ano, i USB flash disk funguje na elektřinu, potřebuje napájení). Napájení i data jsou vedena *společným kabelem* (samozřejmě zvlášť v různých vodičích), uvnitř jde o kroucené dvojlinky (pro redukci přeslechů).

U USB 2.0 se používají dva datové vodiče a dva napájecí, takže v kabelu máme celkem čtyři vodiče. U specifikace USB 3.0 je počet datových vodičů navýšen – z důvodu zpětné kompatibility zůstávají dva pro nižší rychlosti, pro rychlost SuperSpeed byly přidány další čtyři (takže 6 datových a 2 napájecí, celkem 8 vodičů v kabelu).

Pro USB konektory a zásuvky existuje více různých *standardů*, viz obrázek 3.1. Dále se můžeme setkat s proprietárními řešeními, která jsou specifická a nekompatibilní s ostatními – zřejmě jde o snahu výrobců nepouštět tu část trhu, kterou dříve získali.

Na prvním „řádku“ obrázku je USB plug (konektor) a zásuvka (jack) typ A, vlevo pro USB 2.0, vpravo pro USB 3.0. Liší se počtem pinů (kovových proužků napojených na vodiče uvnitř kabelu) – jak bylo výše uvedeno, v typu A pro USB 2.0 jsou čtyři piny, v typu A pro USB 3.0 je jich více – tedy verze rozeznáme je podle počtu pinů. Navíc je jack a konektor pro USB 3.0 většinou uvnitř modrý (většinou, ale ne vždy). Typ A je zpětně plně kompatibilní, tedy konektor 2.0 zasuneme do zásuvky 3.0 a naopak. Typ A se používá především pro datová média (externí disky, USB flash disky apod.) nebo pro případy, pro které nejsou určeny jiné typy, včetně různých redukcí na jiná rozhraní.

Na druhém „řádku“ obrázku je USB jack a konektor typ B. Počet pinů je stejný jako u typu A, ale jsou u verze 2.0 uspořádány do dvojic nad sebou na menší šířku (proto jiný tvar), u verze 3.0 bylo toto uspořádání technicky nemožné a bylo nutno přidat „nástavbu“ pro přebývající piny. Z toho důvodu je

zpětná kompatibilita omezená – starší konektor sice zasuneme do novější zásuvky, ale naopak to nejde – konektor typu B verze 3.0 nezasuneme do zásuvky B verze 2.0, protože se tam prostě nevejde. Typ B se typicky používá pro připojení tiskáren a scannerů.

Dole vidíme variantu Micro-B. Tyto konektory se používají pro malá mobilní zařízení. Z důvodu malých rozměrů je zpětná kompatibilita omezená – konektor a jack pro verzi 3.0 jsou širší, jak vidíme.

Obrázek 3.1: USB konektory a zásuvky typu A a B[1]

Existuje taky varianta Mini-B, ale setkáme se s ní pouze u starých mobilních zařízení, je postupně nahrazována Micro- variantou (proto ani neexistuje Mini-B verze 3.0, je pouze do verze 2.0). Mini-B se dále používá na některých síťových zařízeních (někteří výrobci používají Mini-B jako konzolový port, případně pro připojení datového média například kvůli aktualizaci firmwaru), a také u některých prototypovacích desek jako je Arduino.

	Konektor							
	2.0				3.0			
Zásuvka	Std A	Std B	Micro A	Micro B	Std A	Std B	Micro A	Micro B
2.0 Std A	✓				✓			
2.0 Std B		✓						
2.0 Micro AB			✓	✓				
2.0 Micro B				✓				
3.0 Std A	✓				✓			
3.0 Std B		✓				✓		
3.0 Micro AB			✓	✓			✓	✓
3.0 Micro B				✓				✓

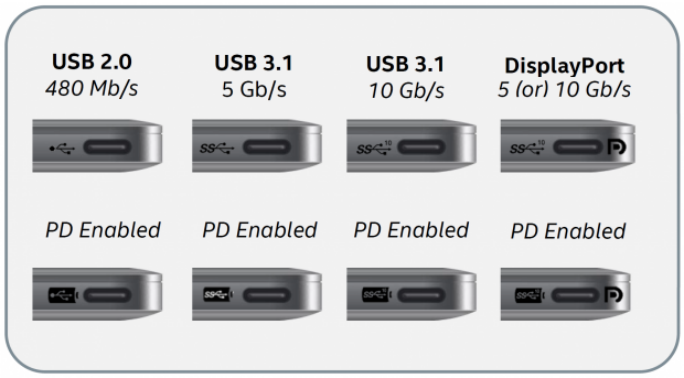
Tabulka 3.3: Kompatibilita typů a verzí USB

V tabulce 3.3 je naznačena vzájemná kompatibilita různých zásuvek a konektorů. Zásuvka Micro-AB slouží k zapojení konektorů typu Micro-A a Micro-B.

[1] Zdroj: http://www.l-com.com/what-is-a-usb-cable (upraveno)
[2] Zdroj: http://www.golem.de/news/ueberblick-wir-entwirren-den-kabelsalat-um-usb-typ-c-1601-118667.html

Čím dál víc rozšířený je *USB typ C*. Je o něco menší než typ A, ale zároveň o něco bytelnější než typ Micro-B, takže na jednu stranu je použitelný i pro malá mobilní zařízení a na druhou stranu je o něco méně náchylný ke zlomení. Má oválnější úzký tvar.

Výhodou je symetričnost (je reverzibilní) – konektor můžeme vložit do jacku otočený „nahoru“ nebo „dolů“, je to jedno. Důvodem je, že všechny piny jsou v jacku dvakrát, na každé ze dvou delších hran je celá sada pinů (ovšem to platí, pokud komunikujeme pouze rychlostí pro USB 3.1).



Obrázek 3.2: USB typ C, různé druhy[2]

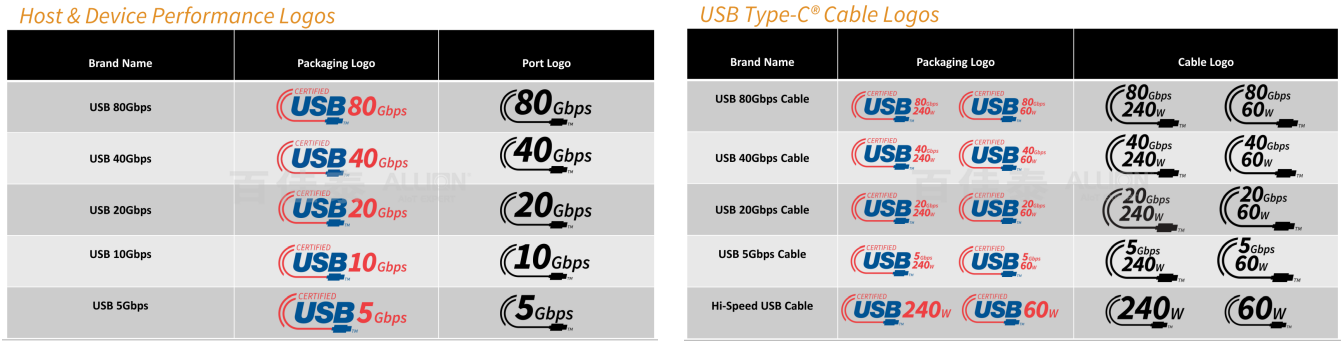
Na obrázku 3.2 vidíme jack USB typ C. Hardwarové rozhraní je ve všech případech stejné, ale (jak bylo výše uvedeno) hardwarové a signální rozhraní jsou dvě různé věci. Do téhož USB typ C může vést signál USB verze 3.1 (což většina uživatelů očekává), ale může to být třeba jen USB 2.0, což je typické zejména (ale nejen) pro levnější malá mobilní zařízení. Na tom, jaké signální rozhraní je použito, závisí rychlost přenosu dat, kterou máme na obrázku taky naznačenu.

Podle čeho poznáme, jak rychle nám přes USB typ C „potečou“ data? Tento konektor je příliš malý, aby se dalo spoléhat na rozlišení barvou, proto se používají ikonky, jak vidíme na obrázku 3.2.

- Pokud je na ikoně pouze logo USB, pak jde jen o USB 2.0.
- Pokud je u loga zkratka „SS“ (a u ní žádné číslo), je to USB 3.0.
- Pokud je u loga zkratka „SS“ a číslo 10, je to USB 3.1.
- Písmeno „D“ (zkratka z DisplayPort) navíc znamená, že je možné připojit externí monitor, který „umí“ USB typ C a standard DisplayPort.
- Pokud je u zásuvky symbol blesku (není na obrázku), je podporováno i signální rozhraní Thunderbolt verze 3 (viz dále).

Ikony u spodní řady zásuvek na obrázku vypadající jako baterie navíc znamenají vylepšenou podporu napájení, k tomu se také v následujícím textu vrátíme.

Ovšem změna je život, jak tvrdí USB Implementers Forum (USB IF), tedy s USB 4 v2 se značení změnilo. Na obrázku 3.3 vlevo je značení na zařízení, vpravo značení kabelu.



Obrázek 3.3: Nová loga USB[3]

Existuje také interní varianta hardwarového rozhraní USB, na to se podíváme na přednášce.

[3] Zdroj: https://www.allion.com/news-center/developerdays_usb_4_version2/

Poznámka

Takže pozor – pokud výrobce označí výrobek jako USB 3.1 Gen 1, očekávejte nižší rychlosti než u Gen 2. Je to netypický a nelogický krok, uživatele to jen mate. Všimněte si, že na obrázku 3.2 máme oba prostřední sloupečky označeny USB 3.1, ale zjevně je jeden z nich Gen 1 a ten druhý Gen 2 (poznáme podle čísla 10 u loga, znamená teoretickou propustnost 10 Gbit/s).

Podobně pokud vidíte jen USB 3.2 bez označení generace, není jisté, o co jde, rozdíl v propustnosti je 5 Gb/s vs. 10 Gb/s vs. 20 Gb/s, což je celkem hodně.

Úkol

Pokud máte možnost, najděte si kabel s USB konektorem a případně zásuvku (ideálně více různých typů) a prohlédněte si ho (nerozebírejte). Srovnejte konektory různých typů a všimněte si jejich velikosti, tvaru, označení, umístění a velikosti pinů uvnitř. Všimněte si, že v zásuvce/konektoru typu A a B jsou piny širší, protože tam je dostatek místa, kdežto u typu Mini- nebo Micro-B jsou piny užší a mnohem víc natěsnané, navíc piny po stranách jsou víc vysunuté ven.

Pokud zůstaneme u USB 2.0 (čtyři piny), pak „vnitřní“ piny u typu A a Mini/Micro variant (piny číslo 2 a 3) jsou určeny pro data, „vnější“ vysunuté piny (číslo 1 a 4) pro napájení – konkrétně na pravý je vedeno +5 V, levý je zem. Pokud se jedná o pětipinové rozhraní, pak pin číslo 4 slouží jako indikační pro technologii OTG (viz dále), pin 5 je zem.

U verze USB 3.0 a vyšší je osm až jedenáct pinů. Čtyři jako u nižších verzí, ze zbývajících jsou čtyři určeny pro data (dva pro každý směr) v rychlosti SuperSpeed, zbytek je dodatečné napájení a zem.

Poznámka

Pozor – nezáleží jen na konektoru a zásuvce. Abychom mohli využít vyšší rychlost, musíme mít i „rychlejší“ kabel. Každý (vlastně skoro každý) pin musí být napojen na některý z vodičů v kabelu, jinak by nemohl být použit. Pokud je v konektoru víc pinů (verze 3.0 a vyšší), musíme mít také v kabelu víc vodičů.

Další informace

Problémy s tím, jaký kabel USB-C koupit, jsou popsány například na
https://www.cnews.cz/konektorove-peklo-usb-c-thunderbolt-3-alternate-mode.

Na začátku jsme si o USB řekli, že lze připojit maximálně 127 zařízení. Proč zrovna tolik a jak se to v praxi projevuje? Víme, že USB huby tvoří strom, v jehož kořeni je hlavní řadič (USB Host Controller). Ke každému rozbočovači mohou být obvykle připojena dvě zařízení, z toho jedno může být opět rozbočovač. Každé zařízení, resp. jeho řadič (včetně rozbočovačů) potřebuje adresu, a adresa zabírá 7 bitů $\Rightarrow 2^7 = 128$ adres (z toho jedna pro hlavní řadič, takže zbývá 127 adres).

Každý rozbočovač potřebuje svůj řadič, a tento řadič má svou adresu. Tím vypotřebujeme poměrně hodně adres (pro celý strom rozbočovačů), a dále každé koncové zařízení potřebuje *minimálně* jednu adresu. Pokud má koncové zařízení více funkcí, pro každou funkci potřebuje jednu adresu (protože struktura uvnitř zařízení je z USB hubů, včetně řadičů). Například:

- multifunkční zařízení: rozbočovač + tiskárna + skener + fax,
- monitor s kamerou a reproduktory: rozbočovač + webkamera + reproduktory + grafika.

Pokud má takové zařízení ještě USB jacky pro připojení externích zařízení, počet dál narůstá. Je třeba brát v úvahu i to, že mnohá zařízení berou energii přes USB, a také rozbočovače je nutné napájet, a i pro USB cesty jsou napájecí limity.

Poznámka

Pokud chceme využívat zařízení s konektorem vyšší verze, kterou náš počítač nepodporuje (resp. takovým konektorem, který nemáme), je možné použít rozšiřující kartu do slotu PCIe (PCI Express).

3.2.3 Napájení

Zařízení s nižšími nároky (fotoaparát, modem apod.) jsou napájena přímo z USB kabelu, zařízení s vyššími nároky (scanner) mají vlastní napájení. Zařízení napájená z USB kabelu však nemusejí být správně rozpoznána, pokud jsou zcela vybitá (např. fotoaparát) – platí pro USB do verze 2.0 včetně. Od verze 3.0 by měl být tento problém vyřešen.

Verze 2.0 a 3.0 umožňují napájet následovně:

- USB 2.0 – napětí 5 V, proud do 0.5 A = 500 mA, výkon 2.5 W,
- USB 3.0 – napětí 5 V, proud do 0.9 A = 900 mA, výkon 4.5 W.

Verze 3.1 přinesla novinky i v oblasti napájení – funkci *Power Delivery*. Standard Power Delivery určuje 5 různých profilů – viz tabulku 3.4. Ne všechny nutně musí být zdrojem napájení podporovány.

USB 3.1 Profil	Napětí	Proud	Výkon	Určení
Profil 1	5 V	2.0 A	10 W	Smartphony
Profil 2	5 V	2.0 A	10 W	Tablety
	12 V	1.5 A	18 W	
Profil 3	5 V	2.0 A	10 W	Ultrabooky
	12 V	3 A	36 W	
Profil 4	5 V	2.0 A	10 W	Notebooky,
	12 V	3.0 A	36 W	hub
	20 V	3.0 A	60 W	
Profil 5	5 V	2.0 A	10 W	Počítače,
	12 V	5.0 A	60 W	dokovací
	20 V	5.0 A	100 W	stanice

Tabulka 3.4: Profily napájení v USB 3.1

Podporu funkce Power Delivery (a tedy možnost napájení i náročnějších zařízení) poznáme podle ikony vedle USB zásuvky – na obrázku 3.2 na straně 28 to je spodní řada, ikona je podbarvena symbolem baterie.

Oproti USB 3.0 jsou také ve specifikaci USB 3.1 různé *úsporné režimy*. Řadiče kontrolují, zda zařízení komunikuje, a připojené nekomunikující zařízení má téměř nulový odběr, tedy šetříme energii.

Poznámka

Pokud se jedná o kabel určený pouze pro nabíjení, obvykle v něm bývají jen vodiče pro napájení (tedy například pro verzi 2.0 pouze dva). Poznáme to podle toho, že v konektoru kabelu vidíme pouze dva piny, místo ostatních jsou jen prázdné pozice.

3.2.4 Srovnání verzí

Nejdřív srovnání verzí 2.0 a 3.0. O rozdílu v přenosových rychlostech už víme (tabulka 3.1, str. 25), ale ve skutečnosti je rozdílů více. Vyšší rychlosti bylo dosaženo navýšením počtu vodičů, dále změnami v kódování signálu (jinak – úsporněji – jsou reprezentována přenášená data) a změnou přenosové frekvence, což má nepříjemný důsledek ve zkrácení dosahu: zatímco signál pro USB 2.0 lze přenést až na vzdálenost 5 m, pro USB 3.0 to jsou jen 3 m. Takže pro USB 3.0 nemůžeme používat tak dlouhé kabely jako pro USB 2.0.

Poznámka

To neznamená, že při využití USB 2.0 můžeme použít pětimetrový kabel nebo pro USB 3.0 třímetrový. Vzdálenost 5 m pro přenos signálu v sobě zahrnuje nejen kabel, ale celou cestu od hlavního řadiče na základní desce až k rozhraní na připojeném zařízení. Pokud výrobce počítače vytvořil uvnitř počítače zbytečně složitou a dlouhou strukturu USB hubů, mohl z této cesty podstatnou část ukrojit.

Na druhou stranu – jestliže použijeme aktivní kabelové spojky (které dokážou zesílit signál), může být naopak dosažená vzdálenost vyšší.

Další rozdíl je ve způsobu komunikace: v USB 2.0 se data přenášejí v tzv. *poloviční duplexu* (tj. komunikace sice může být obousměrná, ale v jeden okamžik lze přenášet jen jedním směrem, komponenty se musejí střídat), v USB 3.0 již v *plném duplexu* (lze komunikovat v obou směrech současně). To lze díky tomu, že vodiče pro rychlost SuperSpeed jsou vyhrazeny zvlášť pro oba směry.

Nyní k verzi 3.1. Propustnost je dvojnásobná z důvodu zvýšení přenosové frekvence z 5 GHz na 10 GHz, mění se také kódování na fyzické vrstvě. Počet vodičů v kabelu zůstává stejný, ale požadavky na provedení kabelu jsou přísnější (útlum, stínění vodičů apod.), přesto je kvalita signálu garantována jen do délky 1 m (bez opakovače), a to jen tehdy, když máme kabel certifikovaný pro USB 3.1 Gen 2.

Standard USB-AV představený v USB 3.1 definuje přenos multimediálních dat (hudba, video), což byl logický krok, protože propustnost zařízení se blíží HDMI 1.4 (ale DisplayPort je na tom ještě lépe).

Co nového je ve verzi 3.2? Kromě toho, že byly přejmenovány předchozí dvě verze (takže když kupujete zařízení nebo kabel „USB 3.2“, zkontrolujte si generaci), nejvyšší garantovaná teoretická propustnost (20 Gb/s) bude pouze přes USB-C se dvěma sadami pinů (tj. opravdu se musí jednat o USB-C plugy/jacky s oběma řadami pinů aktivními napojenými na vodiče, ideálně kabel certifikovaný pro USB 3.2 Gen 2×2). To znamená, že zdvojnásobení rychlosti je způsobeno především tím, že se používá dvojnásobek vodičů. Pokud použijeme jiný typ plugu/jacku, popřípadě USB-C kabel, který nemá „správný“ počet vodičů, klesne propustnost na úroveň některé starší generace.

USB verze 4 je ještě daleko (uvidíme), zatím je jasné, že dojde k sloučení s rozhraním Thunderbolt (viz dále), čemuž odpovídá i teoretická propustnost 40 Gb/s. Konektor zůstane USB-C.

3.2.5 Související technologie

USB mass storage je standard pro paměťová zařízení připojovaná k USB definovaný sdružením USB Implementers Forum. V operačním systému, který tento standard podporuje, lze USB zařízení připojit jako diskovou jednotku (diskový oddíl). Jde obvykle o USB flash disky, externí disky s USB rozhraním nebo některé digitální fotoaparáty. Reálně jde o sadu komunikačních protokolů, které jsou v daném operačním systému podporovány.

Problematika USB mass storage souvisí také s možností *bootovatelnosti USB zařízení* (tj. máme USB

flash disk s operačním systémem a chceme, aby při startu počítače najel OS z tohoto flash disku místo z pevného disku v počítači). Aby mohlo být USB zařízení použito tímto způsobem, musí být příslušný standard podporován také BIOSem, resp. základní deskou. Podporu poznáme snadno v BIOS Setup, podle seznamu možných paměťových zařízení pro zavedení systému. Problém bývá především s USB 3.0 a vyššími, přes které u některých (starších) základních desek BIOS/UEFI nepodporuje bootování.

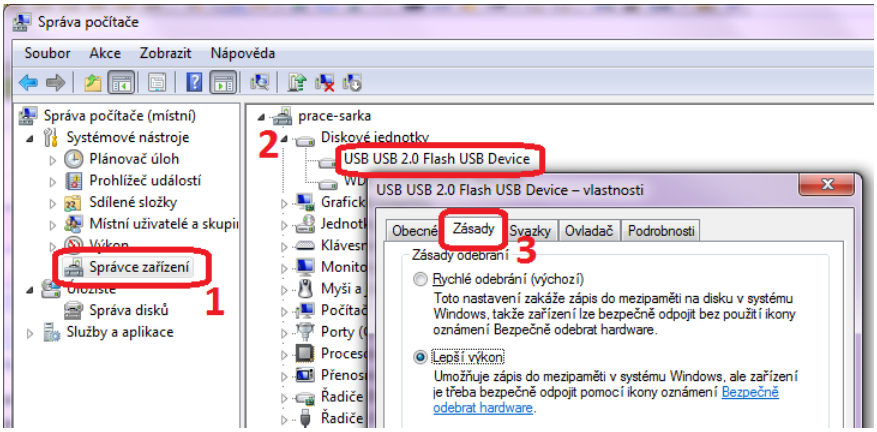
Mezipaměť je vyrovnávací paměť používaná při zápisu do (pomalé) flash paměti. Protože je přístup do flash paměti pomalejší, data se při kopírování na zařízení nejdřív uloží do (rychlejší) mezipaměti a pak se postupně odesílají do zařízení, přičemž zapisující proces není dále zdržován. Tento postup je možné potlačit, například novější verze Windows standardně mezipaměť nepoužívají. Co je výhodnější?

- Pokud používáme mezipaměť, jeví se kopírování dat na zařízení rychlejší. Řadič flash paměti má dost času na optimalizaci uspořádání dat v čipu, to může mít vliv i na životnost flash paměti (k tomu se dostaneme později). Naopak nevýhodou je, že USB zařízení nemůžeme jen tak bez softwarového odpojení vysunout, protože v mezipaměti by ještě mohla zůstat data, která se zatím nestačila přesunout na zařízení. Takže je nutné klepnout na ikonu *Bezpečně odebrat hardware*.
- Pokud nepoužíváme mezipaměť, můžeme zařízení kdykoliv vysunout bez softwarového odpojení. Nevýhodou je, že data se kopírují přímo na zařízení, což zdržuje.

Ve Windows se většinou mezipaměť nepoužívá, v jiných operačních systémech obvykle ano (Linux, MacOS).

Postup

Pokud se rozhodneme upřednostnit bezpečnost a rychlost před pohodlím a budeme chtít ručně „bezpečně odpojovat“ USB flash disk, k příslušnému nastavení se ve Windows dostaneme následovně:



- připojíme příslušné zařízení,
- spustíme *Správce zařízení* (buď přes *Ovládací panely* nebo přes *Správu počítače* (pravým tlačítkem myši na ikonu *Počítač*, vybereme *Spravovat*, případně taky přes *Ovládací panely*),
- ve stromě položek vpravo najdeme *Diskové jednotky* a v jejich seznamu pak to zařízení, které chceme konfigurovat, poklepeme (zobrazí *Vlastnosti*),
- na kartě *Zásady* zvolíme druhou možnost.

Jaké mohou být důsledky, pokud USB flash disk *používající mezipaměť* vytáhneme bez softwarového odpojení?

- To, co chceme uložit na flash disk, se nejdřív „kompletuje“ v mezipaměti a až po určité době je fyzicky zapsáno na flash disk. Pokud flash disk vysuneme před dokončením finálního zápisu, přijdeme o data. V nejhorším případě mohou být data zapsána jen částečně, paměť by byla v nekonzistentním stavu a flash disk můžeme tak akorát zformátovat.
- V USB konektoru jsou piny napájecích a datových vodičů velmi blízko u sebe. Pokud vysouváme flash disk, který je ještě pod proudem (protože jsme ho softwarově neodpojili), může dojít ke zkratu mezi

napájecím a datovým vodičem a hrozí „usmažení“ flash čipu. Pravděpodobnost něčeho takového je sice malá, ale nikoliv nulová.

Ve Windows Vista/7/Server 2008 a novějších je ještě jedna zrada, která se týká právě problému s napájením. Tyto verze Windows totiž bývají často konfigurovány tak, že při softwarovém odpojení nedojde k odpojení napájení. Flash disk sice pak není viditelný ani přístupný, ale neustále zůstává pod proudem. Důvodem je údajně rychlejší znovupřipojení paměti, ale stojí za úvahu, zda je něco takového nutné (když něco odpojujeme, zřejmě to v nejbližších minutách nebudeme znovu připojovat). To můžeme upravit úpravou v registru (ale pozor, každou úpravu v registru bychom si měli dvakrát rozmyslet a vždy postupovat s největší opatrností). Najdeme klíč `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\usbhub\hubg`, klepneme pravým tlačítkem myši v pravém podokně, vytvoříme novou hodnotu typu DWORD (32bit), nazveme *DisableOnSoftRemove* a nastavíme na „1“.

USB OTG (USB On-The-Go) je pojem v poslední době hodně zmiňovaný především u mobilních zařízení. Oficiálně se jedná o dodatek ke standardu USB přidávající možnost peer-to-peer komunikace mezi periferními zařízeními přes rozhraní USB bez nutnosti intervence počítače či jiného výpočetního systému (Fire-Wire to umožňuje v základu, ale USB to ve specifikaci přímo nemá). Ve skutečnosti se nejedná přímo o peer-to-peer komunikaci, ale umožňuje dotyčnému zařízení chovat se jako Master nebo Slave podle potřeby (jinak bývá Masterem počítač s plnohodnotným USB řadičem) – Master iniciuje a řídí komunikaci, má hlavní řídicí roli a např. také poskytuje napájení.

V čem je přínos OTG pro běžného uživatele, proč je dobré tuto vlastnost vyžadovat? Běžná malá mobilní zařízení (smartphony, tablety, MP3 přehrávače, navigace apod.) bývají většinou osazena levným malým USB mikrořadičem (USB microcontroller) místo plnohodnotného (který máme ve větších zařízeních), a mikrořadiče bez OTG nedokážou pracovat v režimu Master, mohou být jen Slave. V praxi to znamená, že sice není problém připojit mobil k počítači (mobil je Slave, počítač je Master), ale zato je problém připojit externí disk (příp. s konektorovou redukcí na Mini USB) k mobilu – externí disk je vždy Slave, a pokud mobil dokáže být jen Slave, kdo bude Master?

Proto pokud máme mobilní zařízení s podporou OTG, můžeme k němu připojit periferie (které jsou vždy Slave). Například k mobilu či tabletu s podporou OTG připojíme externí disk, fotoaparát propojíme s tiskárnou, apod.

Při použití OTG by teoreticky mohl nastat jeden problém – pokud jako Master dokážou pracovat obě propojovaná zařízení (například počítač a mobil s OTG), musejí se dohodnout, kdo bude Master, protože tím může být jen jeden. Řeší se to jednoduše na „fyzické“ úrovni: musí být použit „asymetrický“ kabel, na jednom konci je konektor Mini-A, na druhém Mini-B, přičemž obě zařízení mají jack Mini-AB (který je kompatibilní s oba konektory). ⇒ Master je na začátku ten, na jehož straně je konektor Mini-A. Ale to neznamená, že bude Masterem po celou dobu připojení, to se dohodne pomocí protokolu HNP – Host Negotiation Protocol.

Další informace

- http://hw.cz/teorie-a-praxe/art2569-vysla-specifikace-usb-30.html
- http://hw.cz/Teorie-a-praxe/Dokumentace/ART327-USB—Universal-Serial-Bus—Popis-rozhrani.html

3.2.6 Protokoly vyšší úrovně

Jakékoliv univerzální rozhraní, nejen USB, přenáší různé druhy dat, a je vhodné nějak rozlišit, jaký typ dat je „uvnitř“. V případě USB rozlišujeme různé *třídy*, a příslušnost USB zařízení do určité třídy určuje, jaký typ dat bude přenášen, a tedy s jakým protokolem vyšší úrovně bude komunikovat.

Když připojujeme USB zařízení, toto zařízení nejdřív vyšle signál právě s touto informací. Jedná se o číslo, například:

- číslo 08h (hexadecimálně, stejně jako následující) znamená zařízení typu Mass Storage, tj. paměťové médium, sem většinou patří také digitální fotoaparáky s přímým přístupem k paměti,
- číslo 07h znamená tiskárnu nebo CNC stroj,
- číslo 03h představuje zařízení třídy HID (Human Interface Device) pro jednoduchou komunikaci s člověkem, což znamená klávesnici, myš, touchpad apod.,
- číslo 0Bh představuje čtečku SD karet,
- číslo 0Dh používají čtečky otisků prstů, atd.

Zatímco standard 2.0/3.1/3.2/atd. určuje fyzikální parametry signálu (co na kterém pinu, frekvence apod.), třída stanovuje, jak mají vypadat posílaná data co se týče jejich vnitřní struktury. Jinak se komunikuje s tiskárnou a jinak s klávesnicí.

V případě, že chceme přes USB-C připojit zobrazovací zařízení (typicky externí monitor) a přenášet audiovizuální signál, měli bychom se zajímat o *alternate modes*. Alt mode je vlastně nástavba USB-C umožňující přenášet „cizí“ signál určený původně do jiného typu konektoru. Nejběžnější Alt mody jsou:

- DisplayPort a Thunderbolt, probrány už dříve, jsou indikovány písmenem D nebo bleskem u USB zásuvky,
- MHL
- DisplayPort
- HDMI
- VirtualLink – pro zařízení virtuální reality.

Například pokud chceme k notebooku připojit přes USB-C externí monitor, který má HDMI rozhraní, potřebujeme předně, aby USB-C rozhraní na notebooku umělo Alt mode HDMI, dále kabel musí mít na jedené straně USB-C a na druhé straně HDMI konektor.

Další informace

https://www.cnews.cz/konektorove-peklo-usb-c-thunderbolt-3-alternate-mode

3.3 Další univerzální rozhraní

3.3.1 FireWire

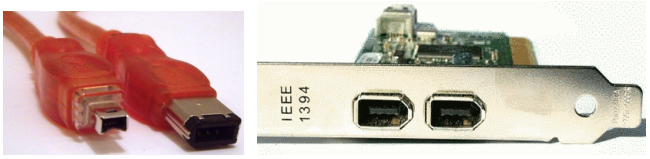
FireWire (I.Link, Lynx, IEEE 1394) je univerzální sériové rozhraní určené pro připojení externích komponent k základní desce, přímá konkurence USB. Pochází od společnosti Apple, asi proto čekalo toto rozhraní velmi dlouho na podporu ve Windows, to je zřejmě jeden z důvodů jeho menší rozšířenosti.

Splňuje vlastnosti hot-plug i plug-and-play. Slouží k připojení až 63 zařízení, délka kabelu je nejvýše 4,5 m, v nejnovějších specifikacích až 100 m. Komponenty pro implementaci FireWire jsou dražší než komponenty pro implementaci USB, ale na druhou stranu je považováno za spolehlivější. Další vlastnosti:

- používá jinou signalizaci než USB, poměr mezi teoretickou propustností a reálnou rychlostí je lepší,
- přenos není tak výrazně „dávkován“ jako u USB, datový tok je stabilnější,
- k jednomu hlavnímu řadiči je možno připojit více zařízení ve stromové struktuře (podobně jako u USB) nebo jako daisy-chain (řetězit jedno za druhým),
- propustnost: podle specifikace S400 až 393 Mbit/s, specifikace S800 až 786 Mbit/s (doporučení IEEE 1394b), specifikace S3200 až 3144 Mbit/s v délce kabelu až 100 m,
- možnost napájení připojeného zařízení (napájecí vodiče v tomtéž kabelu jako datové), přičemž rozvod napájení je kvalitnější než u USB 2.0, spotřeba do 15 W při odběru do 45 mA.

Ve srovnání s USB vychází reálná rychlost FireWire novějších verzí někde mezi USB 2.0 a 3.0.

FireWire používá několik typů *konektorů*, které se liší především počtem signálních okruhů (4, 6 nebo 9) a následně rozměry. Na obrázku 3.4 vidíme první dva typy a také část rozšiřující karty s dvěma 6okruhovými konektory.



Obrázek 3.4: FireWire se 4 a 6 signálními okruhy[4]

Konektor FireWire je vizuálně odlišitelný od konektoru USB-A především svým tvarem (USB-A je trochu „hubenější“, seříznutí a vnitřní struktura jsou jiné). Jednotlivé specifikace FireWire jsou zpětně kompatibilní, ale konektory nemusejí zcela odpovídat, někdy je nutné použít redukci.

Rozhraní FireWire bylo vybráno sdružením HANA (High Definition Audio-Video Network Alliance) jako standard pro AV přenosy, dodnes se používá u některých profesionálních videokamer a dalších zařízení pro přesuny zejména multimediálních dat, protože je spolehlivé, se spojitým tokem data a dostatečně rychlé, není až tak závislé na zatížení procesoru jako USB. FireWire také zvítězilo jako systémová sběrnice v mnoha vojenských zařízeních (stíhačky F-22 Raptor či F-35 Lightning II) a v automobilovém průmyslu (Customer Convenience Port). Implementace FireWire se objevují také v některých síťových zařízeních.

3.3.2 Thunderbolt

V únoru 2011 představila společnost Intel nové univerzální sériové rozhraní s názvem *Thunderbolt* (TB). Toto rozhraní bylo vyvíjeno pod původním názvem *Light Peak* (představeno v roce 2008, v experimentální podobě) coby optické rozhraní, ale s Thunderboltem se momentálně setkáme většinou v podobě pro metalické (měděné) kabely. Optika je dražší, ale kabely mohou být delší. Podobně jako u USB, je u tohoto rozhraní výhodou i cena, protože Intel nevyžaduje placení licenčních poplatků, které by prodražovaly implementaci.

Zatímco u verze 1 byla teoretická propustnost 10 Gbit/s na jeden kanál, přičemž se používaly dva nezávislé kanály (tj. 20 Gbit/s celkem), u verze 2 je propustnost na kanál 1 Gbit/s s možností až 20 kanálů, tj. celková také 20 Gbit/s (jen ji lze lépe škálovat). Verze 3 zdvojnásobila teoretickou propustnost na 40 Gbit/s, což je stejně jako u USB 4. Reálná rychlost je cca o třetinu nižší než teoretická. Thunderbolt 4 zůstává na rychlosti 40 Gbit/s.

Zařízení lze s využitím kabelů řetězit (vlastně stejným způsobem jako FireWire, daisy-chain nebo strom), maximálně 7 zařízení, komunikuje se v plném duplexu.

Metalickým kabelem je možné zařízení i napájet (do 10 W, což je víc než USB 3.0 a méně než FireWire nebo USB 3.1 Power Delivery v profilu 2 nebo vyšším).

[4] Zdroj: http://en.wikipedia.org/wiki/FireWire

Signální a hardwarové rozhraní je zajímavý hybrid. Signál je kódován stejným způsobem jako DisplayPort (tj. používá se protokol DisplayPort, o tomto rozhraní se dozvíme více dál v této kapitole), je přenášen přes sběrnici PCI Express (o té více v kapitole o základních deskách, tato sběrnice se používá i pro přenos signálu z grafických karet), a jack/konektor je jednoho z těchto dvou typů:

- pro Thunderbolt 2 se používá jack/plug Mini DisplayPort, od Mini DisplayPortu s původním významem ho poznáme podle ikony blesku ,
- pro Thunderbolt 3 byl vybrán konektor USB typ C, u něj musí být ikona blesku .

Obě verze jsou zpětně kompatibilní, jen je třeba použít adaptér (TB3 to TB adapter), protože se rozhraní fyzicky liší.

Další vlastnosti Thunderbolt verze 3 a 4 souvisejí s konektorem USB-C, včetně případné podpory PowerDelivery (samozřejmě pokud ji výrobce použije) a příslušných napájecích profilů.

Pro Thunderbolt se dají koupit kabely buď aktivní, nebo pasivní. Pasivní jsou levnější, ale deklarovaná propustnost je garantována jen na krátkou vzdálenost (obvykle 50 cm), delší pasivní kabely (do cca 2 m) dovolují pracovat na poloviční rychlosti. Aktivní metalické TB kabely se obvykle dají sehnat také v délce do 2 m (ale mohou být delší), ovšem i při délce nad půl metru zaručují propustnost 40 Gb/s.

Dají se sehnat i optické TB kabely s délkou do 60 m, ovšem za několikanásobně až řádově vyšší cenu při stejné propustnosti, a připojené zařízení se přes ně nedá napájet, což musíme řešit zvlášť.

Když kupujeme kabel pro Thunderbolt, nestačí se dívat po kabelu s koncovkami USB-C, opravdu musí jít o kabel certifikovaný pro Thunderbolt, jinak „padáme“ na úroveň USB (v nejhorším případě USB 2.0), v určitých případech nebude přenos vůbec fungovat. Pouze u kabelů přímo určených pro Thunderbolt je jistota, že podporují Alternate modes (viz kap. 3.2.6), obrácené tvrzení neplatí (tj. pokud kabel není přímo určený pro Thunderbolt, *může* podporovat *některé* Alternate modes). Ovšem za certifikovaný TB kabel si připlatíme.

Poznámka

Na vývoji tohoto rozhraní spolupracovala společnost Apple (ostatně, použitý konektor Mini DisplayPort byl již v minulosti vyvinut Applem), a jako součást podílu si vymínila výhradní práva na Thunderbolt po dobu dvou let. Tedy po tuto dobu jsme se s Thunderboltem setkávali jen na počítačích značky Apple. Po uplynutí dvou let se toto rozhraní již začalo objevovat především na ultraboocích různých výrobců, ovšem jen takových, které mají procesor od Intelu.

Další informace

- http://www.macrumors.com/2011/02/24/thunderbolt-details-emerge-bus-power-mini-displayport-and-more/
- http://www.intel.com/content/www/us/en/io/thunderbolt/thunderbolt-technology-developer.html
- https://blog.startech.com/post/thunderbolt-3-the-basics/
- http://www.zive.cz/clanky/thunderbolt-jeden-port-ktery-chce-vladnout-vsem/sc-3-a-156032/default.aspx

3.3.3 Srovnání univerzálních rozhraní

V tabulce 3.5 je srovnání různých technologií. Je třeba brát na vědomí, že každá technologie je optimalizována pro určitý způsob použití a určité požadavky.

Rozhraní	Teor. propustnost	Max. počet zařízení	Max. délka přenosové cesty
USB 2.0	až 60 MB/s	127	5 m
USB 3.0	až 600 MB/s	127	3 m
USB 3.1	až 1200 MB/s	127	1 m
USB 3.2	až 2400 MB/s	127	1 m
FireWire 1600	až 200 MB/s	63	4,5 m
FireWire 3200	až 400 MB/s	63	4,5 m
Thunderbolt	až 1200 MB/s	7	3 m
Thunderbolt 2	až 2400 MB/s	7	3 m
Thunderbolt 3, 4	až 4800 MB/s	7	3 m
SATA II	až 375 MB/s	1	1 m
SATA 3.0	až 750 MB/s	1	1 m
Gbit Ethernet	až 125 MB/s	1	100 m

Tabulka 3.5: Srovnání některých parametrů univerzálních a jiných rozhraní

Zatímco u USB je důležitá kvantita (počet připojených zařízení k jednomu hlavnímu řadiči) a rychlost, kdežto délka kabelu není většinou až tak podstatná, u síťové technologie Gigabit Ethernet (poslední řádek) je tomu přesně naopak. FireWire je optimalizováno na plynulý datový tok, má lepší poměr mezi teoretickou propustností a reálnou rychlostí a navíc stabilnější mechanismus napájení, tedy opět se poněkud hůře přímo srovnává. U SATA vypadají hodnoty propustnosti dost špatně, ale pro připojení klasických pevných disků stačí, protože samotný pevný disk nedokáže pracovat rychleji, a délka kabelu je také dostačující. Thunderbolt je v současné době nejprogresivnější, jeho nevýhodou je zatím malé rozšíření.

Poznámka

Při výběru vhodného rozhraní se neohlížejte jen na teoretickou propustnost. Nejdůležitějším kritériem je to, abychom dokázali připojit to, co připojit potřebujeme (kompatibilita), pak samozřejmě reálná rychlost s přihlédnutím ke konkrétnímu využití, délka kabelu apod.

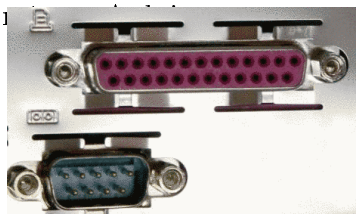
Co se kabelu týče, i na něm záleží, jak rychle data potečou. Kabel je obvykle určen pro určitou verzi rozhraní, a pokud chceme plně využít vlastnosti rozhraní vyšší verze, potřebujeme příslušný kabel. Navíc se někdy nemusí vyplatit přílišné šetření na kabelech – nekvalitní kabel může mít horší stínění (tj. signál je více rušen) a tenčí vodiče (tj. větší odpor), což znamená, že se dosah zkrátí.

3.4 Paralelní a sériové porty, PS/2

Paralelní port LPT (Centronics) se dnes již téměř nepoužívá, u novějších počítačů je často ani nenajdeme. Původně byl určen pro paralelní jednosměrnou komunikaci s tiskárnami, později byly přidány další možnosti včetně obousměrné komunikace (bi-directional).

Paralelní port LPT se dodnes používá pro připojení hodně starých tiskáren (novější se připojují přes USB), ale lze je také využít pro připojení paměťových médií či zálohovacích jednotek, rychlost je pro tento případ zcela dostačující – ovšem dnes jen málokterý počítač je tímto rozhraním vybaven, tedy se jedná spíše o historii.

Sériový port COM (RS-232, V.24, služební port) se vyskytuje ve dvou velikostech (menší má 9 pinů, větší 25 stejně jako paralelní). Více se setkáváme s menším, který se taky označuje jako D-SUB DB-9 (větší je D-SUB DB-25), taky se objevuje nepřesné označení UART, zej



Obrázek 3.5: Paralelní a sériové rozhraní

Dříve se RS-232 používal pro připojení myši a modemu, v současné době se s ním setkáme hlavně v průmyslu a obchodu (tiskárny etiket a čárových kódů, měřicí systémy, kontrolní mechanismy), pak u některých síťových zařízení, především z důvodu velmi snadného naprogramování. Ve skutečnosti jde obvykle o varianty RS-232, a to RS-422 (pro spolehlivé přenosy i na větší vzdálenosti v zarušeném prostředí) a RS-485 (v průmyslu). V poslední době se RS-232 používá u různých prototypovacích destiček, jako jsou například různé moduly pro Arduino, taky z důvodu snadného řízení datového toku.

Obrázek 3.6: Konzolový kabel RS-232 na RJ-45

U síťových zařízení, případně serverů, se používá jako služební rozhraní. Tato zařízení obvykle nemají klávesnici a obrazovku, ale zato mají 9pinový port RS-232. Takže potřebujeme kabel se sériovým rozhraním na obou stranách a notebook se sériovým rozhraním, případně se dá použít redukce z RS-232 na USB, protože notebooky dnes většinou port RS-232 nemají (pak je ale nutné mít na notebooku zapnutou emulaci sériového rozhraní).

Jindy (třeba u Cisca) je situace ještě komplikovanější. na síťovém zařízení je hardwarový port ve tvaru běžného síťového portu (RJ-45) označený jako *console*, jenže signálově funguje jako RS-232. Takže je třeba mít speciální kabel, na jehož jednom konci (počítač) je RS-232 a na druhém (síťové zařízení) je RJ-45 – obrázek 3.6, případně opět použijeme redukci z RS-232 na USB. Více v Počítačových sítích.

Poznámka

25pinové sériové rozhraní je velmi podobné paralelnímu. Rozeznáme je od sebe podle orientace – na straně počítače má sériové rozhraní „kolíky“, kdežto paralelní má „dírky“, jak vidíme na obrázku 3.5. Další možnost záměny je s rozhraním VGA (uvidíme dále).

Obrázek 3.7: Redukce PS/2 na USB

Rozhraní PS/2 bylo původně určeno pro počítače firmy IBM, ale rozšířilo se i u dalších výrobců. U počítačů se ještě někdy používá pro připojení myši a klávesnice. Pro počítače bez PS/2 portu, kde chceme použít PS/2 zařízení, jsou k sehnání redukce PS/2⇒USB (vidíme na obrázku 3.7).

Rozhraní je typu Mini-DIN (DIN rozhraní je kulaté s piny), má 6 pinů a na desktopu je najdeme velmi snadno – barevně je vyznačeno rozhraní pro klávesnici (fialové) a myš (zelené), příp. jack, který je napůl fialový a napůl zelený.

Další informace

- https://allpinouts.org/
- https://vyvoj.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html
- https://www.secomp.cz/default.asp?show=pcpkonektor

Poznámka

Nadále budeme používat pojmy paralelní rozhraní (paralelní přenos) a sériové rozhraní (sériový přenos) v obecnějším slova smyslu. Na začátku kapitoly byly tyto pojmy vysvětleny, tady si je připomeneme.

Pod pojmem *paralelní rozhraní* chápeme takové rozhraní, přes které jsou přenášena data paralelně (víc bitů téhož slova najednou). To se může zdát jako výhodnější a rychlejší, nicméně je třeba si uvědomit, že při paralelním přenosu je třeba neustále provádět synchronizaci přenášených dat. Data jsou přenášena po dávkách (celá dávka bitů najednou), je třeba důsledně od sebe jednotlivé dávky oddělovat a vždy celou dávku a následně posloupnost dávek seskládat do výsledného proudu dat.

Sériové rozhraní přenáší data víceméně ve streamu (proudu), jednotlivé bity (vhodně kódované) následují jeden za druhým. To však neznamená, že by v takovém kabelu byl pouze jeden vodič – najdeme vodiče nejen datové, ale i servisní, zem, napájecí apod., navíc se obvykle pro přenos jednoho streamu dat používá dvojice vodičů (přenášený signál je určen rozdílem potenciálů těchto vodičů). Existence více dvojic datových vodičů obvykle znamená nezávislý přenos ve více kanálech paralelně; z pohledu komponent to vypadá jako několik různých (virtuálních) komunikačních cest, třeba zvlášť cesty pro různé směry.

Jak dále uvidíme, většina takových rozhraní, která dříve existovala v paralelní formě, byla vystřídána sériovou variantou. To platí například u rozhraní pro pevné disky (PATA → SATA, SCSI → SAS), ale třeba i u typů sběrnic na základní desce a síťových rozhraní. Sériová rozhraní totiž mají jednu důležitou výhodu: není třeba neustále provádět synchronizaci přenášených dat. Zrychlovat se dá například takto:

- volbou jiného materiálu pro vodiče, konektor apod.,
- změnou kódování (tj. způsobu mapování bitů na přenášený signál),
- změnou vysílací frekvence (rozšířením na vyšší frekvence), apod.

V každém případě u paralelních rozhraní narážíme na hranici danou potřebou neustálé synchronizace. Především při zvýšení frekvence odesílání a příjmu dat začne docházet k přenosovým chybám, změny v kódování mohou zvýšit složitost synchronizace a tím paradoxně přenos zpomalit, atd.

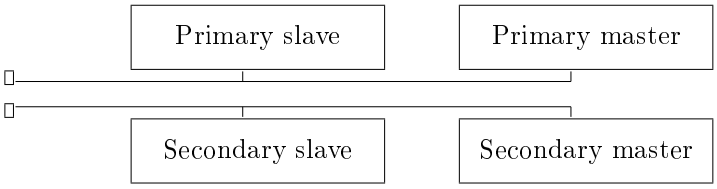
3.5 Datové rozhraní paměťových médií

3.5.1 PATA a SATA

ATA (zkratka z AT Attachment, kde AT znamená Advanced Technology – pro počítače PC AT) je letitý standard pro připojení paměťových zařízení s paralelním přenosem. Po zveřejnění standardu SATA (Serial ATA) byl původní standard ATA přejmenován na PATA (Parallel ATA, kolem roku 2003). V dnes vyráběných počítačích najdeme spíše jen SATA; u starších počítačů je použit PATA.

Pro starý standard *ATA-1* (1986) se vžilo označení rozhraní *IDE* (Integrated Drive Electronics), což je poněkud nepřesný název, ve skutečnosti označuje typ použitých kabelů.

V případě PATA mohou být na jednom kabelu maximálně dva disky, přičemž se vyžaduje hardwarová konfigurace (tj. není to řešení plug-and-play): na jednotlivých discích připojených na tentýž kabel musí být *propojky* nastaveny tak, aby jeden z nich byl *master* a druhý *slave*. Pokud je připojen jen jeden disk, musí být propojkami označen jako *single* (na mnoha discích však tento stav neexistoval, disk se nastavoval jako master). Na obrázku 3.8 je naznačen způsob připojení více disků.



Obrázek 3.8: Připojení PATA disků

Definice (Režim komunikace – PIO a DMA)
Zařízení připojená přes PATA nebo SATA komunikují buď v režimu *PIO* (Programmed Input/Output) nebo v režimu *DMA* (Direct Memory Access). V režimu PIO veškerou komunikaci řídí procesor. V režimu DMA je procesor využíván pouze na začátku, zadá příkaz k práci s daty a zbytek komunikace (samotný přenos) zajišťuje řadič pevného disku. O DMA kanálech se budeme bavit v sekci 5.9.2 na straně 122.

ATA není jenom o tom, jak má být formován signál, obsahuje definici sady nízkoúrovňových ATA příkazů, kterými se dá komunikovat s připojenou komponentou. Jsou to příkazy jako například identifikace zařízení, čtení z určitého sektoru na disku či zápis na něj pomocí PIO či DMA, reset zařízení, práce s vyrovnávací pamětí, kontrola sektorů, atd.

Další generace standardu ATA vždy nějakou vlastnost a příkazy přidávají (například ATA-3 zavádí monitorovací technologii S.M.A.R.T., ATA/ATAPI-4 přichází se standardem ATAPI).

Definice (Signální rozhraní ATAPI)
Standard *ATAPI* (ATA Packet Interface) je rozhraní k ATA na vyšší úrovni; běží nad původním nízkoúrovňovým rozhraním ATA, přidává nové vlastnosti, usnadňuje přístup k zařízení (BIOS, operační systém a procesy dokážou přes ATAPI snadno detekovat připojené disky a jejich vlastnosti). ATAPI přineslo podporu mechanik pro práci s paměťovými médii (dříve to bylo řešeno emulací mechanik jako pevných disků), a také standardizaci logických přenosových metod (DMA, Ultra DMA), S.M.A.R.T., karet (compact flash).

SATA (Serial ATA) je sériové rozhraní k paměťovým zařízením (pevným diskům, optickým mechanikám a dalším mass storage zařízením). Na rozdíl od PATA umožňuje připojení pouze jednoho zařízení na jedno rozhraní (nelze připojit zařízení „na sběrnici“), proto obvykle máme v počítači více SATA rozhraní a není třeba rozlišovat single/master/slave disky.

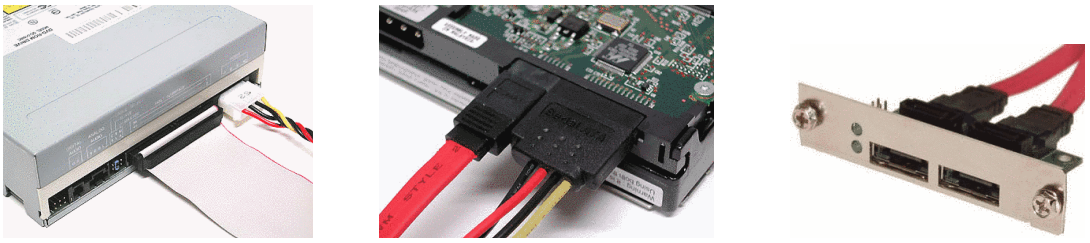
V současné době existuje více verzí SATA:

- původní specifikace (SATA 1) znamená datovou propustnost téměř 1.5 Gb/s (pozor, bity) s rychlostí přenosu dat cca 150 MB/s,
- SATA II by mělo mít propustnost dvojnásobnou (3 Gb/s, rychlost přenosu dat cca 300 MB/s),
- další generace (SATA 3.0) znamená propustnost až 6 Gb/s, rychlost přenosu dat 600 MB/s,
- SATA 3.1 přinesla některá funkční vylepšení jako např. mSATA (viz dále),
- SATA 3.2 přidala M.2 (viz dále) a přišla s vylepšením SATA Express (zrychlení napojením na PCI Express, stejně jako Thunderbolt), propustnost až 16 Gbit/s (reálná téměř 2 GB/s),
- SATA 3.3 (rok 2016) a SATA 3.4 (2018) přidávají další nové vlastnosti jako je podpora SMR disků (viz dále) a lepší monitorování a řízení napájení připojených zařízení.

V obchodech obvykle narazíme přímo na údaj o rychlosti (SATA 3 Gb/s nebo SATA 6 Gb/s). Předpokládá se kabel o délce max. 1 m a rozhraní podporuje hot-plugging.

Kabel a konektor SATA je na první pohled rozeznatelný od PATA. Je užší (protože komunikace není paralelní, ale sériová), obsahuje jen 7 vodičů (z toho 4 datové, přenos je plně duplexní – obousměrný), je lépe izolovaný (silnější izolační vrstva) a konektory na kabelu jsou jen na koncích (jeden kabel = jedno zařízení). Rozdíl vidíme na obrázku 3.9: vlevo je připojený PATA disk (široký bílý kabel je datový PATA/IDE kabel,

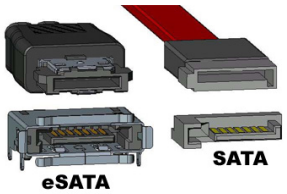
[5] Zdroj: http://www.cubeternet.com



Obrázek 3.9: PATA, SATA a eSATA[5]

vedle je napájecí kabel s barevnými vodiči), uprostřed je připojený SATA disk (červený kabel je datový, vedle opět napájení). SATA kabel bývá většinou červený nebo modrý, ale může být i jiný, barva není standardizovaná. Komunikuje se přes PIO nebo DMA (resp. rychlejší Ultra DMA).

eSATA je speciální verze SATA určená pro připojení externích disků (tj. vyvádí rozhraní SATA ven ze skříně počítače), konektor a jack eSATA jsou robustnější než SATA, jsou navrženy pro časté odpojování, jak vidíme vpravo na obrázku 3.10. Obvykle se směrem dovnitř skříně počítače napojuje na SATA II, proto propustnost odpovídá SATA II.



Obrázek 3.10: Rozdíl mezi SATA a eSATA[6]

Externí disky opatřené rozhraním eSATA jsou obvykle dražší než ty, které lze připojit jen přes USB, a taky ne každý počítač toto rozhraní má (disky s rozhraním eSATA obvykle mají i USB konektor).

Specifikace eSATAp umožňuje také napájet zařízení s malou spotřebou energie, podobně jako USB (ovšem eSATAp musí být podporováno oběma stranami).

Některé operační systémy (konkrétně Windows do verze XP včetně) neobsahují ovladače pro disky SATA. Některé BIOS Setupy obsahují podporu režimu, ve kterém se SATA disk jeví operačnímu systému jako EIDE (tj. řešením může být nastavit v BIOSu řadič SATA disku do režimu „legacy PATA“, výchozí je režim standardní AHCI nebo některý typ RAID, a pak vyhledání vhodných ovladačů na Internetu). Pokud však máme BIOS Setup bez této možnosti, musíme předem (před instalací) sehnat ovladač, uložit na disketu a tuto disketu pak použít během instalace. Ovladač obvykle seženeme buď na paměťových médiích dodávaných s počítačem, a nebo na internetu na stránkách výrobce disku.

Další rozhraní, jehož základem je SATA, je *mSATA* (mini-SATA), standardizované verzí SATA 3.1. mSATA se vyznačuje především vysokou propustností (rychlostí), proto může být využíváno SSD zařízeními. Vypadá podobně jako slot sběrnice Mini PCI Express (o tom později), ale používá zcela jinou signalizaci, proto je třeba mSATA zařízení zapojit opravdu slotu označeného jako mSATA. Setkáme se s ním v některých noteboocích, ale je postupně vytlačováno rozhraním M.2.

Rozhraní M.2 je novější než mSATA (standardizováno v SATA 3.2), ale v současné době je běžnější. Vizuálně je velmi podobné (až na klíčování – umístění zářezů a plastových můstků), funkčně a dalšími vlastnostmi také, je napojeno na PCI Express. M.2 se používá většinou pro rychlejší SSD, ale také například pro Wi-fi moduly a další komponenty, které vyžadují rychlý přenos dat.

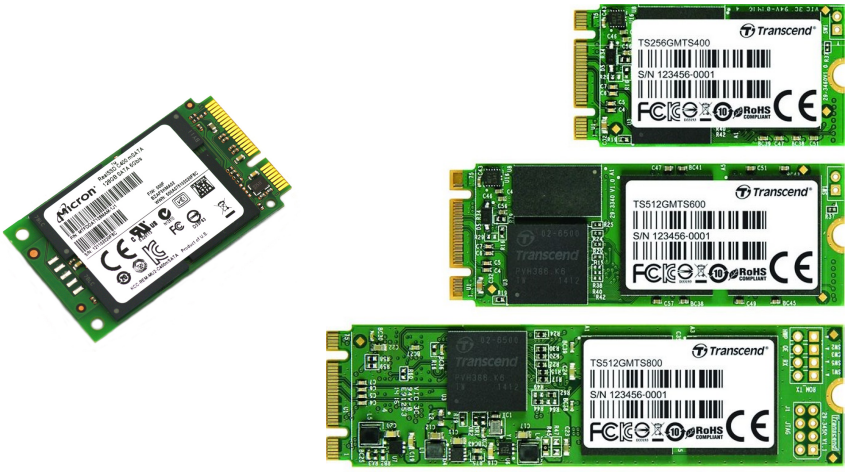
Na obrázku 3.11 je několik SSD – jeden s mSATA rozhraním a tři různých velikostí s rozhraním M.2. Všimněte si rozdílu v klíčování a uspořádání pinů.

Poznámka

Všimněte si, že klíčování je odlišné taky u různých M.2 zařízení. Obvykle se setkáváme s jedním zářezem (M-key) nebo dvěma zářezy (M+B-key), existuje také B-key. Typy M-key a B-key komunikují různým

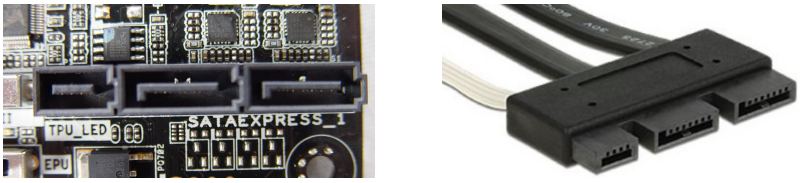
[6] Zdroj: http://www.homestead.co.uk

způsobem, nejsou zaměnitelné, proto je obvykle nejjednodušší koupit komponentu (třeba SSD nebo Wi-fi modul) s jejich kombinací M+B se dvěma zářezy, která umí oba způsoby komunikace. Podrobněji se s rozdíly seznámíme v předmětu Hardware a komunikační technologie.



Obrázek 3.11: Rozhraní mSATA (vlevo) a M.2 (tři různé velikosti)[7]

SATA Express (také SATAe) přišlo také s verzí SATA 3.2. Využívá podobný postup jako Thunderbolt nebo M.2 – napojuje se na sběrnici PCI Express, čímž dosahuje výrazně vyšších rychlostí než SATA 3.0, až 16 Gbit/s (teoretická propustnost). Je do určité míry zpětně kompatibilní – do SATAe jacku (slotu) na základní desce můžeme zasunout buď přímo jeden konektor SATAe nebo dva původní konektory SATA. Ve skutečnosti existuje víc možností, jak může SATAe zásuvka vypadat.



Obrázek 3.12: Vlevo jack SATAe na základní desce a vpravo konektor SATAe[8]

Další informace

- http://www.legitreviews.com/what-is-sata-express-and-why-it-matters_140093
- http://www.hardwaresecrets.com/article/27
- http://pctuning.tyden.cz/hardware/disky-cd-dvd-br/10999-gigabajty_na_cestach-vyzkousejte_esata_reseni
- The Essential Guide to SATA and SATAe: https://books.google.cz/books?id=PqvNBQAAQBAJ&printsec=frontcover

3.5.2 SCSI a SAS

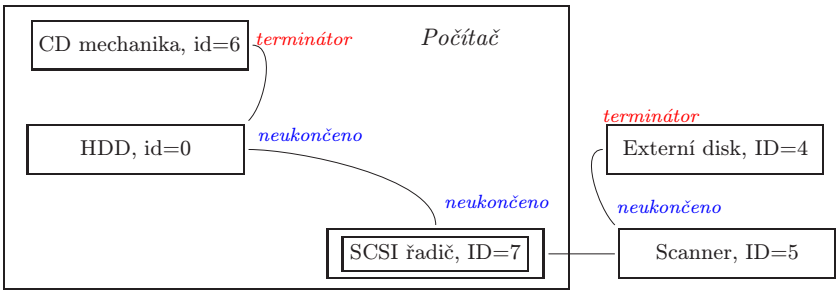
SCSI (Small Computer System Interface) je další rozhraní pevných disků (1986), je však používané i jinými typy zařízení (například SCSI scanery). Je tedy univerzálnější než ATA.

[7] Zdroj: http://www.svethardware.cz/art_doc-02CB0B602D89AD31C1257A460060978F.html, http://german.alibaba.com/product-free-img/msata-to-sata-adapter-109775658.html

[8] Zdroj: http://wccftech.com/asus-motherboard-sata-express-sata-32-interface-teased/

Existuje více standardů lišících se šířkou přenášených dat, přenosovou frekvencí, rychlostí a propustností, postupně se rozšiřovala množina SCSI příkazů. Používá se 50žilový kabel, u novějších 68žilový (také konektory jsou různé).

Lze připojit až 8 nebo až 16 zařízení, z toho jedno je řadič (každé zařízení má svou jednoznačnou číselnou adresu v rozsahu 0–7 nebo 0–15, nejvyšší adresu má řadič). Zařízení mohou být jak interní, tak externí (obvykle vede od řadiče jedna sběrnice dovnitř a jedna ven). Sběrnice musí být na „hraničních“ zařízeních ukončena *terminátory* (zakončovacími odpory), jak vidíme na obrázku 3.13.



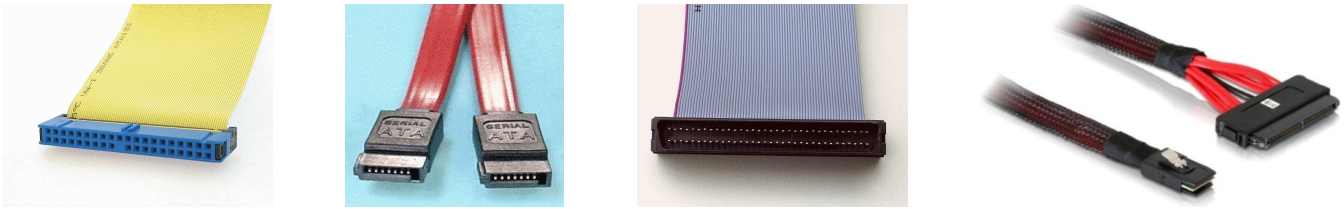
Obrázek 3.13: Síť rozhraní SCSI

Výhodou je vysoká přenosová rychlost, zapojení více zařízení rychlost neovlivní. Sada příkazů SCSI (která se používá pro komunikaci s řadičem) je mnohem rozsáhlejší než u ATA příkazů, včetně příkazů pro hlášení chyb, příkazy mohou být transportovány i po počítačové síti. Nevýhodou je vyšší cena (nejen rozhraní, také zařízení), a menší rozšířenost.

SAS (Serial Attached SCSI) je sériové rozhraní typu point-to-point, je to nástupce původního paralelního SCSI. Používá stejnou komunikační sadu jako SCSI (SCSI příkazy). Propustnost stoupá s verzemi:

- SAS-1: 3 Gb/s
- SAS-2: 6 Gb/s
- SAS-3: 12 Gb/s (rok 2013)
- SAS-4: 22.5 Gb/s (rok 2017)

Pro signály se používá vyšší napětí než u SATA, proto kabely mohou být delší (až 10 m).



Obrázek 3.14: Konektory IDE, SATA, SCSI, Mini SAS a SAS (interní)

Kabely, konektory a jacky jsou různé – podle toho, zda mají být interní nebo externí, taky existuje mini varianta. Na obrázku 3.14 zcela vpravo vidíme dva interní SAS konektory, z nichž první je Mini SAS 36pinový, druhý plný SAS 32pinový.

SAS najdeme u některých serverových pevných disků, protože je považováno za přibližně stejně rychlé jako SATA (ovšem záleží na verzi), ale spolehlivější. Důležitou předností je také možnost „exportu“ rozhraní ven, také do sítě (protokol iSCSI), a zajištění „vysoké dostupnosti“.

Další informace

- http://www.datapro.net/techinfo/scsi_doc.html
- http://www.vahal.cz/cz/podpora/technicke-okenko/rozhrani-sas.html
- http://www.tomshardware.co.uk/sas-6gb-hdd,review-31665.html

- http://www.lsi.com/solutions/Pages/SAS.aspx
- http://www.storagesearch.com/sas-art2.html

V následující tabulce vidíme srovnání různých technologií určených pro přenos dat, technologie jsou seřazeny podle teoretické propustnosti. Znovu upozorňuji, že údaje jsou relativní, ostatně srovnejte například reálnou rychlost pro USB 3.1, Thunderbolt 1 a SAS-3. Další odlišnosti (a to velmi výrazné) mohou být způsobeny implementací (naprogramováním) řadiče konkrétního zařízení, konkrétním použitím (přenosy velkých souborů nebo sekvence malých souborů, délka přenosu apod.), skutečnou délkou přenosové cesty, atd.

Rozhraní	Teor. propustnost	Rychlost	Kabel
Thunderbolt 3	40 Gbit/s		3 m (měď), 60 m (optika)
SAS-4	22,5 Gbit/s		10 m
Thunderbolt 2	20 Gbit/s	2.44 GB/s	3 m (měď), 60 m (optika)
USB 3.2	20 Gbit/s	2.42 GB/s	1 m
SATA 3.2	16 Gbit/s	1.97 GB/s	1 m
SAS-3	12 Gbit/s	1.2 GB/s	10 m
Thunderbolt 1	10 Gbit/s	1.22 GB/s	3 m (měď), 60 m (optika)
USB 3.1	10 Gbit/s	1.21 GB/s	1 m
SAS-2	6 Gbit/s	600 MB/s	10 m
SATA 3.0	6 Gbit/s	600 MB/s	1 m
USB 3.0	5 Gbit/s	400 MB/s	3 m
FireWire 3200	3.144 Gbit/s	393 MB/s	100 m+
SAS 1.0	3 Gbit/s	300 MB/s	10 m
SATA 2.0	3 Gbit/s	300 MB/s	1 m
USB 2.0	480 Mbit/s	35 MB/s	5 m

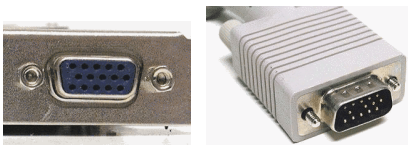
Tabulka 3.6: Srovnání rozhraní pro mass storage

3.6 Konektory na grafických kartách

Na grafických kartách bývají především rozhraní VGA (také D-SUB) a DVI. Dále se setkáme s čistě digitálními rozhraními HDMI a DisplayPort, výjimečně ještě s dalšími.

3.6.1 VGA

VGA (resp. D-SUB) je plně *analogový*. Písmeno „D“ v názvu pochází od typického „děčkového“ tvaru konektoru.



Obrázek 3.15: Rozhraní VGA

Rozhraní VGA vidíme na obrázku 3.15. Svými rozměry a uspořádáním se může zdát zaměnitelné se sériovým rozhraním RS-232, ale jak vidíme, VGA na straně počítače je opatřeno „dírkami“, kdežto RS-232 má „kolíky“ – viz obrázek 3.5 na straně 38 (na straně kabelu je to samozřejmě naopak, aby bylo možné kabel připojit). Další rozdíl je v uspořádání kolíků/dírek – zatímco RS-232 má dvě řady, VGA tři.

Poznámka

Konektor či jack (zásuvka) může mít „dírky“ nebo „kolíky“. První možnost se běžně označuje *female* (samice, čteme [fi:meil]), druhá možnost *male* (samec, čteme [meil]). O konektoru VGA tedy mluvíme jako o male (samec) konektoru, VGA jack je female (samice). U RS-232 je to naopak – konektor je female a jack male.

VGA je určen především pro projektory CRT monitory (ty pracují na analogovém principu), ale toto rozhraní někdy najdeme i na (digitálních) LCD monitorech. U zobrazovacích zařízení (například monitorů) zpracovávajících digitální obraz, typicky LCD obrazovek, je obvykle možné použít i analogový VGA, ale pak dochází ke zbytečné konverzi

digitální (grafická karta) → analogový (VGA) → digitální (LCD)

Důsledkem bývá trochu horší obraz, než jakého lze dosáhnout s digitálním rozhraním (konverzí se ztrácí část informace), navíc může dojít ke zhoršení odezvy. U kancelářských aplikací je to celkem jedno, důsledky se projevovují například při hraní rychlejších her.

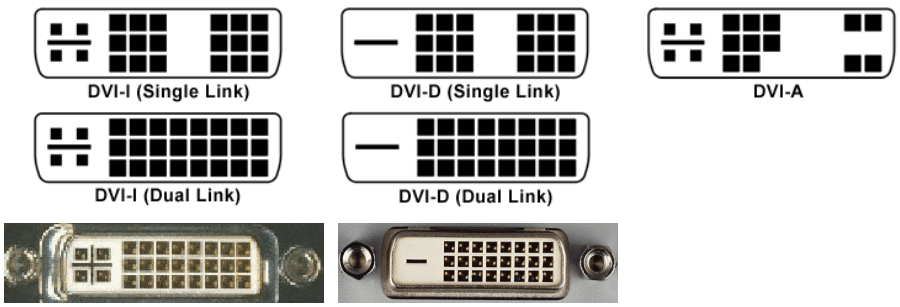
3.6.2 DVI

DVI (Digital Visual Interface) je rozhraní chápané spíše jako digitální, ale ve skutečnosti některé jeho typy jsou kombinované (dokážou přenášet analogový signál bez konverze, tj. kombinace digitálního a analogového rozhraní) a dokonce byla navržena i plně analogová varianta.

Existuje více typů DVI konektorů, jak vidíme také na obrázku 3.16:

- DVI-A – jen analogový, pro CRT monitory a některé TV karty, konverze D/A na analogový výstup, v praxi se s ním téměř nesetkáme,
- DVI-D – používá pouze digitální piny, je nejběžnější, existuje ve dvou verzích:
  1. dual (používá všechny digitální piny),
  2. single (některé piny nejsou použity),
- DVI-I (Integrated D/A) – digitální i analogové piny jsou přítomny a funkční, také existují varianty single a dual.

Dual DVI se také označuje jako DVI-DL. Jeho propustnost je vyšší (Single má 4,9 Gb/s, Dual 9,9 Gb/s) a je určeno pro monitory s vysokým rozlišením a vysokou obnovovací frekvencí (pro ně je nutný, včetně kabelu), pro kancelářské použití nepřináší žádné výhody.



Obrázek 3.16: Rozhraní DVI[9]

[9] Zdroj: http://www.deltapage.com/

Poznámka

Z toho vyplývá, že pokud chcete připojit monitor s vysokým rozlišením a vyšší obnovovací frekvencí (což dražší kvalitní monitory splňují), třeba volit variantu kabelu (vč. konektorů) dual, a takový jack by měl být i na počítači, ke kterému bude monitor tímto kabelem připojen. Pokud se jedná dokonce o monitor s rozlišením 4k, nelze rozhraní DVI použít vůbec (nebo ano, ale s nižším rozlišením).

Rozhraní DVI přenáší jen obraz, tedy pokud máme na monitoru reproduktory, musíme ho propojit také se zvukovou kartou.

Další informace

- http://www.hardwarebook.info/Digital_Visual_Interface_(DVI)
- http://www.datapro.net/techinfo/dvi_info.html

3.6.3 HDMI, DisplayPort

HDMI a DisplayPort jsou na rozdíl od předchozích plně digitální rozhraní.

HDMI (High-Definition Multimedia Interface) představuje rozhraní pro přenos digitálního nekomprimovaného obrazu a zvuku (na rozdíl od DVI, které přenáší jen obraz). Výhodou je, že HDMI používá stejnou specifikaci pro signál s obrazem jako DVI-D (jen k tomu přidává zvuk), tedy redukce mezi těmito dvěma rozhraními jsou velmi jednoduché, pasivní.

Poznámka

Proč je důležité, že HDMI (a DisplayPort) přednáší *nekomprimovaný* multimediální signál? Protože komprimace sice na jednu stranu snižuje množství dat, které je nutno přes rozhraní přenést, ale na druhou stranu je zbytečně časově náročná. Tato rozhraní tedy poskytují vysokou propustnost, při které není komprimace nutná, důsledkem jsou dostačující rychlosti (přenášený obraz a zvuk lze v reálném čase přehrávat).

Také u rozhraní HDMI existují různé *verze*. Dnes je nejběžnější verze 1.4 z roku 2009, která s přehledem postačuje pro rozlišení Full HD s frekvencí až 120 Hz: teoretická propustnost je 10,2 Gbit/s (reálná asi o pětinu nižší). Ale pokud bychom chtěli přenášet video v rozlišení 4K, půjde to jen s nižší frekvencí (30 Hz) – každý rychlejší pohyb na videu nebo ve hře by byl trhaný.

Rozlišení 4K s dostačující frekvencí (60 Hz) zvládá až verze HDMI 2.0 z roku 2013, které oproti verzi 1.4 má zhruba dvojnásobný datový tok. Verze 2.1 z roku 2017 by měla zvládnout rozlišení až 8K při frekvenci 120 Hz. Takže když kupujeme HDMI kabel nebo cokoliv s HDMI konektorem či jackem, měli bychom vědět, pro jaký účel to bude využíváno, jaké budou nároky.

Poznámka

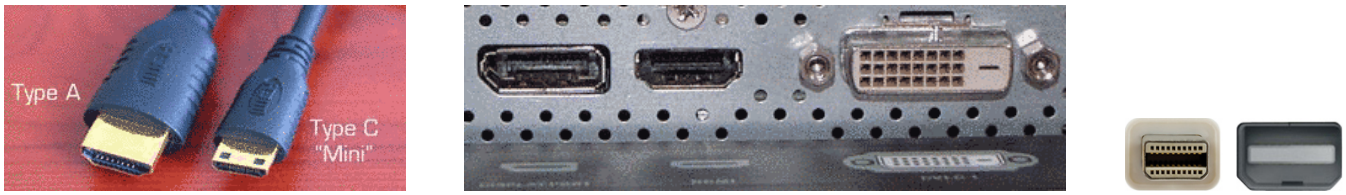
Nemá moc smysl srovnávat propustnost DVI a HDMI, protože DVI přenáší jen obraz, kdežto HDMI také zvuk.

Existuje několik typů HDMI konektorů – většinou se setkáváme s A (klasickým) a C (Mini), oba (vidíme na obrázku 3.17), existuje i varianta Micro. Konektor a celé rozhraní je menší než DVI. Doporučuje se

používat maximálně 5metrový kabel, ale za určitých okolností je použitelný i delší (kvalitní silné vysílače signálu v rozhraních zařízení). U kabelu nad 10 metrů jsou v koncovkách aktivní zesilovače. Existují také optické kabely HDMI, které mohou být mnohem delší.

HDMI je určeno především pro televize (včetně přenosu HD) a některou další spotřební elektroniku, a dále pro LCD monitory a další zařízení. Mini variantu najdeme například na videokamerách, Micro varianta je pro malá mobilní zařízení.

DisplayPort je digitální rozhraní určené pro LCD monitory a jiné zobrazovací periferie. Přenáší nekomprimovaný digitální obsah s podporou ochrany se 128bitovým šifrováním AES, a 8kanálového zvuku. Od HDMI se mimo jiné odlišuje volnější licencí (a samozřejmě také tvarem portu – má jeden roh „seříznutý“ a trochu jiné rozměry i vnitřní uspořádání).



Obrázek 3.17: HDMI konektory A a C, dále DisplayPort a jeho porovnání s HDMI a DVI, vpravo Mini DisplayPort[10]

Také DisplayPort existuje v různých *verzích*. Verze 1.0 (rok 2006) má teoretickou propustnost 10,8 Gb/s, což je o něco více než dual-link DVI a mírně více než HDMI verze 1.4. DisplayPort verze 1.2 (rok 2009, zatím nejběžnější) má dvojnásobnou propustnost (konkuruje novějšímu HDMI 2.0, rozlišení 4K přenese s frekvencí 60 Hz) a verze 1.3 a 1.4 dokonce trojnásobnou (32,4 Gb/s). Mezi verzemi 1.3 a 1.4 není přímo rozdíl v rychlosti, vylepšilo se však například vzorkování přenášeného zvuku, korekce chyb, atd.

Kabel může být dlouhý až 15 metrů s tím, že kratší délka kabelu znamená možnost přenášet vyšší rozlišení obrazu (tj. u verze 1.0 do 3 metrů rozlišení 2560×1600, u delšího kabelu o něco méně).

DisplayPort má standardní konektor přibližně stejně velký jako HDMI, ale působí poněkud pevnějším a odolnějším dojmem. Na obrázku 3.17 uprostřed (strana 47) vidíme zásuvky tří rozhraní – postupně DisplayPort, HDMI a DVI-D. Často narazíme na zmenšenou variantu – Mini DisplayPort; kromě jiného proto, že se používá jako fyzické rozhraní pro Thunderbolt verze 2.0 (informace o Thunderboltu je na straně 35). Na obrázku 3.17 vpravo je konektor a zásuvka pro Mini DisplayPort.

Poznámka

DisplayPort zachází se signálem jinak než DVI a HDMI (VGA, DVI a HDMI přenášejí zvlášť v různých drátech jednotlivé základní barvy, v dalších drátech synchronizační informaci apod.). Přenos je podobnější tomu, co funguje v počítačových sítích, nazývá se *mikro-paketový přenos*. Znamená to, že vše je přenášeno v malých datových jednotkách (barva se nedělí do drátů podle základních barev) bez nutnosti použití zvláštního vodiče pro hodinový signál kvůli synchronizaci (časový údaj je součástí paketu). Ovšem DisplayPort umí emitovat DVI a HDMI signál, viz dále.

[10] Zdroj: http://pcworld.cz/hardware/displayport-zabijak-dvi-3290

Existují redukce DisplayPortu na DVI a HDMI (ale opačný směr je značně problematický), i přes rozdílný způsob přenosu signálu zpětná kompatibilita jedním směrem existuje: některá zařízení mohou být označena jako *Dual-mode* (příslušné logo vidíme vpravo). Jedná se o zařízení schopná komunikovat přes DisplayPort signálem jak pro DisplayPort, tak i HDMI, případně DVI (tj. přes DisplayPort dokážou posílat DVI nebo HDMI signál). Při zapojení takové zařízení (například grafická karta) detekuje rozhraní použité na druhé straně (přes redukci DisplayPort/HDMI nebo DisplayPort/DVI) a podle tohoto přizpůsobí vysílaný signál.

Jednoduché pasivní redukce mezi VGA a DisplayPort neexistují, je potřeba využít elektricky napájený konverter. Podobně u konverzí s DVI a HDMI signálem, pokud naše zařízení nespadají do kategorie popsané v předchozím odstavci.

Další informace

- http://www.datapro.net/techinfo/hdmi_info.html
- http://www.displayport.org/
- http://notebook.cz/clanky/technologie/2008/0518-technologie-displayport
- http://www.datapro.net/techinfo/displayport_info.html

3.6.4 MHL

MHL (Mobile High-Definition Link) je signální rozhraní využívající na nižší úrovni rozhraní HDMI. Používá se zejména pro propojení malých mobilních zařízení s televizory či projektory. Existuje několik verzí, z nichž MHL 3.0 umožňuje přenášet i 4K video (ale pouze na nižší frekvenci – 30 Hz), superMHL přenáší až 8K video při frekvenci 120 Hz (záleží i na typu kabelu).

Používá se kabel, který má na jedné straně HDMI konektor (strana pro televizor/projektor) a na druhé mikro USB (smartphone apod.). Kabel může být buď pasivní (pouze tehdy, když mobilní zařízení „umí“ MHL) nebo aktivní s MHL adaptérem (použijeme, pokud mobilní zařízení neobsahuje funkcionalitu MHL).

Další možnost je využití USB-C s Alternate mode MHL pro USB 3.1 a vyšší, takový kabel má na jednom konci USB-C (mobilní zařízení) a na druhém HDMI (televizor).

3.6.5 Další rozhraní pro přenos obrazu

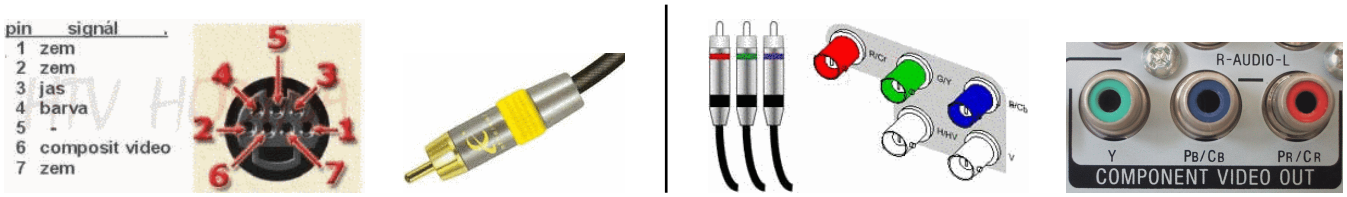
Na grafických kartách (případně na back panelu, pokud je grafika integrovaná) můžeme najít i další (analogová) rozhraní. Liší se samozřejmě způsobem přenosu informace, ale také kvalitou přenášených dat, čímž je dáno typické využití.

S-Video (Separate Video) přenáší odděleně jas a barvu. Má několik variant:

- MiniDIN-4 – 4pinový, pro spotřební elektroniku (DVD, VCR, TV),
- MiniDIN-7 – 7pinový, pro počítače (často se takto připojují projektory).
- další – 3kolíkový, 9kolíkový atd.

Signál není nijak zvlášť kvalitní (žádné HD).

Composite Video je určeno pro TV, bezpečnostní kamery, některé monitory. Kvalita signálu je hodně špatná. Používá se obvykle buď konektor typu jack (přesněji RCA jack), případně u televizí to může být SCART.



Obrázek 3.18: 7pinový S-Video a Composite Video jack[11]

Component Video přenáší zvlášť signály pro základní barvy R, G, B (zvlášť tři kabely). Na rozdíl od předchozího je signál poměrně kvalitní, dodnes se používá například ve spotřební elektronice.

Z těchto analogových rozhraní se asi nejčastěji setkáme se 7pinovým S-Video, kterým můžeme připojit počítač k projektoru (projektory také lze připojit přes D-SUB, který je na nich také označován jako VGA – záleží, jaký máme kabel a jaké rozhraní je k dispozici na obou přístrojích).

Na některých grafických kartách se také setkáme s rozhraními *TV OUT* a někdy i *TV IN*. Přes TV OUT jde výstup televizního signálu do televize, videa nebo projektoru, přes TV IN vedeme do počítače vstup z videa, videokamery nebo digitálního fotoaparátu (i když u toho fotoaparátu bývá častější USB).

3.7 Síť a možnosti bezdrátového propojení

Aby zařízení mohlo komunikovat v počítačové síti, musí mít komponentu, která mu to umožní – síťovou kartu nebo její integrovanou variantu. Tato komponenta má rozhraní, přes které se k síti připojuje.

Ethernet a Wi-fi: Jestliže se k síti připojujeme přes kabel, jde většinou o síť Ethernet s konektorem/zásuvkou RJ-45 (správnější označení je 8p8c). Pokud se jedná o bezdrátovou síť (Wi-fi, mobilní sítě), pak je fyzickým rozhraním anténa, která často (hlavně u mobilních zařízení) bývá integrovaná. Dále pokud připojujeme ADSL/VDSL modem, potřebujeme kabel s konektory typu RJ-11 (telefonní). Signální rozhraní ve všech případech zajišťují *síťové protokoly*.

Se síťovými rozhraními se seznamujeme především v předmětu *Počítačová síť a internet*.

Poznámka

U předchozích probíraných rozhraní jsme se někdy zmínili o redukci (možnosti kombinovat různá rozhraní). Se síťovými rozhraními to je podobné. Pokud chceme připojit k síti takové zařízení, které není opatřeno příslušným síťovým rozhraním, pak buď pořídíme interní síťovou kartu (pokud je to technicky možné), nebo se napojíme na některé univerzální rozhraní, typicky USB. Existují externí síťové karty jak pro Ethernet (kabel), tak i bezdrátové, které mají rozhraní USB. Funguje to, jen může být problém s tím, že těmito datovými přenosy je zatěžován procesor (což je typická vlastnost USB).

Bluetooth: Je to chronicky známé rozhraní pro bezdrátové propojení zařízení na malou vzdálenost. Existuje několik výkonových tříd – čím vyšší výkon, tím větší dosah (třída 3 znamená dosah do 1 m, třída 1 až 100 m). Propustnost není nijak závratná, typicky jde o jednotky Mbit/s, záleží na vzdálenosti a výkonu (třídě), a samozřejmě na verzi (verze 4.0 až 24 Mbit/s). V každém případě Bluetooth není určeno k přenovu velkých kvant dat. S tímto rozhraním se setkáváme hlavně u malých mobilních zařízení a některých ovladačů.

Ve srovnání s IrDA (přenos po infračerveném záření) je výhodou Bluetooth všesměrovost (nemusíme přímo mířit na cíl) a taky rozšířenost.

[11] Zdroje: http://www.weethet.nl/english/video_connect_pc2tv.php, http://en.wikipedia.org/wiki/Component_video

UPnP (Universal Plug-and-Play) je technologie a sada protokolů, jejímž účelem je maximálně zjednodušit existenci zařízení v síti. Jde o to, aby zařízení bylo schopno po připojení co nejdřív komunikovat: dostat adresu, zorientovat se v síti (zjistit, jaká jiná zařízení v síti jsou a jak je lze využívat), naopak prezentovat v síti sebe, zjišťovat události v síti a reagovat na ně. To vše bez ohledu na to, o jakou síť jde (Ethernet, Wi-fi, atd.). UPnP je základem pro fungování dalších služeb, především v souvislosti s multimédii, audio-video službami.

UPnP se používá hlavně v domácnostech, ovšem i tam je trochu problém s tím, že se zařízení prakticky neautentizují a služba je poměrně snadno zneužitelná. Podporu UPnP na routeru se z bezpečnostních důvodů doporučuje vypnout, protože se dá zneužít k neautorizovanému přístupu do sítě zvenčí (využitím protokolu UPnP IGD).

Další informace

Další informace například na
https://cdr.cz/clanek/upnp-otevira-utocnikum-cestu-k-milionum-routeru-stenata-hledaji-xss.

DLNA (Digital Living Network Alliance) je technologie umožňující sdílení dat z multimediálních zařízení (založená na UPnP), stojí za ní především společnost Sony. Setkáváme se s ní u notebooků, malých mobilních zařízení, televizorů, multimediálních přehrávačů, herních konzolí, tiskáren a dalších zařízení, takové zařízení by mělo být označeno jako „DLNA Certified“.

Zdroj multimediálních dat označujeme DLNA server (například NAS server, mobil, herní konzola), cíl je DLNA klient (například televize přehrávající video z mobilu). Na DLNA serverech a klientech musí běžet UPnP, naopak vůbec nemusí běžet na routeru. Důležitým předpokladem je, že multimediální obsah musí být „srozumitelný“ DLNA klientovi, klient si musí umět poradit s formátem, kodeky apod., DLNA server pouze data posílá. Takže například pokud si kupujeme televizi s podporou DLNA, měli bychom se taky zajímat o to, co reálně dokáže přehrát. DLNA server sice může být konfigurován tak, aby před odesíláním data konvertoval do jiného formátu, ale nemusí (a ani to nemusí jít), DLNA specifikace to nevyžaduje.

Další informace

https://pctuning.tyden.cz/index.php?option=com_content&view=article&id=51595&catid=32&Itemid=72

Miracast, Chromecast a AirPlay jsou konkurenti DLNA. Miracast je určen ke zrcadlení obsahu obrazovky vysílajícího zařízení (tj. výpočetní zátěž je spíše na vysílači) a funguje pouze přes bezdrátovou síť, je třeba počítat s drobným zpožděním obrazu (ale na druhou stranu nebývá problém se zobrazováním titulků u videa). Chromecast od Googlu umí jak zrcadlit obsah obrazovky, tak i streamovat multimediální obsah (video, hry, podle podpory v aplikacích). AirPlay pochází od společnosti Apple a slouží k podobným účelům jako Chromecast.

Další informace

- https://www.samuraj-cz.com/clanek/bezdratove-promitani-pomoci-technologie-miracast/
- https://avmania.zive.cz/co-umi-miracast—bezdratovy-prenos-obrazu
- https://www.alza.cz/vse-o-google-chromecast-art16497.htm

Další bezdrátové náhrady HDMI: V posledních letech se objevují speciální bezdrátová rozhraní pro vedení signálu s daty, vpodstatě bezdrátové náhrady rozhraní HDMI. Mohou například sloužit k pro-

pojení televizoru s počítačem nebo malým mobilním zařízením, pokud ovšem zajistíme podporu rozhraní na obou stranách. Můžeme se setkat například s těmito řešeními:

- *WirelessHD* – za tímto standardem stojí víc než 40 firem převážně z oblasti spotřební elektroniky, jedná se o bezztrátový přenos dat (u méně energeticky náročných zařízení také energie) na frekvenci kolem 60 GHz, je použitelný i pro přenos HD videa do vzdálenosti až 10 m (nízký dosah je zaviněn vysokou frekvencí). V současné době se s ním moc nesetkáme, bývá to jedna z možností propojení u některých projektorů Epson.
- *WHDI* je (co se týče kvality signálu) řešení srovnatelné či mírně lepší než WirelessHD, taky má podobné využití. Nicméně pracuje na nižší frekvenci (kolem 5 GHz), tedy má vyšší dosah (cca 30 m) a má menší problém s pronikáním za zeď. S rozšířeností je to podobné jeko u WirelessHD, ovšem spíše u produktů BenQ.
- *Intel WiDi* (Wireless Display) je technologie společnosti Intel dostupná na některých mobilních procesorech této firmy, ovšem Intel už od této technologie ustupuje. Jde o obdobu WirelessHD.
- *UWB* (Ultra-Wide Band) je specifikace fyz. vrstvy pro rádiový přenos na frekvenci 3.1–10.6 GHz, dosah je vyšší než WirelessHD (až 50 m), ale přenosové rychlosti nižší, tuto technologii využívají například *IEEE 802.15* (Wireless PAN – Personal Area Network) nebo WirelessUSB.

Poznámka

Pokud jde o propojení dvou či více zařízení s využitím výše zmíněných bezdrátových technologií a o přenos HD videa, obvykle jde o to mít v zařízení vestavěnou podporu dané technologie nebo mít speciální vysílač/přijímač připojený k zařízení obvykle přes HDMI. Například pokud chceme doma využít WiDi (předpokládejme, že na notebooku máme procesor podporující WiDi), ale naše televize WiDi nepodporuje, je možné pořídit WiDi konverter, který k televizi připojíme kabelem HDMI – tím televizi dodáme podporu tohoto rozhraní. Podobně se dají koupit USB dongly pro Chromecast, Miracast, AirPlay, DLNA (typicky několik z toho najednou).

Další informace

- http://www.svethardware.cz/art_doc-6833BBB427ADEB69C12575C500323EF3.html
- http://hw.cz/teorie-praxe/art2004-wireless-usb-bezdratove-usb.html
- http://www.cnet.com/news/wireless-hd-video-is-here-so-why-do-we-still-use-hdmi-cables/

Další informace

Existuje mnoho stránek s přehledy různých rozhraní, například:

- http://www.tomshardware.com/reviews/pc-interfaces-101,1177-2.html
- http://www.hd.cz/rady/prehled-pocitacovych-konektoru-zadni-panel-usb.php
- http://media.techtarget.com/whatismedia/images/hardware-chart-large.png

Teoretickou propustnost různých rozhraní můžeme porovnat na
https://en.wikipedia.org/wiki/List_of_device_bit_rates

Kapitola 4

Case a základní deska

Rychlý náhled: Celá kapitola je věnována nejdůležitější součásti počítače, základní desce (motherboard, mainboard). Probereme fyzickou strukturu základní desky, nejdůležitější součásti, rozhraní, integrovaná zařízení, chlazení a problémy, které souvisejí s rozvržením komponent na základní desce. Na začátku kapitoly se podíváme na to, v čem vlastně základní deska a většina komponent bývá uložena.

Klíčová slova: Skříň počítače, case, chassi, tower, rack, node, blade server, mainframe, superpočítač, HTPC, PCB, Form Factor, ATX, ITX, STX, front panel, back panel, sběrnice, PCI, PCI Express (PCIe), AGP, chipset (čipset), North-South Bridge Design, One-Chip Design, FSB, DMI, HyperTransport (HT), QuickPath Interconnect (QPI), SoC, propojka (jumper).

Cíle studia: Po prostudování této kapitoly budete vědět, k čemu slouží základní deska, co vše na ní najdeme, co jsou a jak fungují sběrnice na základní desce, jak vypadá a pracuje čipset, jaký vliv má čipset na vlastnosti základní desky a celého počítače.

4.1 Skříň počítače

Skříň počítače (case, chassi; množné číslo je chassis) není jen estetickou záležitostí. Na materiálu, ze kterého je vyrobena, velikosti, tvaru a vnitřním uspořádání závisí funkčnost počítače (schopnost uchlazení – cirkulace vzduchu, možnosti osazení komponentami včetně výběru základní desky, atd.). Pokud jsou stěny skříně příliš tenké nebo z nevhodného materiálu, mohou přenášet vibrace (z větráčků, z roztočeného pevného disku či optické mechaniky) a tím se podílet na zesilování zvuku vydávaného počítačem.

Rozlišujeme různé typy skříní:

- desktop, tower, minitower, big tower, small form factor – různě velké pro desktopové počítače,
- notebookové chassi – „skříň“ pro notebook,
- skříně pro servery:
  - tower – obvykle pro menší firmy, nenáročné použití,
  - server v racku[1] (serverové skříni, „rack-mount“) – do racku se umísťuje obvykle více serverových jednotek s unifikovanou velikostí, viz dále,
  - blade server,

[1] Rack – čti [ræk], s otevřeným „e“. V angličtině to znamená stojan, regál či polici.

  - mainframe (sálový) pro řízení složitých provozů, hodně se prolíná s následující kategorií,
  - superpočítač – velmi výkonný počítač ve formě clusteru (tj. mnoho výkonných počítačů propojených vysokorychlostní sítí, obvykle v jedné hale nebo budově, hardwarově a softwarově propojených tak, že virtuálně fungují jako jeden stroj),
- skříně pro HTPC (Home Theatre PC, malé domácí servery sloužící obvykle jako sklad filmů, fotek a jiných rodinných souborů) – malý rozměr, předpokládá se tichý chod, tedy důraz na cirkulaci vzduchu i v malém prostoru,
- obaly pro embedded (zařízení, která sama o sobě nejsou počítačem, ale mají v sobě počítač vestavěný, třeba DVD přehrávače nebo síťová zařízení), menší formáty (mobilní telefony, tablety apod.).

Rack (serverová skříň, rozvaděč) obsahuje tzv. *nodes* („uzly“ – servery, switche, routery, hardwarové firewally, apod.) s výškou v násobcích 1,75 in = 4,45 cm (označuje se 1U, 2U, ..., podle násobku uvedené výšky), maximální výška celého racku je 42U (ale obvykle se setkáme se spíše menšími skříněmi).

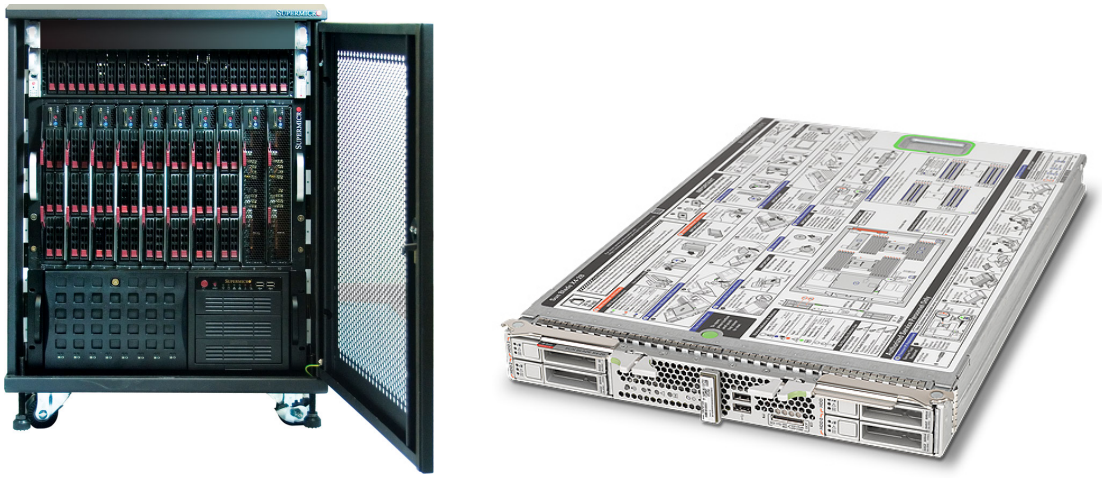
Každý node obsahuje základní desku, procesory, paměti, atd., jedná se o samostatný systém. Rack všechny nody navzájem propojuje, zajišťuje správu napájení, chlazení a komunikačních rozhraní. To znamená, že když instalujeme node do racku, připojujeme jej k napájecímu a chladicímu systému racku a samozřejmě také k síťovým rozhraním. Aby bylo především chlazení dostatečně účinné, měli bychom jednotlivé nodes rozmístit tak, aby ve skříni mohl vzduch dobře proudit, tj. „nekumulovat“ je jen v jedné oblasti skříně.

Některé towery se také dají zapojit do racku.

Obrázek 4.1: Rack (serverová skříň)[2]

Systém Blade představuje další stupeň integrace. Celý systém je integrován na jedné (základní) desce uzavřené v pouzdře, uvnitř nejsou žádné kabely, více takových desek bývá umístěno v jednom *blade chassi*, což je často node do racku, anebo tower.

Typickou vlastností bladů je nekompatibilita: pokud máme blade chassi od konkrétního výrobce, musíme nakupovat blady od téhož výrobce. Naopak výhodou je určitá kompaktnost a také obvykle jednodušší zprovoznění a správa, nižší spotřeba energie.



Obrázek 4.2: Blade server – blade v racku (vlevo) a deska se systémem (vpravo)[3]

[2] Zdroj: http://www.shaktigroupindia.net/etsi-telecom-server-racks.html

[3] Zdroj: http://www.supermicro.com/servers/blade/, http://www.oracle.com/us/products/servers-storage/servers/blades/overview/index.html

4.2 Co je to základní deska

Základní deska je základní komponenta počítače, notebooku a vlastně jakékoliv elektroniky. Poskytuje univerzální datová a napájecí rozhraní a fyzické uložení pro jednotlivé komponenty. Fyzicky je realizovaná pomocí několikavrstvé desky tištěných spojů osazené mnoha elektronickými obvody a konektory.

Nejznámější výrobci základních desek jsou Gigabyte, Intel, MSI, Asus, ASRock, ECS.

4.2.1 PCB

PCB (Printed Circuit Board, deska tištěných spojů) je základní součástí každé základní desky. Na PCB se umísťují další elektronické součástky a konektory. Podobnou strukturu mají také rozšiřující desky (adaptéry), například grafická karta. PCB má tyto funkce:

1. mechanická – udržuje všechny komponenty fyzicky pohromadě,
2. elektronická – poskytuje komunikační spojení mezi jednotlivými komponentami na nejnižší fyzické úrovni,
3. energetická – pomáhá napájet připojené komponenty,
4. tepelná – slouží k základnímu chlazení komponent.

Obvyklá struktura PCB je nastíněna na obrázku 4.3. PCB tedy členíme do vrstev, z nichž některé se ve struktuře objevují vícekrát. Na množství vrstev, jejichž pořadí je patrné z obrázku, závisí tloušťka desky. Význam jednotlivých vrstev je následující:

- *Povrchová ochranná vrstva* – chrání desku před vnějšími vlivy, ochranný film, nástřik,
- *Signálová vrstva* – zajišťuje logickou komunikaci, tvořena tenkou vrstvou vodivého materiálu, většinou mědi,
- *Izolační vrstva* – odděluje signálovou a zemnící vrstvu, většinou z Preprogu (sklo-uhlíková výztuž prosycená polymerní matricí),
- *Napájecí-zemnící vrstva* – zajišťuje zemnění, napájení a odvod tepla, převážně z mědi,
- *Nosné jádro* – zajišťuje mechanické vlastnosti desky.

Povrchová ochranná vrstva
Signálová vrstva 1
Izolační vrstva
Napájecí-zemnící vrstva 1
Nosné jádro
Napájecí-zemnící vrstva 2
Izolační vrstva
Signálová vrstva 2
Povrchová ochranná vrstva

Obrázek 4.3: Struktura PCB základní desky

4.2.2 Form factor

Používá se mnoho různých typů základních desek, odlišnosti jsou zejména v těchto vlastnostech:

- rozměry desky,
- rozměry, způsob uchycení a typ napájecích konektorů,

- výrobce,
- vybavenost rozhraními, paticemi apod., tj. také jaké procesory podporuje, jakou má čipovou sadu, kolik karet lze připojit, atd.,
- možnosti přetaktování (souvisí také s vybavením BIOSu základní desky) a způsob chlazení komponent.

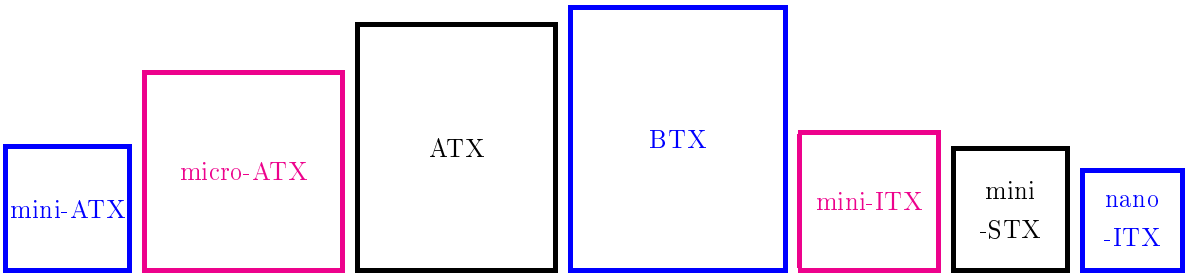
Nejobvyklejší typy základních desek (*Form Factor*) najdeme v tabulce 4.1. Liší se především rozměry, způsobem chlazení (směrem proudění vzduchu), a dále uspořádáním, rozměry a způsobem uchycení napájecích konektorů, patic, čipsetů, apod. Výrobci menších zařízení (ale taky notebooků) si tvar základní desky často „personalizují“ co se týče rozměrů, tvaru a rozložení součástí – někdy taková deska ani vzdáleně nepřipomíná obdélník.

Nejpoužívanější formáty jsou následující:

- *ATX* (Intel, 1995, od 1999 standard) – populární u desktopů, 7 slotů s odstupy 20 mm, variantou je *micro-ATX* – zmenšená asi o 25 %, má méně slotů pro rozšiřující karty,
- *BTX* (Intel, 2004) – jinak chlazené komponenty (změna způsobu cirkulace vzduchu), neujal se,
- *ITX* (VIA, 2001) – menší, komponenty na výšku, přímo s tímto formátem se dnes nesetkáme, variantou je *mini-ITX* často používaný v netboocích a malých desktopech, procesor bývá (ne vždy) přiletován na základní desce,
- například základní deska pro Raspberry Pi má rozměry něco mezi pico-ITX a mobile-ITX,
- nejnovější základní formát je *STX*, používá se *mini-STX* zabírající přibližně 3/4 rozměru mini-ITX; na těchto základních deskách sice najdeme běžné procesorové patice, ale například paměťové moduly jsou kratší (notebookové).

Standard	**Rozměr (mm)**	**Standard**	**Rozměr (mm)**
ATX (Intel)	305×244	mini-ITX (VIA)	170×170
micro-ATX	244×244	nano-ITX	120×120
flex-ATX	229×191	pico-ITX	100×72
mini-ATX	150×150	mobile-ITX	75×45
BTX (Intel)	325×266	DTX (AMD)	244×203
mini-STX	140×147	mini-DTX	203×170

Tabulka 4.1: Nejpoužívanější formáty základních desek (Form Factor)



Obrázek 4.4: Srovnání některých běžných form factorů

Pro *embedded zařízení* (set-top-boxy, různá spotřební elektronika, routery a další síťová zařízení, bankomaty, počítače v automobilech, apod.) a pro malá mobilní zařízení se používají buď některé z výše

zmíněných desek (například odvozené od ITX – spíše pro větší embedded), nebo form factory speciálně navržené pro tyto účely – ETX (95×114), COM Express (několik variant, varianta Compact má rozměr 95×95), PCI/104 (opět několik variant) a další.

U *serverů* se můžeme setkat například s formátem EATX (Extended ATX – 305×330) pro servery do racku nebo WTX (355.6×425.4) určeným pro tower servery. Obsahují více patic pro procesory, SATA portů a slotů pro paměti, naopak často se nesetkáme se sběrnicemi, které se obvykle používají pro grafické karty.

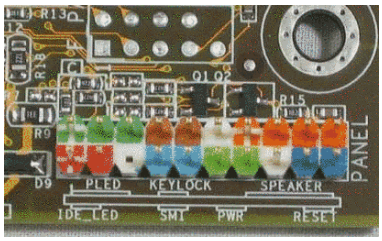
Poznámka

Pojem form factor se ve skutečnosti používá i v jiných významech. Například u pevných disků se takto rozlišují různé rozměry – pevný disk s Form Factorem 3.5" apod.

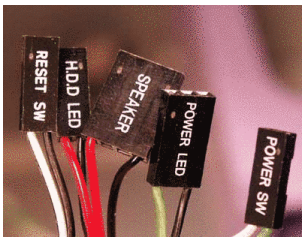
4.2.3 Interfaces

Rozhraní (interfaces) mají rozšiřovat funkce a/nebo výkon základních desek o další možnosti. Účelem je jednoznačně definovat způsob přenosu dat a způsob napájení mezi základní deskou a rozšiřující kartou. Jako součást interfaces můžeme také brát I/O back panel či front panel.

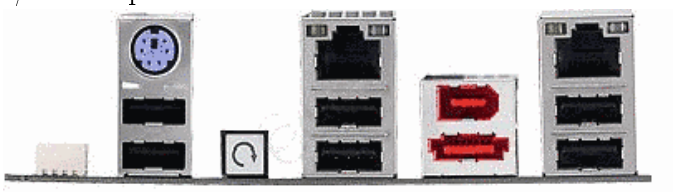
Front panel



Konektory k front panelu



I/O back panel



Obrázek 4.5: Front panel a I/O back panel[4]

Front panel (čelní panel) je součást základní desky, která obvykle přiléhá zevnitř k čelní straně počítače (nebo je v některém předním rohu) a jejím účelem je propojit základní desku se vším, co má být vyvedeno ven ze skříně počítače na čelní straně. Na obrázku 4.5 ho vidíme nahoře vlevo. Vedle je sada konektorů, které se sem zapojují (k různým indikačním diodám – třeba pro indikaci činnosti pevného disku, ale také třeba k tlačítku napájení nebo resetu).

Back panel (taky I/O panel, na obrázku 4.5 dole) se směruje k zadní straně počítače a jsou na něm vyvedeny ven zásuvky od integrovaných komponent – USB řadičů (tj. USB jacky), zásuvky grafických rozhraní VGA, DVI, HDMI (pokud máme integrovanou grafiku), zvuku, sítě apod.

Back panel má pro nás ještě jeden důležitý význam – podle něj spolehlivě poznáme, jak má být základní deska nasměrovaná, když ji vkládáme do skříně – vždy bude vzadu.

[4] Zdroj: http://infoarchena.blogspot.com/2011/04/front-panel-ii.html

Poznámka

Když kupujeme základní desku, obvykle k ní dostaneme i kus plechu, do kterého jsou vyraženy díry v místech, kde na back panelu desky vyčnívají různá rozhraní.

Back panel totiž může být různě poskládán podle toho, co vše obsahuje, a výrobce skříně počítače nemůže předem vědět, jak bude vypadat back panel uživatelem vybrané základní desky – takže zadní strana skříně bývá prostě „bez stěny“, tuto stěnu bude tvořit kombinace plechu a back panelu. Pokud nemáme po ruce ten správný kus plechu, můžeme si kleštěmi na plech upravit nějaký jiný, který téměř pasuje.

Další informace

- http://www.formfactors.org/formfactor.asp
- http://en.wikipedia.org/wiki/Computer_form_factor (zajímavé srovnání rozměrů)
- http://www.tomshardware.com/reviews/best-motherboard-guide,2546-4.html
- http://www.anandtech.com/show/1157/2 (neberte moc vážně, BTX se neujala)
- http://www.tomshardware.com/reviews/atx-btx-cases,1187-19.html
- https://smallformfactor.net/articles/editorial/stx-changed-view-sff/

4.3 Sběrnice

Důležitou součástí základní desky jsou sběrnice. Jde o *komunikační linky*, fyzicky realizované tištěnými spoji v případě interních sběrnic, kterým se zde budeme věnovat (ve skutečnosti můžeme hovořit např. o USB sběrnici, to je taky komunikační linka, i když externí). Na sběrnice se napojují rozšiřující karty nebo externí zařízení, další sběrnice slouží ke komunikaci mezi procesorem a čipsetem, pamětí (paměťová sběrnice) apod., sběrnice jsou také uvnitř procesoru nebo naopak jsou vyvedeny ven z počítače.

Sběrnice je komunikační cesta, na kterou je třeba se „nějak“ dostat. V předchozí kapitole jsme se seznámili s rozhraními USB, SATA a dalšími, pro která byly definovány konektory (plug) a zásuvky (jack). Pojem sběrnice zobecňuje tato rozhraní (i USB je ve skutečnosti sběrnice, jak je výše uvedeno), a co se týče sběrnic na základní desce, zásuvkám obvykle říkáme slot nebo socket (patice). *Slot* má piny v jedné řadě, *socket* má piny ve formě matice.

Sběrnice mají obvykle tyto **části** (několik sad komunikačních kanálů pro různé účely):

- řídící – těmito kanály se posílají řídicí signály,
- adresová – pro přenos adres,
- datová – přímo přenáší data.

Důležitá je jejich šířka (celkem, ale i zvlášť pro výše zmíněné části) – obvykle 32 bitů, 64 bitů apod.

Komunikace po sběrnici probíhá v pravidelných intervalech, které jsou určeny timerem (časovačem). Timer je obvod, který v pravidelných intervalech generuje impuls, ten řídí činnost komponenty nebo její komunikaci po sběrnici. Rychlost generování impulsů označujeme jako frekvenci – vyšší frekvence znamená rychlejší timer a kratší časové intervaly.

Přes sběrnici se dá komunikovat *paralelně* nebo *sériově* – tyto pojmy již známe z předchozí kapitoly. Většina současných sběrnic pracuje sériově (PCI Express, USB, SATA, SAS, FireWire, Thunderbolt, RS-232, Lightening), paralelní sběrnice se už téměř nepoužívají (PCI, ISA, EISA, PATA, SCSI, LPT).

Sběrnice také dělíme na

- *synchronní* – komunikace mezi komponentami napojenými na sběrnici je řízena společným timerem, všechny pracují na stejné frekvenci,
- *asynchronní* – komponenty mají své timery, ale aby komunikace fungovala, musí se „domluvit“ (v rámci inicializace komunikace se dohodnou parametry komunikace včetně časování).

Paralelní sběrnice je vždy synchronní. Sériová komunikace může být synchronní nebo asynchronní, například USB je sériová sběrnice, která může fungovat synchronně i asynchronně.

Sběrnice dělíme do dvou typů:

- *systémová sběrnice* (CPU bus) – propojuje procesor, koprocesor, cache paměť, operační paměť, řadič cache paměti a operační paměti, grafický čip (cokoliv, co potřebuje rychlý přístup k hardwaru),
- *rozšiřující sběrnice* – umožňuje připojovat rozšiřující karty a různá vnitřní/vnější zařízení, obecně cokoliv, co napatří do předchozí odrážky.

4.3.1 ISA, EISA

ISA (Industry Standard Architecture, od IBM) je již spíše historická záležitost. Sloužila pro připojení interních rozšiřujících karet, a to v rozmezí let 1981 a 1993. Je to 8- nebo 16bitová sběrnice (jde o její šířku – kolik bitů lze zároveň přenést) s obnovovací frekvencí 8 MHz a paralelním způsobem přenosu dat. Rychlost přenosu je 8 nebo 16 MB/s podle šířky sběrnice.

EISA (Extended ISA) je potomek sběrnice ISA, v současné době také historický. Je to 32bitová sběrnice, která umožnila přenosy teoretickou rychlostí 33 MB/s, ale reálná přenosová rychlost se blížila spíše 20 MB/s.

4.3.2 PCI a PCI-X

PCI (Peripheral Component Interconnect) je rozhraní uvedené v roce 1993, které v modifikované podobě přetrvalo dodnes.

Konektory na kartách jsou fyzicky modifikovány tak, aby se omezila možnost poškození rozšiřujících karet při užití nesprávného slotu (tzv. *klíčování*), princip klíčování se používá i na novějších sběrnicích. PCI jako první sběrnice také přišla s podporou technologie Plug-and-Play.

Se sběrnicí PCI se na některých základních deskách můžeme dodnes setkat (je levná), pokud se tedy na základní desku vejde, do této sběrnice seženeme některé zvukové a síťové karty.

Další vlastnosti:

- 32/64bitová šířka sběrnice, 33/64 Mhz obnovovací frekvence s paralelním způsobem přenosu dat,
- teoretická přenosová rychlost podle verze 133 (32bit, 33 Mhz), nebo 266 (64bit, 33 Mhz), nebo 533 (64bit, 66 Mhz) MB/s,
- napájecí napětí se pohybuje v rozmezí 3,3–5 V, vyšší verze PCI pracuje s nižším napětím; použité napětí nepopleteme, protože karty vyžadující napětí 3,3 V mají vespod zářez (klíčování) na jiném místě než karty pro napětí 5 V (resp. když je zářez na obou místech, karta zvládá obojí).

PCI-X vznikla v roce 2004 jako modernizace základní specifikace PCI, došlo k dramatickému zvýšení frekvence na 133, 266 a následně 533 MHz ⇒ dosáhlo se prostupnosti 1, 2 respektive 4 GB/s. Šířka sběrnice je 64 bitů. V současné době toto rozhraní najdeme spíše u serverových komponent, například serverových síťových karet.

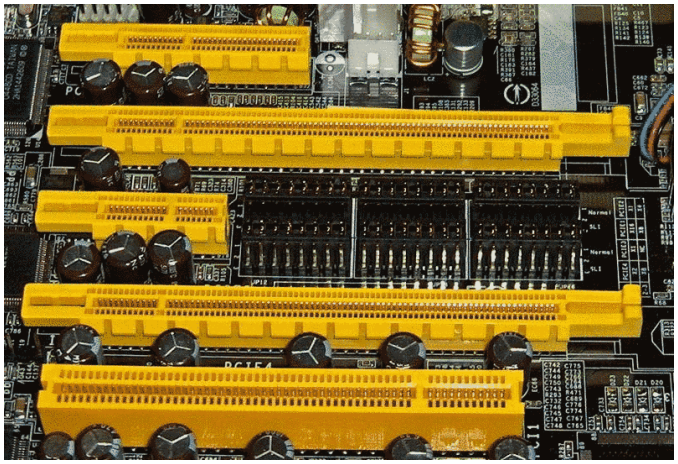
Rozhraní PCI-X je částečně zpětně kompatibilní. Karty PCI-X lze zasunout do PCI slotu, pokud je na desce podporováno napětí 3,3 V. Naopak PCI karty podporující napájení 3,3 V (PCI verze 2.1) lze provozovat v PCI-X slotu. Ale pozor, pokud takto kombinujeme PCI kartu a PCI-X slot, všechny PCI-X karty budou pracovat na „pomalé“ frekvenci PCI karty.

4.3.3 PCI Express

PCI Express (PCI-E, PCIe; rok 2004) je momentálně nejpoužívanějším sběrnicovým rozhraním pro rozšiřující karty. Základní myšlenkou je přechod od paralelního přenosu celého slova (jak je to v PCI) k sériové komunikaci.

Komunikační pásmo je rozděleno do několika kanálů, které označujeme *linky* (PCI Express Lines). Fyzické rozhraní tak může mít k dispozici 1, 4, 8, 16, 32 linek, hovoříme pak o rozhraní rychlosti ×1, ×4, atd. Nejběžnější jsou ×1 určená zejména pro zvukové a síťové karty, a ×16 pro grafické karty.

Na základní desce je celkem k dispozici určitý počet linek (typicky 16, 32 nebo 64), a tyto linky jsou pak rozděleny do jednotlivých *slotů* na základní desce. Na obrázku 4.6 vidíme čtyři sloty PCIe (a jeden PCI), do těchto čtyř slotů jsou linky rozděleny.



Obrázek 4.6: Sloty sběrnic, shora PCIe×4, PCIe×16, PCIe×1, PCIe×16, PCI[5]

Jeden kanál (linka) je tvořen celkem čtyřmi vodiči, dvěma pro každý směr. V rámci kanálu/linky se komunikuje pouze sériově, pokud je kanálů více, každý z nich přenáší nezávisle samostatnou komunikaci (přenášené slovo se nedělí mezi více kanálů a není nutno je synchronizovat na úrovni rozhraní).

PCIe sloty s různým počtem linek mají *odlišnou délku* (ostatně to je vidět na obrázku 4.6), obvykle zasazujeme kartu do slotu o takové délce, kterou potřebujeme. Nicméně lze kartu zasadit i do „delšího“ slotu, pak jsou aktivovány pouze ty linky, které karta využije, ke kterým je připojena.

Teoreticky by to mohlo jít i naopak (delší kartu zapojíme do kratšího slotu) – to však jde pouze tehdy, pokud je krátký slot upraven (na konci má rýhu, aby bylo možné delší kartu fyzicky zasunout, pak se aktivuje jen ta část pinů, která je ve slotu připojena.

Tento princip funguje i uvnitř slotu: některé sloty ×16 mají ve skutečnosti místo 16 aktivních pouze 8 linek, tedy pokud do takového slotu zasuneme kartu PCIe×16 (což samozřejmě jde), komunikuje se pouze po 8 linkách a karta pracuje pomaleji než v plnohodnotném slotu PCIe×16 (ostatně, některé karty vypadající jako ×16 stejně používají jen 8 linek).

Propustnost závisí jak na počtu linek, tak i na verzi – také u PCIe totiž rozlišujeme různé *verze*:

- verze 1.x: propustnost 250 MB/s na jednu linku a jeden směr, tj. na základní desce s 32 aktivními linkami by to bylo 8 GB/s pro každý směr, o které se dělí všechny komponenty napojené na PCIe. Pokud je grafická karta zapojena ve slotu s 16 linkami (×16), má k dispozici 4 GB/s,
- verze 2.x: propustnost 500 MB/s na linku, tedy dvojnásobek,
- verze 3.x: propustnost 985 MB/s na linku, opět dvojnásobek předchozí verze,

[5] Zdroj: https://cs.wikipedia.org/wiki/PCI-Express

- verze 4.x: propustnost 2 GB/s (1969 MB), přibližně dvojnásobek předchozí,
- verze 5.x rovněž zdvojnásobuje rychlost.

Některé komponenty využijí rychlost spíše jen v jednom směru (třeba grafická karta), jiné v obou směrech (například SSD komunikující přes PCIe).

Poznámka

Má smysl se při koupi základní desky dívat jak na verzi, tak i na počet linek v jednotlivých slotech, přičemž oba tyto parametry ovlivňují celkovou propustnost rozhraní. Slot totiž může vypadat jako 16linkový, ale může mít aktivních jen 8 linek. Ovšem když si uvědomíme, že vyšší verze zdvojnásobuje datový tok, znamená to, že 8 linek vyšší verze poskytne zhruba stejný datový tok jako 16 linek nižší verze.

Hlavní výhodou sběrnice PCIe je její rozšířenost. Ta je ovšem důsledkem několika vlastností: předně slušné rychlosti (zejména ve vyšších verzích), ale také pružnosti návrhu – výrobce komponenty se může sám rozhodnout, jaký výkon a kolik linek bude potřebovat.

Přes sběrnice jsou karty obvykle také *napájeny*. Jednodušším kartám bohatě stačí napájení přímo ze slotu PCIe, ale typicky lepší grafické karty vyžadují dodatečné napájení – o tom více v kapitole o rozšiřujících kartách.

Poznámka

Pokud jste dávali pozor, pak vám pojem PCI Express určitě něco říká – v kapitole o rozhraních je uvedeno, že rozhraní k paměťovým médiím mSATA a M.2 se napojují právě na PCI Express ×4, a Thunderbolt vlastně dělá totéž. Takže sběrnice nemusejí nutně sloužit jen k připojování rozšiřujících karet, mohou být součástí komunikační cesty jiných rozhraní.

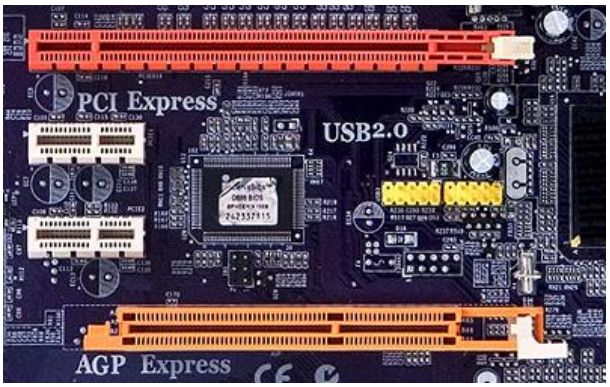
4.3.4 AGP

Sběrnice AGP (Advanced Graphics Port či Accelerated Graphics Port) vznikla roku 1997 jako rozhraní, které mělo nahradit nedostatky PCI pro grafické karty (specializovaná modifikace PCI). V té době byly k dispozici pouze sběrnice ISA, EISA a PCI (sběrnice PCIe ještě neexistovala), takže nebylo do čeho zapojit výkonnější grafické karty.

Nicméně problémem AGP byla existence více variant slotů, které se lišily kromě jiného i klíčováním (tj. na kartách byly výřezy a ve slotech můstky u různých variant na různých místech). Uživatel kupující novou kartu vždy musel vědět, do jakého typu AGP slotu ji chce použít.

Dnes sběrnici AGP prakticky vytlačila sběrnice PCIe.

Slot sběrnice AGP trochu připomíná sloty PCIe×16. Na první pohled je však poznáme podle jejich umístění na základní desce – AGP slot je vždy více odsunutý od zadní strany desky. Na obrázku 4.7 vidíme porovnání slotů AGP a PCIe×16. Horní červený je PCIe, spodní oranžový je AGP, přičemž na obrázku vlevo je zadní strana základní desky. AGP slot začíná zleva o pár centimetrů dál než PCIe.



Obrázek 4.7: Porovnání sběrnic PCIe a AGP[6]

[6] Zdroj: http://ixbtlabs.com/articles3/video/guide-p3.html

4.4 Čipset

Čipset (chipset, čipová sada) je sada integrovaných obvodů, které zajišťují především komunikaci mezi jednotlivými komponentami (procesor, paměti, rozšiřující karty, BIOS, nejrůznější rozhraní a konektory), jedná se o vlastně o jakési komunikační centrum. Také obsahuje některé řadiče, například může (nemusí) obsahovat řadič paměťových modulů, řadič USB, řadič SATA.

Čipset určuje, jakým způsobem budou jednotlivé komponenty mezi s sebou logicky i fyzicky propojeny. Na kvalitě čipsetu do značné míry závisí kvalita samotné základní desky, protože právě tudy vedou hlavní komunikační cesty komponent.

Nejznámější výrobci čipsetů jsou Intel, AMD, VIA, SiS, nVidia (čipsety nForce). Tyto čipsety jsou pak používány tak jak jsou různými výrobci základních desek.

Dnes se používají dvě základní koncepce – North-South Bridge Design a One Chip Design. Rozdíl mezi nimi je v počtu čipů (obecně částí), které tvoří čipset. Existují také jejich modifikace, například z konceptu One Chip Design se vyvinuly dnes nejpoužívanější čipy pro mobilní zařízení – SoC.

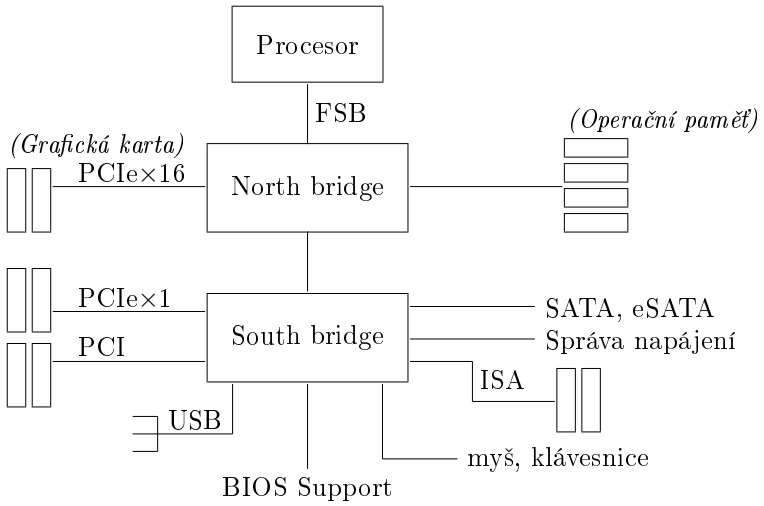
4.4.1 North-South Bridge Design

Koncepce *North-South Bridge Design* se dnes používá u většiny počítačů, notebooků a serverů. Čipset je rozdělen do dvou čipů, které nazýváme North Bridge a South Bridge (severní a jižní most/můstek).

- *North Bridge* (severní most) obstarává vysokorychlostní komunikaci s grafickou kartou, operační pamětí a procesorem, je blíže k procesoru,
- *South Bridge* (jižní most) obstarává komunikaci s ostatními komponentami: s rozšiřujícími kartami, disky, řadiči externích rozhraní a různými mechanikami, a zprostředkovává služby BIOSu. Určuje maximální přenosovou rychlost disků, USB, atd. Bývá umístěn blíže slotům pro rozšiřující karty.

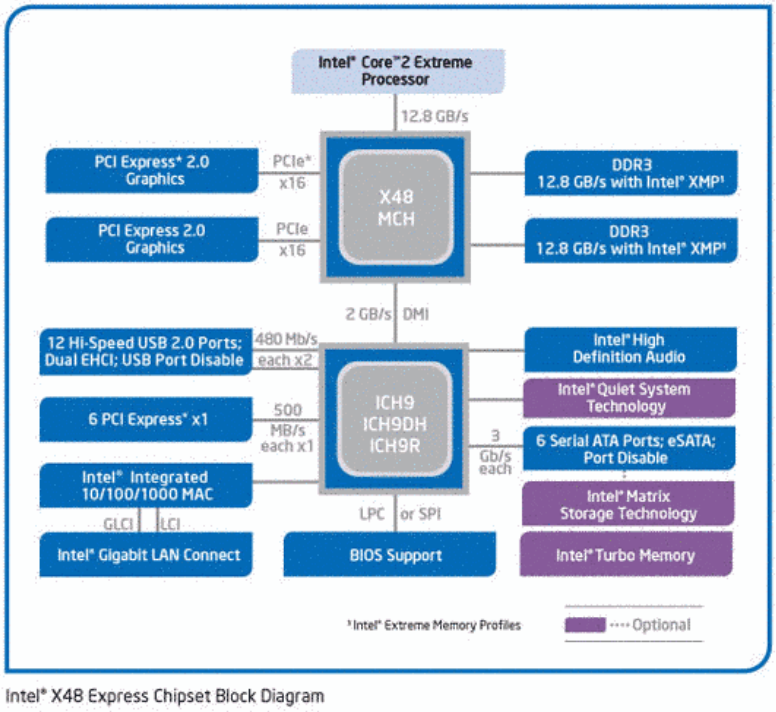
Pro jižní most se také používá název *I/O Controller Hub* (nepřesně česky vstupně-výstupní řadič).

Obecné schéma čipsetu se sběrnicí FSB je na obrázku 4.8, na obrázku 4.9 je již konkrétní čipset od společnosti Intel pro procesor Core 2 Extreme, na obrázku 4.10 je schéma čipsetu pro základní desky s procesory do patice AM3 (od AMD).



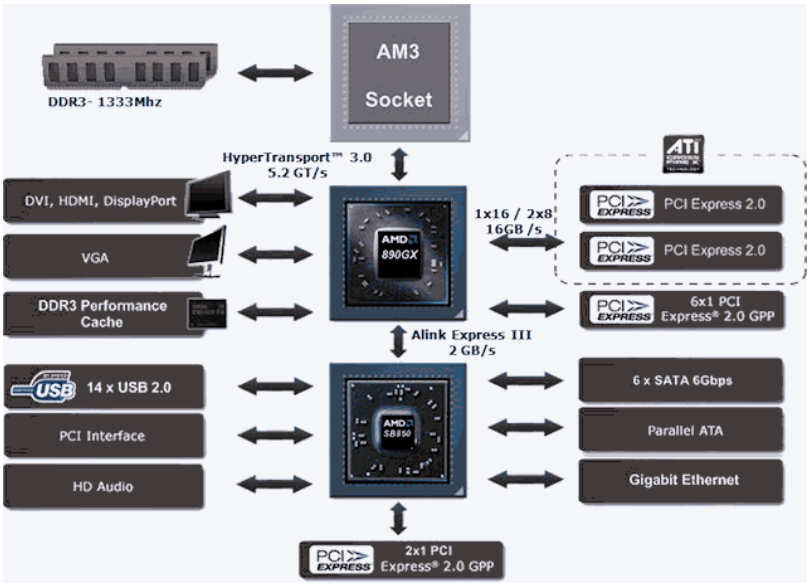
Obrázek 4.8: Obecné schéma čipsetu se sběrnicí FSB

Tato schémata se obvykle dají najít na internetu, stačí zadat například „chipset block diagram“ (nebo místo „chipset“ přímo název konkrétního čipsetu) a zvolit vyhledávání obrázků.



Intel® X48 Express Chipset Block Diagram

Obrázek 4.9: Schéma čipsetu X48 Express pro procesor Intel Core 2 Extreme[7]



Obrázek 4.10: Schéma čipsetu společnosti AMD pro procesory do patice AM3[8]

Poslední dobou jsou tendence integrovat některé funkce severního mostu, anebo dokonce celý severní most, do čipu procesoru, především z důvodu urychlení komunikace s operační pamětí a lokálními sběrnicemi, úspory místa na základní desce a snadnějšího uchlazení.

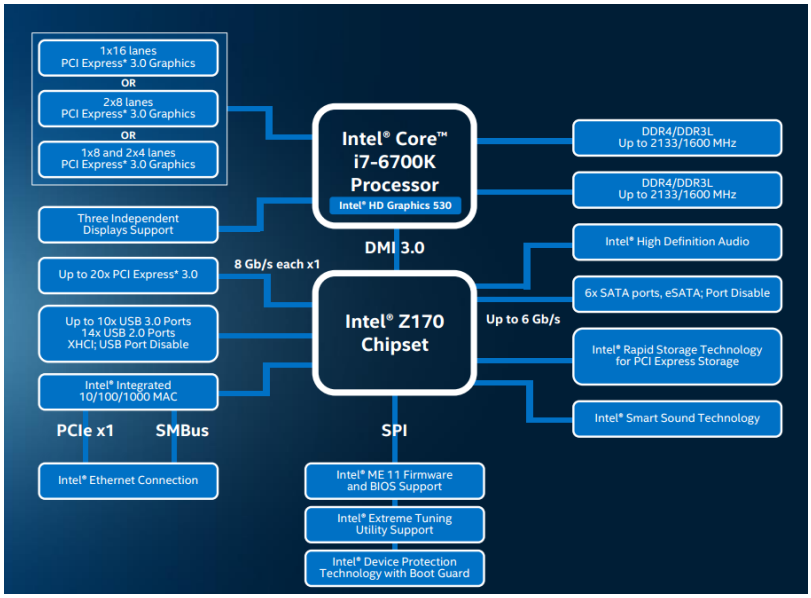
Funkce *paměťového řadiče* (tj. komponenty, která zprostředkovává komunikaci procesoru s moduly operační paměti), která dříve běžně bývala v severním mostě, se přestěhovala do pouzdra procesoru. S tímto

[7] Zdroj: http://ixbtlabs.com/articles3/mainboard/nvidia-nforce-790i-intel-x48-chipsets-p2.html, zajímavý popis najdeme na http://techreport.com/articles.x/14316

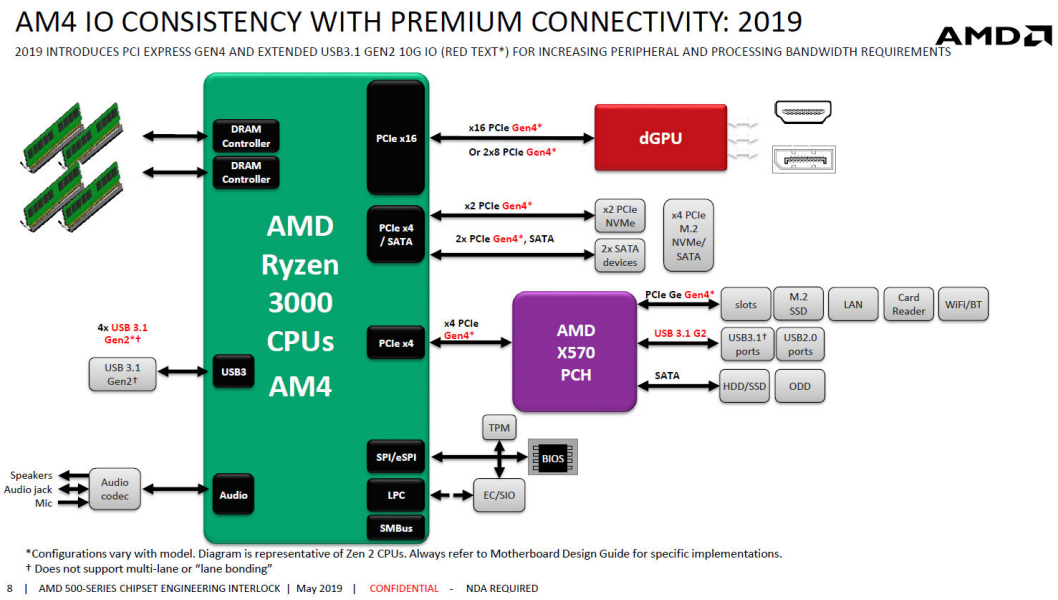
[8] Zdroj: http://www.abclinuxu.cz/blog/virtualizace/2010/8/hardware-3-vyber-chipsetu-a-zakladni-desky

trendem začala společnost AMD u 64bitových procesorů, po několika letech se k tomuto trendu připojila také společnost Intel.

Takže paměťový řadič byl prvním „uprchlíkem“. Dalším stupněm je přestěhování celého severního mostu do pouzdra procesoru – modul severního mostu v procesoru se nazývá *System Agent* (u Intelu).



Obrázek 4.11: Schéma čipsetu pro procesory Intel Skylake[9]

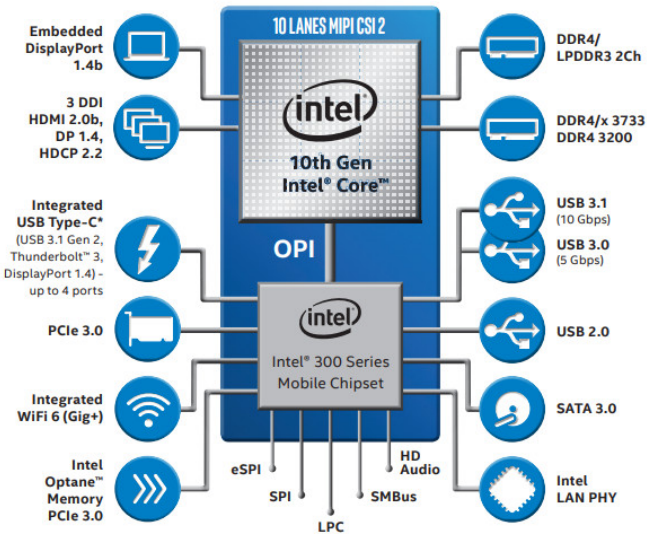


Obrázek 4.12: Schéma čipsetu X570 pro procesory AMD Ryzen 3. generace (rok 2019)[10]

Když srovnáme obrázky 4.9, 4.10 a 4.11, zjistíme, že:

- obrázek 4.9 plně odpovídá základnímu schématu z obrázku 4.8 (tj. v centru schématu jsou tři velké čipy – nahoře procesor, pod ním severní most MCH X48 a ještě níže jižní most ICH9, procesor je

[9] Zdroj: http://www.anandtech.com/show/9483/intel-skylake-review-6700k-6600k-ddr4-ddr3-ipc-6th-generation/5
[10] Zdroj: https://www.cnews.cz/oficialni-schema-amd-x570-ryzen-3000-konektivita-pcie4-0-usb3-1-gen2/
[11] Zdroj: http://linuxgizmos.com/intel-launches-first-10th-gen-ice-lake-cpus-with-10nm-fabrication/



Obrázek 4.13: Schéma čipsetu pro procesory Intel Ice Lake (rok 2019)[11]

přímo připojen jen k severnímu mostu, k severnímu mostu jsou zprava připojeny moduly operační paměti – DDR3, zleva sloty pro grafickou kartu – PCI Express 2.0),

- na obrázku 4.10 se sice stále počítá s tím, že máme čipset ve dvou velkých čipech mimo procesor, ale paměti (taktéž DDR3) jsou již připojeny k procesoru, nikoliv k severnímu mostu (takže část severního mostu – paměťový řadič – se stěhuje do procesoru),
- na obrázku 4.11 je již celý severní most integrován do pouzdra procesoru, tedy všechno to, co na obrázku 4.9 bylo připojeno k severnímu mostu, je teď připojeno k procesoru.

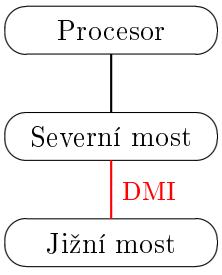
Na obrázcích 4.12 a 4.13 jsou čipsety pro procesory AMD Ryzen 2. generace a Intel Ice Lake. Severní most je integrován v procesoru. Všimněte si, že AMD už podporuje PCIe 4. generace, Intel ještě ne.

Úkol

Prohlédněte si všechny nákresy čipsetů včetně všeho, co je na nich připojeno k severnímu a jižnímu mostu. Zjistíte, že mnohé vám už něco říká – s těmito pojmy jsme se seznámili v této a předchozí kapitole.

4.4.2 Čipset a sběrnice

Na výše uvedených obrázcích čipsetů jste si určitě všimli, co je s čím propojeno a jak jsou tato propojení označena. Procesor s čipsetem a jednotlivé části čipsetu vzájemně komunikují pomocí speciálních sběrnic, se kterými se teď seznámíme.



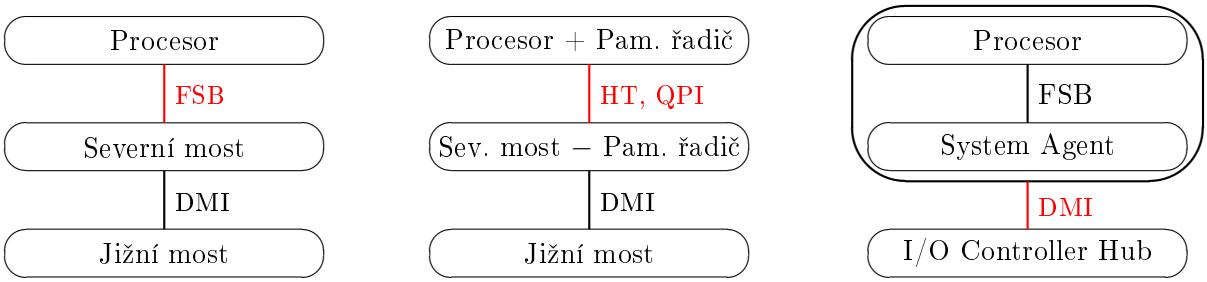
Obrázek 4.14: Sběrnice DMI

Mezi severním a jižním mostem existuje spojení s vyšší rychlostí přenosu. U intelovských architektur se tato sběrnice označuje *DMI* (Direct Media Interface) a jde o modifikaci sběrnice PCIe×4, případně zde přímo najdeme PCIe×4. U základních desek AMD se označuje ALink Express (založena na PCIe×4) nebo opět přímo PCIe×4.

Velmi důležitá je sběrnice zajišťující komunikaci *mezi severním mostem a procesorem*. Její rychlost má zásadní vliv na rychlost celého systému. Mezi čipsetem a procesorem se můžeme setkat především s těmito sběrnicemi:

- *FSB* (Front Side Bus) je původní označení této sběrnice v klasické formě, kdy je celý severní most v samostatném čipu,

- *HT* (HyperTransport) – sběrnice na architekturách AMD (od architektury K8), řadič paměti byl přenesen ze severního mostu do procesoru, což znamená zvýšení propustnosti této sběrnice (a tím i rychlosti) jednoduše tím, že je po ní přenášeno méně dat, navíc procesor může s operační pamětí komunikovat velmi rychle, když to nemusí být přes další čip,
- *QPI* (QuickPath Interconnect, starší název CSI) je intelovská sběrnice pro výkonnější sestavy, řadič paměti byl přenesen procesoru (tj. obdoba AMD HT),
- *DMI* – u Intelu v případě, že je severní most integrován do procesoru.



Obrázek 4.15: Sběrnice mezi severním mostem (resp. čipsetem) a procesorem

Ve všech případech se však můžeme setkat s označením FSB, jde totiž také o základní označení sběrnice propojující čipset a procesor. Na obrázku 4.15 na straně 65 je vývoj této sběrnice (včetně přesouvání příslušných funkcionalit) názorně zachycen.

4.4.3 One Chip Design

V konceptu *One Chip Design* jsou oba mosty integrovány do jednoho čipu. Jde o řešení, které se snaží minimalizovat náklady na výrobu zařízení nebo množství zabraného místa a je tedy určeno spíše pro levná či menší zařízení.

Výhody:

- relativně malé rozměry, šetří se místo na základní desce (jeden čip zabere méně místa než dva),
- cena i více než o polovinu nižší než u konceptu North-South bridge,
- výrazně nižší spotřeba, atd.

Nevýhody: omezená modularita (nelze kombinovat různé severní a jižní mosty) a obvykle nižší výkon.

Omezená modularita již dnes není takový problém a výhoda malých rozměrů, nižší ceny a spotřeby tuto nevýhodu bohatě převažuje především v malých zařízeních, takže se s konceptem One Chip Design setkáváme jak v malých mobilních zařízeních (smartphonech, tabletech), tak i v embedded zařízeních (především v obývákové elektronice), případně HTPC (multimediálních centrech). Ukázku schématu najdeme na obrázku 4.16.

Poznámka

Pozor, nepleťte si One Chip Design s novější formou konceptu North-South Bridge, ve které je severní most integrován do procesoru. V obou případech sice máme na základní desce dva velké čipy, z nichž jeden je procesor, ale u North-South Bridge Designu je tím druhým čipem jižní most (tj. jen část čipsetu), kdežto u One Chip Designu je v druhém čipu celý čipset.

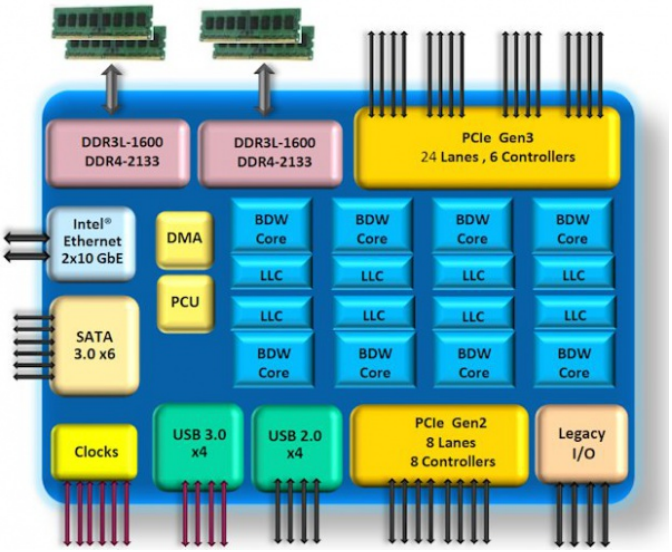
Další informace

Zajímavou recenzi základní desky s jednočipovou sadou NVidia MCP7A (FormFactor Mini-ITX, s integrovanou grafikou a procesorem Atom 330 se dvěma jádry) najdeme na adrese
http://www.thinkcomputers.org/asrock-a330ion-nvidia-mcp7a-ion-mini-itx-motherboard-review/.

4.4.4 SoC čipy

SoC (System on Chip) je vlastně speciálním případem One Chip Designu, kde oproti „mateřskému“ konceptu došlo k těmto změnám:

- celý čipset (který byl původně v jednom samostatném čipu na základní desce) se stěhuje do pouzdra procesoru, tedy čipset je integrován do procesoru (nebo procesor do čipsetu, záleží na úhlu pohledu),
- do tohoto (nyní už jediného) velkého čipu se dále integruje všechno, co jen je technicky možné.



Obrázek 4.16: Schéma SoC čipu Xeon D z roku 2015[12]

V jednom pouzdře čipu je tedy procesor, čipset, generátor časových impulsů, časovače, řadiče komunikačních rozhraní (USB, Ethernet, podle potřeby), grafika, apod., tedy jak původní čipset, tak i další moduly, které bychom jinak našli v samostatných čipech na základní desce, případně paměti.

Tento trend se nejdřív objevil u embedded zařízení, ale postupně se dostal i do levnějších běžných zařízení. V současné době využívají SoC čip prakticky všechna malá mobilní zařízení a také většina zařízení Internetu věcí (IoT).

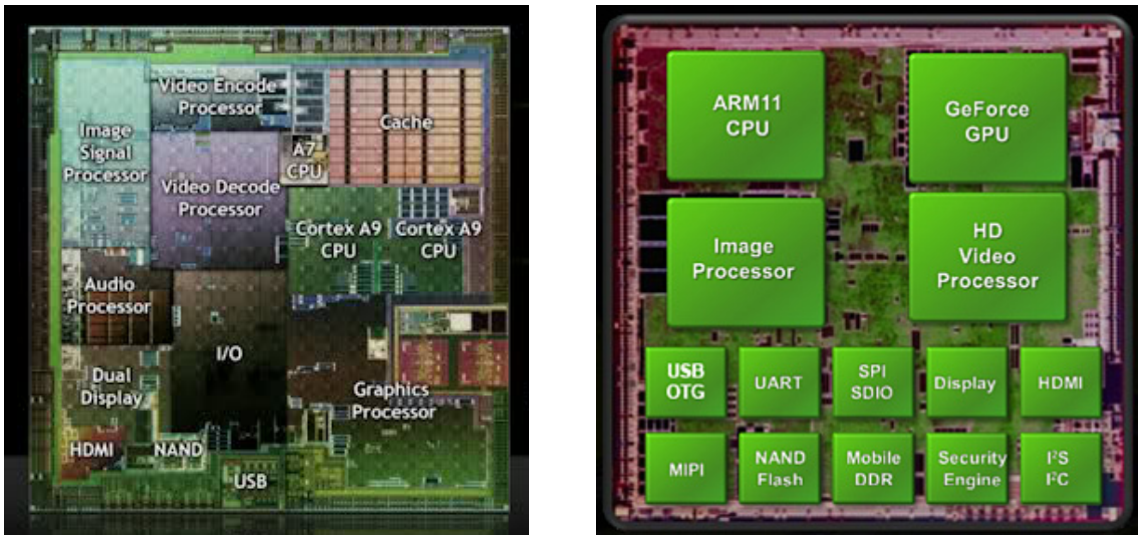
SoC čipsety/procesory dřív vyráběly především společnosti Qualcomm (Snapdragon), Apple (pro svá mobilní zařízení), Broadcom, Samsung, Transmeta, nVidia (procesory Tegra), Atmel, v současné době už existují také SoC procesory společností AMD (AMD Geode) a Intel.

Ale pozor, SoC čipy nemusejí nutně být určeny pro levná zařízení. Na obrázku 4.18 vidíme blokové schéma SoC čipu Ampere z roku 2018 (typu ARM) určeného pro servery, vyráběného 16nm výrobním procesem. Má 32 výpočetních jader, podporuje až 1 TB operační paměti a nabízí 42 linek PCIe verze 3. Mimochodem, zakladatelka společnosti Ampere je bývalou prezidentkou Intelu.

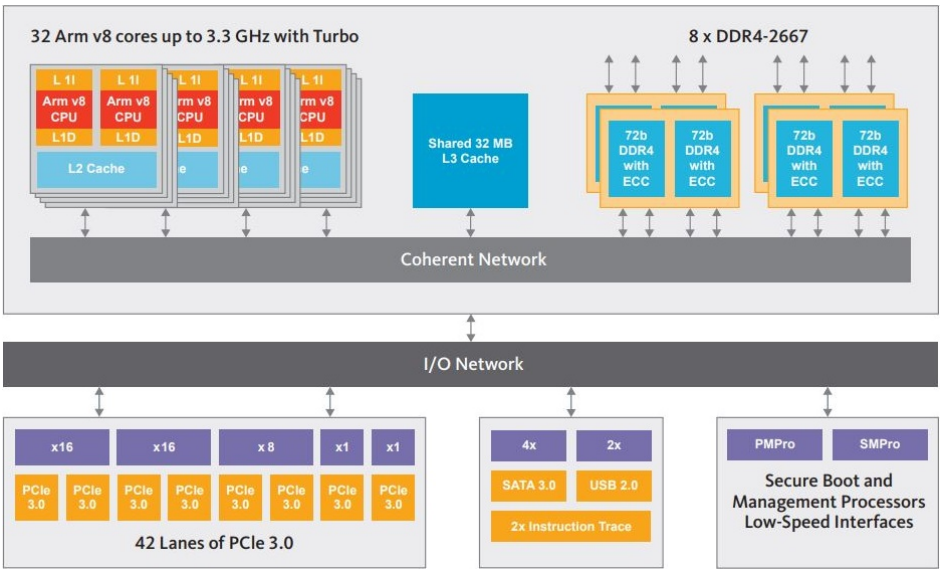
Poznámka

Většina SoC čipů je typu ARM (z výše uvedených opravdu téměř všechny). Co to znamená? Společnost ARM Holdings přímo tyto procesory nevyrábí, pouze je vyvíjí, a následně poskytuje v licenci svým partnerům specifikaci k jejich výrobě. Tedy například procesory Qualcomm Snapdragon, mobilní Apple, Samsung Exynos, nVidia Tegra apod. jsou všechny postaveny na architektuře ARM.

[12] Zdroj: https://www.pcper.com/category/tags/xeon-d
[13] Zdroje: http://i48.tinypic.com/o7t3y9.jpg, http://www.xbitlabs.com/news/mobile/display/20080603141353_Nvidia_Unleashes_Tegra_System_on_Chip_for_Handheld_Devices.html
[14] Zdroj: https://www.servethehome.com/ampere-32-core-64-bit-arm-chip-x-gene-3-ip/



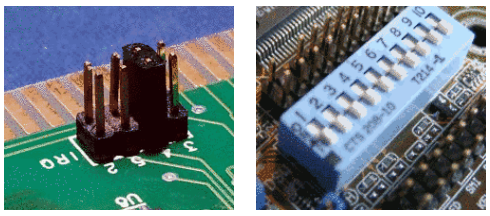
Obrázek 4.17: SoC procesor ARM Cortex A9 a SoC procesor NVidia Tegra s procesorem ARM 11[13]



Obrázek 4.18: SoC procesor Ampere typu ARM určený pro servery[14]

4.5 Přepínače, propojky

Prepínače (switche) na základní desce mají dvě polohy – zapnuto (on) a vypnuto (off). Slouží k hardwarové konfiguraci takových vlastností základní desky, které mají dva stavy. Přepínač se vyskytuje ve více různých formách – jednu z forem vidíme na obrázku 4.19 vpravo.



Obrázek 4.19: Propojky a přepínače[15]

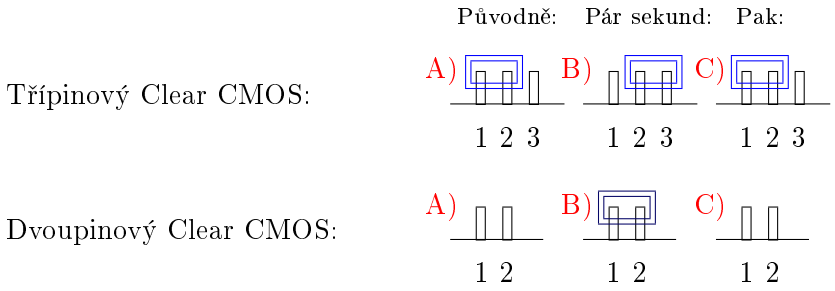
Propojky (jumpery) na rozdíl od přepínačů mohou mít více různých stavů. Obvykle jde o blok kolíků (pinů) uspořádaných do matice, na které je možné nasunout propojovací klobouček s vodivým můstkem. Klobouček je obvykle na dva sousední kolíky, a podle toho, které dva kolíky z bloku jsou takto propojeny, je zapnutá jedna konkrétní z možných voleb.

[15] Zdroj: http://www.infopackets.com/news/hardware/2004/20041228_install_sound_card_to_computer_with_onboard_integrated_sound.htm

Poznámka

Pokud na dva sousední piny nasuneme klobouček, pak je takzvaně *zkratujeme* (vytvoříme vodivou cestu, která před nasunutím kloboučku neexistovala).

Přepínače a jumpery slouží na základních deskách k důležitým nastavením, která se obvykle za provozu počítače nemění. Dnes, v době mechanismu Plug & Play, jsou v mnoha případech (ale ne vždy) nahrazovány daleko pohodlnějšími auto-detekčními algoritmy.



Obrázek 4.20: Reset BIOSu pomocí propojek

Přepínačem nebo propojkou může být také specializovaný *Clear CMOS*. Má za úkol vymazat uživatelská nastavení BIOSu a uvést jej do továrního nastavení (reset BIOSu), což je stejný efekt jako vytažení záložní baterie udržující nastavení BIOSu. U většiny základních desek jde o třípinovou propojku, kterou vidíme na obrázku 4.20. Standardně je propojka nastavena na piny 1,2. Po zapojení na 2,3 (při vypnutém počítači) se CMOS vymaže, pak bychom neměli zapomenout vrátit propojku na piny 1,2. Postup se může lišit u různých základních desek, konkrétní postup najdeme v dokumentaci k základní desce.

Kromě použití pinů nebo vyndání baterie nabízejí některé základní desky i jiné možnosti, může na nich být třeba tlačítko Clear CMOS (takže stačí stisknout tlačítko).

4.6 Rozvržení komponent na základní desce

Určité problémy mohou vzniknout špatným rozvržením prvků na základní desce:

- chlazení – k socketům, konektorům, atd. na základní desce se připojují různé komponenty, které při nevhodném rozmístění příslušných připojovacích prvků mohou překážet proudění vzduchu,
- špatné osazení napájecích obvodů (teplo), nevhodné chlazení čipových sad (nevhodný materiál, chladicí materiál špatně přilnul na čipy, atd.), umístění datových a napájecích kabelů (překážejí proudění vzduchu),
- připojované komponenty si mohou navzájem překážet – například socket pro procesor a patice pro paměťové moduly: některé procesory s chladičem po osazení „tlačí“ na paměťové moduly, někdy je problém s umístěním rozšiřujících karet a různých konektorů.

Další informace

- Intel Chipsets: https://ark.intel.com, tlačítko *Chipsets*
- AMD Chipsets: https://www.amd.com/en/products/chipsets-motherboards-desktop
- List of Intel Chipsets: https://en.wikipedia.org/wiki/List_of_Intel_chipsets
- List of AMD Chipsets: https://en.wikipedia.org/wiki/List_of_AMD_chipsets

- Jak se vyrábí základní deska – Gigabyte (video, možno zapnout české titulky):
  https://www.youtube.com/watch?v=Va3Bfjn4inA
- Okomentovaný nákres základní desky:
  - http://www.laptopparts101.com/system-board-motherboard-mainboard/
  - http://hothardware.com/Tags/itx.aspx
  - http://technet.idnes.cz/ktera-zakladni-deska-je-pro-vase-pc-nejlepsi-s-nami-to-poznate-pt1-/hardware.asp?c=A080409_101120_hardware_mbo
- Základní desky na CNews
  http://www.cnews.cz/tagy/zakladni-desky-0
- Jak vybírat základní desku na Svět Hardware
  http://www.svethardware.cz/art_doc-670D52C33DF691CFC1256E83004CC0B9.html?lotus=1&Highlight=0,jak,vyb%C3%ADrat,z%C3%A1kladn%C3%AD,desku

Úkoly

1. Zjistěte informace o základní desce ASUS PRIME X299-DELUXE II. Můžete použít například tyto zdroje:
   - stránky výrobce základní desky,
   - https://techreport.com/review/34295/asus-prime-x299-deluxe-ii-motherboard-reviewed/
   - některý obchod s počítačovými komponentami.

   Všímejte si především Form Factoru, patice procesoru, patic na sběrnicích (paměti, PCIe, PCI, SATA, atd.), čipu BIOSu, apod.
2. Podobně zjistěte informace o základních deskách MSI MPG X570 GAMING PLUS a ASUS ROG ZENITH EXTREME. Zjistěte také obvyklé ceny těchto základních desek.
3. B360 (od Intelu) je typickým příkladem čipsetu se severním mostem integrovaným v procesoru. Najděte na webu schéma tohoto čipsetu (můžete použít třeba klíčový výraz „Intel B360 schema“, hledejte mezi obrázky). Všimněte si, jak tento čipset komunikuje s procesorem, jaké jsou rychlosti připojených linek a co vše lze připojit.

Kapitola 5

Procesory

Rychlý náhled: V této kapitole se budeme zabývat nejdůležitější komponentou počítače, jeho srdcem – procesorem. Bude nás zajímat, jak procesor funguje, jaké jsou typické vlastnosti procesorů a jaké procesory v současné době existují.

Klíčová slova: Procesor, CPU, APU, IGP, MCU, DSP, FPGA, ASIC, technologie výroby, litografický proces, matematický koprocesor, Mooreův zákon, řadič, ALU, FPU, registr, instrukční sada, AVX, AES, L1, L2, L3 cache, chráněný (privilegovaný) režim procesoru, NX bit, XD bit, CISC, RISC, pipelining, skalární architektura, Out-of-Order Execution, HyperThreading, SMT, virtualizace, TDP, Turbo Boost, Turbo Core, PGA, ZIF, BGA, LGA, architektura a mikroarchitektura procesoru, Intel, AMD, ARM, IRQ, DMA, I/O porty, adresy paměti zařízení.

Cíle studia: Po prostudování této kapitoly budete vědět, z čeho se skládá a jak funguje procesor, jaké existují instrukční sady, jaké vlastnosti mají současné procesory, do jakých patic na základní desce patří. Získáte přehled o současných výrobcích procesorů a jejich produktech, naučíte se se v jejich nabídce orientovat a vybírat procesor podle budoucího využití.

5.1 Co je to procesor

Procesor je složitý integrovaný obvod (milióny či miliardy tranzistorů), je to křemíková destička (no, s tím křemíkem to není tak doslova), do které jsou v několika vrstvách vryty spoje obvodů fotoelektrickým procesem.

Mikroprocesor je procesor integrovaný do obvodu na jediné destičce v jediném pouzdře, pracuje univerzálně (není specializován na určité typy úloh). Dále pod pojmem procesor budeme rozumět mikroprocesor a budeme ho také označovat zkratkou CPU (Central Processing Unit).

Pokud je do mikroprocesoru integrováno i grafické jádro (GPU), pak se označuje jako *APU* (Accelerated Processing Unit, u procesorů společnosti AMD) nebo *IGP* (Integrated Processing Unit, u Intelu).

Existují ještě tři další druhy procesorů:

- *MCU* (Micro Controller Unit) – vyrábějí se pro určitou konkrétní činnost (specializované), ale mohou být také univerzální, nízká cena, malé rozměry, nízká spotřeba, menší výkonnost, malá možnost rozšíření.

Tyto jednoduché levné procesory se dnes používají do jednodušších elektrozařízení (mikrovlnky, pračky, alarmy, semafory, apod.), pracují jako řadiče (disků, tiskáren, klávesnic, monitorů, řídí větráky, atd.).

- *DSP* (Digital Signal Processor) – obdoba MCU (taky specializované na určité typy úkolů), ale s vyšším výkonem, optimalizované pro zpracování velkého množství dat stejného typu.

  Jsou dobré hlavně ve zpracování matematických funkcí na velkých objemech dat, součástí bývají také digitálně-analogové a analogově digitální převodníky. V počítačích se používají jako pomocné procesory, například u zvukových karet.
- FPGA (Field Programmable Gate Array) jsou maximálně přizpůsobivé čipy. Zatímco jiné typy čipů jsou „naprogramovány“ z výroby (mají výrobcem danou instrukční sadu), FPGA lze konfigurovat, tedy určovat, co konkrétní čip má umět. Tyto čipy najdeme spíše v robotických zařízeních, kde je programovatelnost čipu prakticky nezbytná.
- ASIC (Application-Specific Integrated Circuit) je výpočetní čip upravený a optimalizovaný pro konkrétní účel. Ve skutečnosti jsou například FPGA taky typem ASIC, ale obvykle se o nich takto neuvažuje. Existují například specifické ASIC čipy pro těžbu konkrétních kryptoměn, také se používají pro hardwarovou podporu zpracovávání síťového provozu na síťových zařízeních, šifrování, akcelerometry a jiné speciální čipy v automobilech, apod.
- NPU (Neural Processing Unit) může sice být samostatný čip, ale dnes spíše vyskytuje jako součást „většího“ výpočetního čipu. NPU jsou jednotky pro podporu instrukcí využívaných umělou inteligencí, neuronovými sítěmi apod.

Poznámka

V počítačích vlastně obvykle máme více procesorů – hlavní mikroprocesor a pak určitě ještě několik DSP nebo MCU (zvuková karta, případně grafická a další, v klávesnici, pak řadiče všeho druhu).

Fyzická struktura procesorů a vlastně každého čipu se blíží fyzické struktuře základních desek, kterou jsme popsali výše. Základem je křemíkový *wafer* („oplatek“), který se vyrobí nařezáním křemíkových ingotů na velmi tenké destičky, které se musí dokonale vyleštit. Současné procesory obvykle mají i několik desítek vrstev. Následně se *fotoelektrickým procesem* (taky hovoříme o litografii) nanesou obvody:

- pro každou vrstvu se připraví *fotolitografická maska* coby šablona schématu obvodů, masky se vytvářejí ve specializovaných CAD systémech,
- na wafer se nanesou potřebné vodivé a izolační materiály,
- maska se přiloží a celek se vystaví ultrafialovému záření, čímž oblasti nezakryté maskou změknou,
- použije se chemické rozpouštědlo, které naleptá pouze změklé oblasti,
- totéž se provádí pro další vrstvy obvodů.

5.1.1 Technologie výroby

Technologie výroby (manufacturing process) znamená, jak vypadají tranzistory, jak jsou uspořádány, propojeny a napájeny, jaké materiály jsou použity, jaká vlnová délka UV záření se používá v litografickém procesu, jak je UV záření usměrňováno, a z toho vyplývá, jak velké jsou tranzistory. Na konkrétní vlnové délce ultrafialového záření závisí, jaké materiály mohou být použity, jak „hustá“ může být maska a tedy

jak moc mohou být vytvářené obvody stěsnány, ale například také na jak vysoké frekvenci může výsledný procesor pracovat.

Zjednodušeně se hovoří o xxx-nanometrové technologii výroby (v angličtině často „litography"), protože tento údaj souvisí s vlnovou délkou UV záření, materiálem i tvarem tranzistoru.

Příklad

Například u Intelu se setkáme s těmito údaji:

- „prehistorický" procesor i4004 byl vyroben 10μm technologií (mikrometry),
- pro Core 2 je to 65 nebo 45 nm (nanometrů),
- procesory Core i7, i5 a i3 první a druhé generace jsou vyráběny 32nm nebo 45nm technologií,
- procesory Core i7, i5 a i3 třetí a čtvrté generace jsou vyráběny 22nm technologií,
- procesory Core i7, i5 a i3 páté a šesté generace jsou vyráběny 14nm technologií,
- sedmá generace je vyráběna s litografií 14+, osmá a devátá 14++,
- desátá generace je částečně vyráběna litografií 14++, částečně novějším 10nm procesem.

Pro představu: u procesorů vyráběných 45nm technologií jsou masky osvicovány UV zářením o vlnové délce 193 nm, pro procesory vyráběné 22nm a novější technologií se používá vlnová délka 13,5 nm.

AMD vyrábí své nové procesory Ryzen třetí generace 7nm litografií (ve skutečnosti pro AMD ty procesory vyrábí TSMC). V případě ARM je to složitější, každý výrobce to má jinak. Nejdále je zatím Samsung, přibližně na úrovni AMD.

Některé nejnovější procesory již nemají litografii uváděnou v nanometrech, ale v nové (menší) jednotce – angstromech. Jeden angstrom je desetina nanometru, tedy například litografie 14A vlastně odpovídá 1,4 nm. Účelem je snadnější reprezentace menších rozměrů.

Další informace

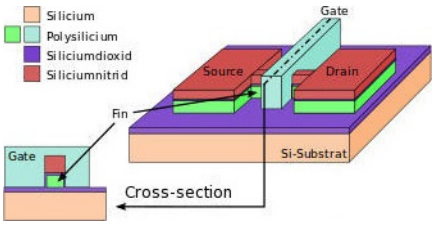
- Výroba procesorů u Intelu:
  - Intel Factory Tour – 32nm Manufacturing Technique:
    https://www.youtube.com/watch?v=SeGqCI3YAaQ
  - Intel: The Making of a Chip with 22nm/3D Transistors | Intel:
    https://www.youtube.com/watch?v=d9SWNLZvA8g
- Výroba procesorů u AMD (video, animace): https://www.youtube.com/watch?v=qLGAoGhoOhU
- Angstromy u tranzistorů: https://www.youtube.com/watch?v=bL3swp8-2oY
- Přehled některých technologií najdeme například na
  (článek je z roku 2006, tedy již ne zrovna nejaktuálnější, ale vysvětluje princip celkem dobře).

5.1.2 Tranzistory

Tranzistor je drobná polovodičová součástka, která umožňuje pomocí malé změny napětí na vstupu vyvolat velkou změnu napětí nebo proudu na výstupu.

V procesorech najdeme unipolární tranzistory (taky se označují FET – Field Effect Tranzistor), kde je průtok proudu řízen elektrodou, které říkáme *brána*. Brána buď propouští elektrony nebo ne, podle napětí elektrického pole, kterým bránu řídíme (od toho označení FET).

Proud prochází ve směru od elektrody „source“ (S, emitor) a „drain“ (D, kolektor), přičemž může/nemusí být propuštěn bránou (gate, G, taky hradlo). To vše je na povrchu (substrátu) z polovodiče. Elektroda G je také označována jako vstupní elektroda, protože přes ni je ovlivňován průtok proudu. Takže mezi elektrodami S a D vede kanál pro elektrony, který může být přerušen elektrodou G. Na obrázku 5.1 je schéma tranzistoru s označením těchto částí.



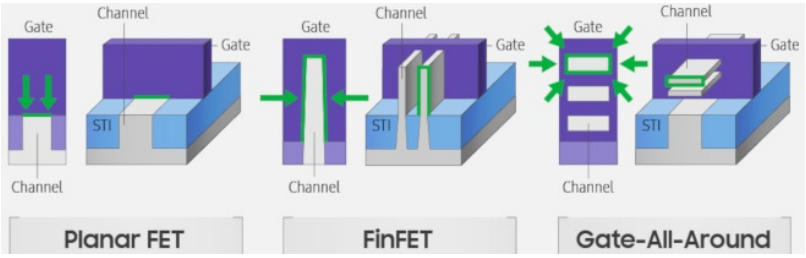
Obrázek 5.1: FinFET tranzistor[1]

A co tedy ten tvar tranzistoru? Dříve se používaly *planární tranzistory* (Planar FET), ve kterých je brána umístěna nad kanálem mezi S a D, a napětí jejího elektrického pole ovlivňuje průchodnost kanálu, který vede substrátem (základnou). Planární tranzistor vidíme v levé třetině obrázku 5.2.

Jenže jak se výrobci bližili 20nm technologii, začal se výrazně projevovat tzv. *tunelový jev* (anglicky „short-channel effect“, protože s menšími tranzistory se S a D bliží k sobě a kanál je čím dál kratší, brána tedy čím dál tenčí). Brána začala být tak tenká, že i ve stavu uzavření přes ni pronikaly elektrony.

Pak se tedy u výrobní technologie přešlo na *FinFET tranzistory* (protože slovo „fin“ znamená ploutev, a tvar kanálu zde opravdu ploutev připomíná). Kanál je nadzvednutý nad substrát a brána jej obklopuje ze tří stran, snížilo se tedy riziko tunelového jevu (ovšem zase jen do určité doby). Kvůli nadzvednutí nad substrát a tedy přechodu do „dalšího rozměru“ také hovoříme o 3D tranzistorech. Navíc kanál může být rozdělen do dvou či tří proudů (vlastně ploutví), aby vliv brány byl co největší. FinFET tranzistor je na obrázku 5.2 uprostřed. Na menších obrázcích je naznačen směr působení elektrického pole brány. Hranice použitelnosti FinFET je před 3nm procesem.

Tuto hranici by měly překonat tranzistrory se strukturou *GAA-FET* (Gate-all-around FET), resp. u Samsungu s názvem *MBCFET* (vidíme na obrázku 5.2 vpravo), kde brána obklopuje celý kanál (resp. všechny ploutve) ze všech stran.



Obrázek 5.2: Typy tranzistorů: planární, FinFET, GAA-FET[2]

Kromě struktury tranzistoru je samozřejmě důležitý i materiál. Jde nejen o to, aby materiál měl příslušné „užitné“ vlastnosti, když má pracovat jako tranzistor, resp. integrovaný obvod, ale také aby výroba obvodů pomocí fotoelektrického procesu byla proveditelná.

V elektronice včetně procesorů se můžeme setkat například s technologií MOSFET (Metal Oxide Semiconductor Field Effect Tranzistor), kdy je elektroda brány vyrobena z kovu (Metal) a potažená izolační vrstvou oxidu křemičitého (to je ten Oxide v názvu), která odděluje kov od polovodiče (substrátu). Jako kov se dřív používal hliník, později to byl upravený křemík – polycrystalline silicon (i když to už neladilo s „M“, přesto se tomu pořád říkalo MOSFET), ale postupně se přešlo k jiným kovům.

[1] Zdroj: https://www.computerhope.com/jargon/f/finfet.htm

[2] Zdroj: https://www.cdrinfo.com/d7/content/samsung-electronics-announced-mbcfet-transistor-structure-advanced-foundry-nodes

MOSFETy se poměrně hodně rozšířily, kromě procesorů a paměťových čipů se používají také při realizaci napájení, v nejrůznějších senzorech, spotřební elektronice,...

MOSFET technologie taky není jen jedna. Například u čipů se hodně používá typ MOSFETu zvaný CMOS. Dnes se CMOS používá kromě procesorů pro paměťové čipy (například konfigurace BIOSu je uložena v CMOS čipu), ale také pro výrobu analogových obvodů (zesilovače, snímače obrazu apod.).

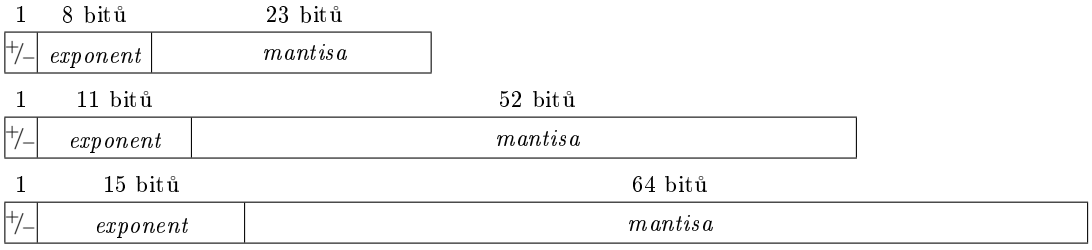
Bavili jsme se o 22nm tranzistorech, 14nm, 10nm, 7nm tranzistorech. To „pěkné malé číslo“ už dávno neodpovídá velikosti tranzistoru, obvykle jde o rozměr nějaké hodně malé části tranzistoru, třeba šířku brány. Například u 10nm procesorů Intelu (Ice Lake) je vzdálenost mezi bránou a kovovými vodiči, na které jsou napojeny elektrody drain a source, 36 nm (tj. tato vzdálenost od brány jak na jednu, tak i druhou stranu), v případě 7nm výrobního procesu u TSMC (procesory AMD) to je 40 nm. Šířka ploutve u 10nm Intelu je 34 nm, u 7nm výrobního procesu TSMC jde o 30 nm, u 7nm procesu Samsungu 27 nm. Marketing ví, na co (ne)upozornit.

5.1.3 Matematický koprocesor

Matematický koprocesor (FPU, floating point unit) je pomocný procesor určený pro vykonávání operací s čísly s plovoucí desetinnou čárkou. Může to být samostatná jednotka nebo součást CPU (dnes běžné) a nebo emulován softwarově jako mikroprogram.

Matematický koprocesor pracuje s těmito čísly:

- *normalizované číslo* – ve tvaru $\pm 1.\text{mant} \cdot 2^{\text{exp}}$, ukládáme číslo mant (ne jedničku před des. tečkou); pro všechna čísla kromě nuly a maximální hodnoty
- *denormalizované číslo* – exponent je nula, mantisa jakákoliv kromě nuly, ukládáme i číslici před des. tečkou; pro velmi malá čísla
- *nula* – exponent i mantisa jsou nulové
- *nekonečno* – nejvyšší možný exponent a nulová mantisa; vznikne jako výsledek dělení nulou
- *nečíselná hodnota* – nejvyšší možný exponent a nenulová mantisa; vznikne například při dělení nuly nulou



Obrázek 5.3: Číslo v plovoucí řádové čárce o délce 32, 64 a 80 bitů

5.1.4 Mooreův zákon

Procesory (stejně jako mnohé jiné elektronické komponenty) se skládají především z tranzistorů (polovodičových součástek) v integrovaných obvodech. Už desítky let se výrobci snaží miniaturizovat – chtějí, aby byly procesory menší, nebo sice stejně velké, ale výkonnější (tedy se do stejného prostoru má „vejít“ více tranzistorů). Toho se dá dosáhnout tak, že se zmenšují právě tranzistory.

typ čísla	exponent	mantisa v paměti	význam mantisy
normalizované	0...255	m je libovolná	desetinná část skutečné mantisy (tvar 1,m)
denormalizované	0	libovolná nenulová	skutečná mantisa s pevnou desetinnou čárkou za nejvyšším bitem
nula	0	0	0
nekonečno	255	0	–
nečíselná hodnota	255	libolná nenulová	–

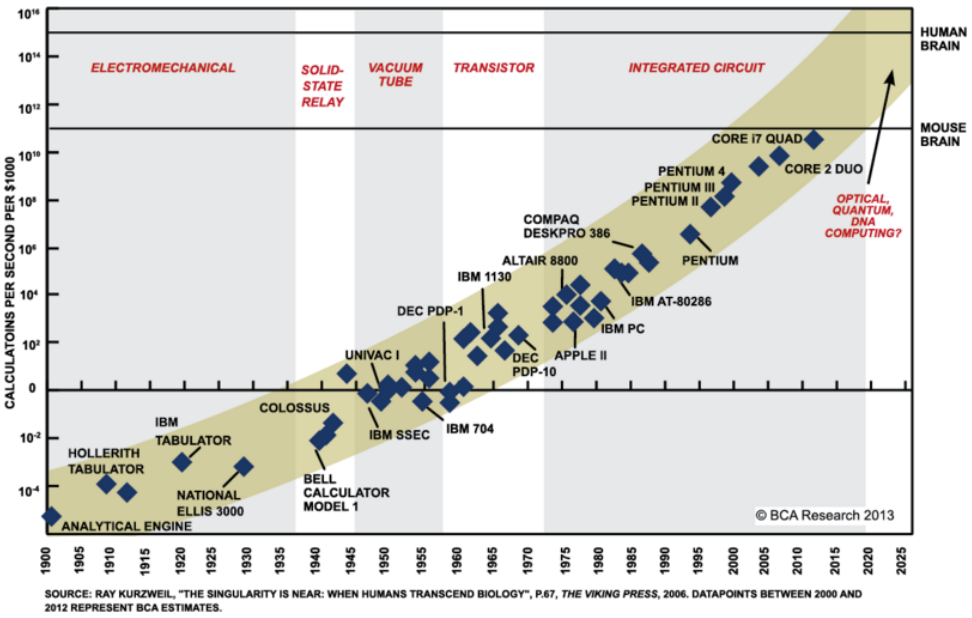
Tabulka 5.1: Typy čísel u matematického koprocesoru

Tvrzení (Mooreův zákon (Moore's Law))

Každé dva roky se počet tranzistorů v procesorech zdvojnásobí.

Gordon E. Moore (spoluzakladatel společnosti Intel), 1965, revize 1975

V původním znění před revizí se hovořilo o zdvojnásobení počtu tranzistorů každých 18 měsíců (tedy revize znamenala zpomalení). Grafické znázornění Mooreova zákona vidíme na obrázku 5.4.



Obrázek 5.4: Moorův zákon[3]

V praxi vidíme uplatnění Mooreova zákona v tom, že se neustále vylepšuje výrobní proces, co se týče velikosti tranzistorů (nyní v nanometrech).

Poznámka

Moorův zákon neříká nic o výpočetním výkonu – není řečeno, že když se zdvojnásobí počet tranzistorů, tak se zdvojnásobí výkon. Ve skutečnosti se výkon zvyšuje rychleji:

- Intel i80286 (rok 1982, 134 tis. tranzistorů, frekvence 6–12 MHz) měl výkon cca 2 MIPS (miliónů instrukcí za sekundu) při vyšší frekvenci,

[3] Zdroj: https://managementmania.com/cs/mooruv-zakon

- Intel Core i7-2600K (rok 2011, 995 mil. tranzistorů, frekvence 3,4 GHz na jádro) má výkon 128 300 MIPS

(kdyby se výkon zdvojnásobil co dva roky, byly by v r. 2011 procesory s výkonem $2^{15} = 32\,768$ MIPS)

Výrobci mají se zmenšováním tranzistorů čím dál větší problém. Tranzistory jsou totiž už natolik malé, že při další miniaturizaci narážíme na limity dané materiálem – pokud má elektron před sebou hranici o šířce jen pár atomů, pak v takové hranici určitě bude hodně děr, kterými tento elektron projde jinam než by měl. Mimochodem, šířka atomu křemíku je těsně pod půl nanometru.

5.2 Struktura procesoru

5.2.1 Logická struktura

Procesor se skládá z následujících součástí:

- *řadič* (řídicí jednotka) – řízení činnosti procesoru,
- *dekodér instrukcí* – převádí číselný kód na instrukci, které rozumí procesor, načítá operandy instrukcí,
- *ALU* (aritmeticko-logická jednotka pracující s celými čísly) – jedna nebo více,
- *FPU* (matematický koprocesor, jednotka pro operace v plovoucí řádové čárce) – jedna nebo více,
- *vnitřní sběrnice* – součásti procesoru mezi sebou komunikují,
- *registry* – malé rychlé paměti pro operandy instrukcí a mezivýsledky,
- *MMU* (Memory Management Unit) – provádí překlad virtuální adresy na fyzickou adresu v paměti,
- další součásti, také může být integrován paměťový řadič, grafické jádro, atd.

Dnes jsou většinou procesory vícejádrové. *Jádro* procesoru je relativně samostatná součást procesoru, prováděné instrukce jsou přidělovány jednotlivým jádrům. Struktura jádra může být u různých procesorů odlišná (především pokud jsou od různých výrobců), obvykle v jádře najdeme vlastní ALU, FPU (nebo je společný FPU, ale vícevláknový a každé jádro má své vlákno), registry, dekodér instrukcí, atd. Nicméně jádro může být také méně samostatné, například u starších procesorů od společnosti AMD najdeme ochuzená jádra, kde dvojice jader sdílí dekodér instrukcí, FPU a některé další součásti.

Řadič je sekvenční obvod, který řídí (skoro) všechny části procesoru a tedy zprostředkovaně počítače, a to pomocí řídicích signálů (vysílá) a stavových hlášení (přijímá), jak jsme viděli na nákresu von Neumannova schématu. Koordinuje také činnost ostatních řadičů v počítači (řadič paměti, řadič disku, atd.).

Aritmeticko-logická jednotka (ALU) provádí aritmetické a logické operace s daty. Má tyto části:

- část pro aritmetické operace (ADD, SUB, MUL, DIV), tedy umí sčítat, odčítat, násobit, dělit,
- část pro logické operace (OR, AND, XOR, apod.),
- barrel-shifter (válcový posouvač) – bitové posuny vpravo a vlevo (SHR, SHL).

Příklad

Co je to bitový posun? Vezměme si binární číslo, třeba `00100101` (máme osm bitů, osm binárních číslic). Bitový posun vlevo znamená, že všechny tyto binární číslice posuneme o jedno místo směrem vlevo (tedy na významnější pozici v čísle), na uvolněnou pozici na konci čísla dosadíme nulu:

`00100101` ⇒ `01001010`

Kdyby šlo o číslo v desítkové soustavě, pak bychom posunem a přidáním nuly na konec vlastně původní číslo násobili deseti. Ale protože jsme v binární soustavě, bitový posun doleva odpovídá násobení dvěma.

Pro lepší názornost si tato binární čísla převedeme do desítkové soustavy:

37 ⇒ 74

První číslici původního (binárního) čísla jsme vlastně umazali – zde to nevadí, přišli jsme o nulu v nejvyšším řádu, žádná informace se neztratila. Ale pokud by šlo o jedničku a datový typ by nám nedovolil číslo „roztáhnout“ na víc bitů, pak se nám ztratila část informace.

Bitový posun vpravo je přesně opačný:

00100101 ⇒ 00010010

Kdyby šlo o desítkovou soustavu, pak by umazání poslední číslice znamenalo celočíselné dělení deseti. V binární soustavě to je celočíselné dělení dvěma.

Všimněte si, že jsme vlastně poslední číslici umazali. Takže sice nehrozí ztráta informace při přechodu do vyššího řádku (když nestačí datový typ), ale na druhou stranu se ztrácí informace o nejnižším řádu. Jak je výše uvedeno, jde o *celočíselné* dělení dvěma, to znamená, že zbytek po dělení „zapomeneme“.

Sběrnice je, jak víme, komunikační cesta. Komunikovat se musí i uvnitř procesoru, tedy uvnitř procesoru máme interní sběrnici, ale také procesor musí komunikovat se svým okolím (severním mostem, nebo pokud je integrován, pak s jižním mostem) – k tomu máme sběrnici FSB, HT, QPI nebo DMI (se kterými jsme se seznámili v předchozí kapitole). Dále budeme pro zjednodušení psát o FSB.

Rychlost (takt) sběrnice FSB určuje, jak rychle procesor komunikuje se svým okolím. Čím vyšší frekvence, tím rychleji procesor komunikuje (pozor, není to totéž jako rychlost samotného procesoru).

Rychlost sběrnice FSB se u novějších procesorů a desek udává nikoliv v jednotkách Hz (tedy čistě frekvence), ale v jednotkách T/s (Transfers), obvykle GT/s (giga-). Je to proto, že tato sběrnice je vícecestná (může být třeba čtyřcestná, osmicestná, apod.), s určitou šířkou (počtem přenesených bitů najednou, například 64bitová, 128bitová apod.). Vyšší číslo je samozřejmě lepší, ale těžko se srovnává tato hodnota u různých procesorů. Vedle frekvence se zohledňuje, kolikacestná je tato sběrnice a jakou má šířku (tj. jak moc dat v jednom cyklu dokáže přenést), a na druhou stranu taky počet jader procesoru (více jader znamená vyšší vytížení sběrnice).

Násobič (multiplikátor, multiplier factor) je hodnota, se násobí frekvence FSB, výsledkem je rychlost procesoru, přesněji jeho jader. Takže násobič vlastně určuje násobnost rychlosti procesoru (jader) vzhledem k rychlosti jeho sběrnice. Násobič typicky nabývá hodnot v jednotkách nebo desítkách, například 26 (ale nemusí to být nutně celé číslo, může to být např. 5.5). V současných procesorech není nastaven trvale na stejné hodnotě, ale dynamicky se pohybuje v určitém stanoveném rozsahu.

Příklad

Předpokládejme, že násobič procesoru je nastaven na hodnotu 24, frekvence sběrnice je (pro zjednodušení) 100 MHz. Procesor pak pracuje rychlostí

frekvence procesoru	=	hodnota násobiče	×	frekvence FSB

V našem případě 24×100 MHz $= 2400$ MHz $= 2{,}4$ GHz

Frekvenci procesoru lze tedy zvýšit dvěma způsoby – buď navýšením hodnoty násobiče, anebo zvýšením frekvence FSB (resp. jejích novějších variant).

Všechny potřebné údaje – tedy rychlost procesoru (každého jádra), hodnotu násobiče (anglicky multiplier), frekvenci FSB a rychlost této sběrnice můžeme zjistit například v nám známém nástroji CPU-Z.

Paměťová sběrnice je mezi paměťovým řadičem a paměťovými moduly. Protože je dnes paměťový řadič většinou součástí procesoru, vede v reálu mezi procesorem a paměťmi. Čím rychlejší je paměťová sběrnice, tím rychleji dokáže procesor například načítat z paměti údaje, které má zpracovat. Podobně jako se přetaktovává sběrnice FSB, také paměťová sběrnice se může přetaktovat.

Poznámka

Ale pozor – u sběrnic nejde jen o co nejvyšší rychlost. Předně při nadměrných rychlostech může dojít k nestabilitě systému, a také – rychlosti různých sběrnic (a procesoru) musí být v souladu, synchronizovány. Pokud například frekvence procesoru a frekvence paměťové sběrnice nejsou synchronní, může docházet k situacím, kdy jedna komponenta musí čekat na druhou, než se jejich cykly „potkají“.

5.2.2 Instrukční sada

Instrukční sada je sada instrukcí procesoru. *Instrukce* je číslo složené z těchto informací:

- identifikace instrukce (operační znak),
- typ operandů (určení jejich délky),
- další informace, například který implicitní registr se má použít,
- samotné operandy, které se mají instrukcí zpracovat.

Příklad

Například pro assemblerovou instrukci `MOV AX, 0x1234` z instrukční sady Intelu je příslušná strojová instrukce na 32bitovém systému `0xB8 0x34 0x12`, binárně `1011 1000 0011 0100 0001 0010`. V tomto čísle je zakódováno více informací – například že jde o instrukci `MOV` (přesun, zde uložení, čísla do registru), má být použit registr `AX` a do něj má být uloženo hexadecimální číslo `0x1234`.

Identifikátor instrukce je první údaj v čísle, říká se mu *operační kód* (opcode). Za ním následují další informace o instrukci. To vše zpracuje dekodér instrukcí a předá dále ke zpracování.

Instrukční sada ve skutečnosti neurčuje jen samotné instrukce, ale celý související ekosystém – datové typy, registry, pravidla pro ošetření přerušení IRQ, atd.

Strojový kód je posloupnost instrukcí v číselné formě. Strojový kód je charakteristický pro daný procesor (hardwarovou architekturu), ale pokud máme v operačním systému spustitelný soubor (třeba .EXE ve Windows), nejedná se čistě o strojový kód – je tam i spousta informací určených operačnímu systému. Operační systém zde vlastně funguje jako zprostředkovatel, který načítá údaje ze spustitelného souboru, a to, co je přímo strojovým kódem procesoru, posílá procesoru.

Ještě k tomu se během zpracovávání kódu procesu ve skutečnosti provádí i nějaký kód „navíc“, například pokud proces provede systémové volání (chce otevřít soubor, vykreslit něco na obrazovku, žádá o další kus operační paměti, atd.), operační systém spouští v kontextu procesu svůj vlastní kód zpracovávající toto systémové volání.

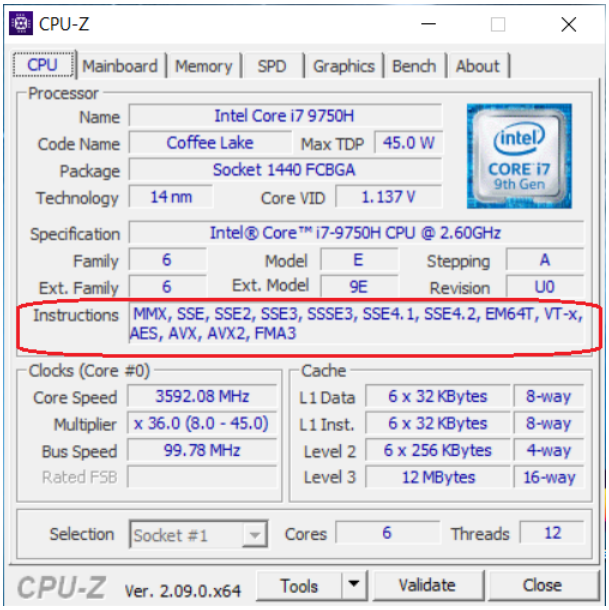
Používané instrukční sady. Každý procesor má především svou „hlavní“ instrukční sadu pokrývající běžné úkony (u dnešních procesorů 64bitová sada instrukcí), pak kvůli zpětné kompatibilitě může

podporovat starší instrukční sady, aby bylo možné spouštět programy přeložené pro starší architektury. Starší sady mohou být jen emulované, takže bývají pomalejší. Jenže to není všechno – v procesoru máme i *rozšiřující instrukční sady* obsahující například multimediální instrukce nebo instrukce pro zrychlenší šifrování. Například:

- MMX (MultiMedia Extension, 1995) – jedna z nejstarších multimediálních sad podle principu SIMD (57 instrukcí, 4 nové datové typy), od Intelu.
- 3DNow! – konkurenční multimediální technologie od AMD (1998), pracuje také s čísly v plovoucí řádové čárce, později byla zveřejněna nová varianta – Enhanced 3DNow!.
- SSE (Streaming SIMD Extensions, 1999) – původně se označovala jako MMX2, jde o vylepšení MMX, byly přidány nové instrukce a další registry.
- 3DNow! Professional – opět odpověď AMD. Obsahovala sadu 3DNow! a intelovskou SSE.
- SSE2, SSE3, SSE4. Windows 11 24H2 vyžadují podporu instrukční sady SSE 4.2. Další SSE5 (2007) – představena společností AMD.
- AVX (Advanced Vector Extensions, vznik 2008/09, nasazení 2010–2011), AVX 2.0 – druhá verze, dnes již běžná, a čím dál rozšířenější další verze AVX-512.
- AES – hardwarová podpora šifrování algoritmem AES, mohou být i další podobné instrukční sady týkající se kryptografie, např. sada SHA nebo instrukce pro podporu šifrování s Galoisovými tělesy.
- Můžeme se setkat s instrukční sadou pro zrychlení práce s neuronovými sítěmi a obecně pro umělou inteligenci, např. DL Boost (Deep Learning Boost) u Intelu.

Multimediální instrukční sady (momentálně hlavně AVX v různých verzích) slouží nejen pro zpracovávání multimediálních dat (obraz, zvuk, video), ale taky pro urychlení složitých matematických výpočtů (když to zobecníme, zpracování multimediálních dat je vlastně tvrdá matematika) nebo šifrování (například OpenSSL používá pro urychlení šifrování AVX a AVX2).

ARM procesory mají taky několik instrukčních sad. Kromě podpory šifrování taky například multimediální sadu SVE (Scalable Vector Extension).



Obrázek 5.5: Seznam podporovaných instrukčních sad v okně programu CPU-Z

Každá multimediální sada potřebuje své registry. Protože se obvykle jedná o zpracování velkých racionálních čísel (s plovoucí řádovou čárkou), jsou tyto registry poměrně velké (například AVX v první verzi používá 16 YMM registrů, každý může obsahovat buď osm 32bitových racionálních čísel, nebo čtyři 64bitová čísla).

Multimediální a šifrovací instrukční sady jsou typicky *SIMD* (Single Instruction Multiple Data, vektorové sady) – viz Flynnova taxonomie, str. 12, což znamená, že jedna (tatáž) instrukce se používá na celý vektor dat (s různými částmi obrazu provádíme totéž).

5.2.3 Cache paměť

Cache [keš] je vyrovnávací paměť. Obecně slouží jako „mezisklad“ při komunikaci mezi dvěma různě rychlými komponentami, v případě procesorové cache jde o (rychlý) procesor a (pomalou) operační paměť. Do

cache se přednačítají data a instrukce z operační paměti, o kterých se předpokládá, že je procesor bude v následujících krocích potřebovat.

Cache paměti jsou výrazně rychlejší než běžné moduly pro operační paměti, a to především z důvodu použité technologie a vnitřní struktury. Jsou bohužel také výrazně dražší, proto velikost cache má velký vliv na cenu procesoru a logicky z finančních důvodů bude cache pamětí mnohem méně než operační paměti.

Rozlišujeme L1 (level 1), L2 (level 2) a L3 (level 3) cache, pak mluvíme o cache úrovně 1, 2 nebo 3. První a někdy i druhá úroveň cache bývají rozděleny na dvě části – cache pro data (D-cache) a cache pro instrukce (I-cache), pak se také kapacita obou částí zapisuje zvlášť.

L1 cache je umístěna přímo v procesoru, ve vícejádrových procesorech má každé jádro svou vlastní L1 cache. Je nejrychlejší a má (logicky) nejmenší kapacitu ze všech tří úrovní cache (u desktopů většinou v desítkách kB). L1 cache obsahuje data a instrukce, které procesor potřebuje okamžitě. Svůj obsah načítá obvykle z L2 cache.

L2 cache je také v pouzdře procesoru, může být společná pro všechna jádra nebo každé jádro může mít vlastní, případně může být společná pro modul dvou jader. Je prostředník mezi L1 cache a operační pamětí, nebo L1 a L3 cache, pokud ji procesor má. Bývá větší než L1, obvyklá velikost je ve stovkách až tisících kB (pokud se nepoužívá L3 cache, pak v jednotkách MB), ale bývá o něco pomalejší než L1 (zvláště pokud L2 získává data přímo z operační paměti, bývá taktována na frekvenci sběrnice sloužící ke komunikaci s paměťmi).

L3 cache se dříve objevovala jen sporadicky (v některých Applech a některých serverových řešeních), dnes se s ní setkáme běžně. Bývá společná pro všechna jádra (často se chápe jako společné komunikační prostředí pro jádra), nebo společná pro skupinu jader, kapacita v jednotkách (výjimečně desítkách) MB. Obecně platí, že do L3 může zapisovat v jeden okamžik jen jeden procesor (jádro), ale číst jich může více zároveň. V procesorech s integrovanou grafikou může být napojena i na grafické jádro (pak grafické jádro může přes L3 komunikovat s běžnými výpočetními jádry).

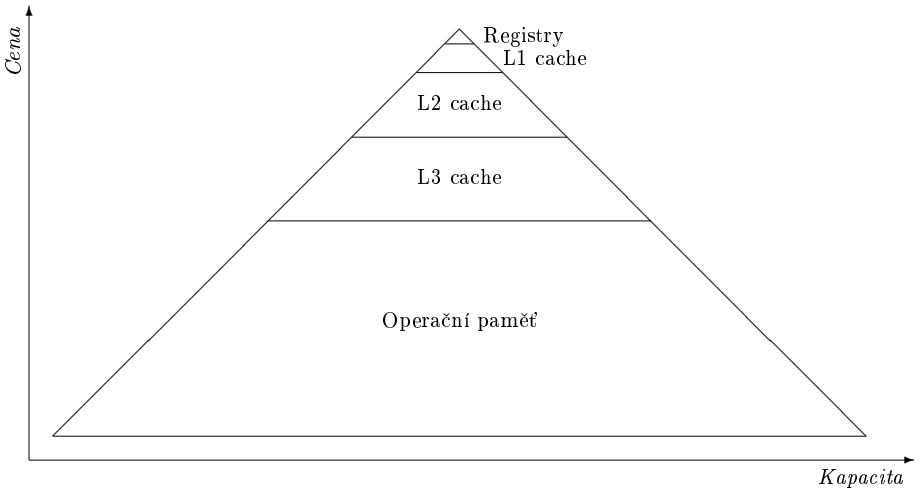
Obrázek 5.6: Přenos dat či instrukcí z paměti přes cache do výpočetních jednotek procesoru

Poznámka

Velikost cache nejvyšší použité úrovně (například Intel ji nazývá *Smart Cache*) má velký dopad na rychlost celého systému. Zvláště u úsporných procesorů bychom si na velikost L2 (nebo L3) cache měli dávat velký pozor.

Na obrázku 5.7 je znázorněna tzv. *paměťová pyramida*, kde vidíme, že čím dražší paměť, tím je jí méně (a tím blíže bývá jádrům procesoru).

V procesoru ve skutečnosti máme poněkud více paměťových obvodů než jen registry a L1/L2/L3 cache. Například jednotka MMU (Memory Management Unit), se kterou jsme se setkali ve výčtu logických součástí procesoru, používá pro urychlení překladu virtuální adresy na fyzickou adresu svou vlastní cache zvanou TLB (Translation Lookaside Buffer). Pokud už některou adresu v nejbližší minulosti přeložila, je výsledek uložen do TLB, a při opakovaném přístupu k téže stránce paměti není nutné znovu překládat.



Obrázek 5.7: Paměťová pyramida – dražší typy pamětí mají menší kapacitu

Další informace

O cache pamětech se můžete dočíst další informace například na

- https://www.tomshardware.com/reviews/cache-size-matter,1709.html
- https://www.root.cz/clanky/vyrovnavaci-pameti-cache/

L3 cache může zabírat podstatnou část čipu procesoru. Například pro architektury od Intelu vidíme nákresy procesorů na stránce https://pctuning.cz/article/narust-vykonu-cpu-za-poslednich-osm-let-amd-vs-intel?chapter=2, další kapitola téhož článku obsahuje nákresy architektur od AMD.

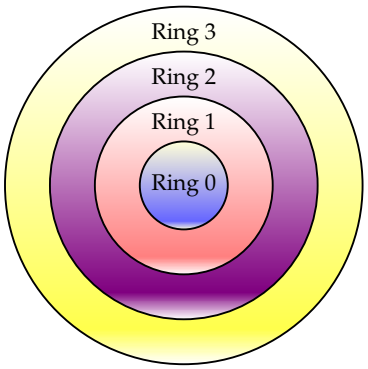
Cache je specifická tím, že urychluje přístup k něčemu jinému, a hledaný obsah v ní buď je (přednačten) nebo není. Pokud je, opravdu dochází k urychlení přístupu, pokud není (položka v ní chybí – missing, tj. stav „miss“), je nutné položku zjistit jinde (a případně do cache doplnit). Pro výpočetní jednotky je práce s cache naprosto transparentní (výpočetní jednotka netuší, že se jde přes cache). Pokud jádro procesoru například potřebuje načíst další instrukci nebo data a příslušná položka není v L1 cache, nastává *read miss* (chybí to, co chceme číst), a provede se postup vyhledávání v dalších úrovních cache, případně v operační paměti, ovšem jádro procesoru jednoduše „nějak“ dostane výsledek. Podobně při překladu virtuální adresy je nejdřív (přeložená) položka hledána v TLB, a když tam není, nastavává „TLB miss“ a překlad musí provést modul MMU.

5.2.4 Úrovně hardwarové ochrany prostředků

Současné procesory Intel a AMD vždy pracují v jedné z těchto dvou úrovní ochrany:

1. *reálný mód* – procesor se chová jako i8086 z roku 1978, používá jen 20 bitů na adresové sběrnici (adresuje maximálně 1 MB operační paměti), pro adresování se používají reálné adresy, nelze použít ochranu paměti a tedy ani multitasking, procesy mají přístup ke všemu,
2. *chráněný mód* – procesor používá celý rozsah adresové sběrnice, využívá podpůrné obvody pro tento režim včetně dalších registrů (přístupných jen v chráněném módu), implementace ochrany paměti, předpoklad multitaskingu, obvykle dovoluje přepínat mezi až 4 režimy, většinou se používají tyto dva:
   - režim jádra (kernel mode) – určený pro jádro operačního systému,
   - uživatelský režim – pro běžné procesy.

Současné operační systémy reálný mód nevyužívají, setkáme se s ním například v původním BIOSu, v některých diagnostických bootovacích programech apod.



Obrázek 5.8: Chráněný režim na procesorech Intel

Chráněný (privilegovaný) mód znamená především možnost (ne nutnost) hardwarové ochrany paměti. Od reálného se liší přidáním některých součástí (jsou přidány některé registry a příznaky) a odlišným zpracováním adres.

Na procesorech Intel a AMD existují pro chráněný mód celkem čtyři *okruhy* (ring) – Ring 0, Ring 1, Ring 2 a Ring 3. První z nich (Ring 0) je nejlépe zabezpečený, v něm obvykle běží jádro operačního systému, kdežto procesy běžící v jiných okruzích nemají do tohoto okruhu přímý přístup. Obvykle platí, že v Ring 3 běží vše kromě jádra (tj. běžné procesy).

Čím vyšší číslo okruhu, tím dál od procesoru a jiného hardwaru (co se týče možnosti přístupu). Co běží v Ring 0, to má přímý přístup k hardwaru, ale okruhy s vyšším číslem přistupují k hardwaru zásadně přes API toho, co ve „vnitřnějších“ okruzích.

Ring 1 a Ring 2 většinou nejsou používány. Přesto je lze využít pro další škálování přístupu a například s Ring 1 se setkáme u některých virtualizačních technik. Ve skutečnosti existují i „záporné“ okruhy – tento kód je přímo součástí procesoru.

Procesory ARM (používají se v mobilech, tabletech, síťových zařízeních, spotřební elektronice, atd.) to mají složitější. Základním režimem pro běžné procesy je *User mode*, ve kterém nelze přistupovat k systémovým prostředkům, je to obdoba okruhu 3 u Intelu. Kód běžící v ostatních módech již má alespoň částečný přístup k systémovým prostředkům, například (tento výčet není úplný):

- *supervisor mode* – používá se při zpracovávání softwarových přerušení, včetně systémových volání iniciovaných běžnými procesy,
- *system mode* – v tomto režimu běží speciální privilegované aplikace, které potřebují přístup k systémovým prostředkům (ne každá ARM architektura tento režim poskytuje),
- *IRQ mode*, *FIQ mode* – režimy pro ošetření hardwarových přerušení (IRQ),
- *hypervisor mode* – pouze u procesorů s rozšířenou hardwarovou podporou virtualizace (na zařízení může plnohodnotně běžet více operačních systémů zároveň).

Ať už se jedná o kterýkoliv procesor, vždy je v některém řídícím registru několik bitů, podle kterých procesor pozná, v jakém módu má zrovna pracovat. Pokud tyto bity určují, že je právě v uživatelském módu (resp. Ring3) a přijde instrukce, která v tomto módu nemá co dělat (protože přistupuje k systémovým zdrojům), procesor ji odmítne zpracovat a vyvolá výjimku. Typickým důsledkem je okamžité ukončení příslušného procesu.

U novějších procesorů se objevila ochrana proti spouštění kódu na paměťových stránkách s daty. U procesorů AMD jde o *NX bit* (Non-Executive), u Intelu *XD bit* (Execute Disable). Pokud má stránka nastaven tento bit na 1, je považována za datovou a při pokusu o zacházení s obsahem stránky jako s kódem (spuštění instrukcí zde uložených) je vyvolána výjimka, která většinou končí ukončením procesu, který se o to pokusil. Účelem je zabránit útokům typu přetečení paměti.

5.2.5 Registry

Registry jsou malá velmi rychlá paměťová místa nacházející se v procesoru, slouží obvykle k uložení adres, mezivýsledků výpočtu a operandů instrukcí. Velikost registrů je obvykle 16, 32, 64, 128, 256 bitů,

apod. (některá mocnina čísla 2).

Rozdělení registrů podle jejich určení:

- *datové* – pro data, například číslo, které má být násobeno, u Intelu spíše brány jako obecné,
- *adresové* – pro adresy proměnných z operační paměti (jakési pointery) nebo indexy polí,
- *řídicí* – obsahují stavové informace, mohou například informovat o tom, že právě dokončená instrukce má jako výsledek záporné číslo (samotné číslo bude uloženo v některém datovém registru).

V literatuře se také můžeme setkat s označeními DMR (Data Memory Register) pro datové a AMR (Address Memory Register) pro adresové registry.

Sady registrů jsou obvykle implementovány jako paměťový blok v procesoru (jádru), který se nazývá *register file*, je to vpodstatě pole registrů. Pro tyto obvody se obvykle používá SRAM (statická RAM) s více vstupy/výstupy (porty). Pokud procesor podporuje *register renaming* (což dnes podporují téměř všechny procesory), pak jednomu „názvu“ registru může odpovídat několik fyzických registrů v register file, konkrétní fyzické umístění se určuje podle právě prováděné instrukce. Díky tomu je možné paralelizovat výpočet dokonce i v rámci jednoho jádra.

V jádře procesoru může být více než jeden register file, například mohou být tyto bloky zvlášť pro přídavné SIMD sady včetně sady pro operace s plovoucí řádovou čárkou.

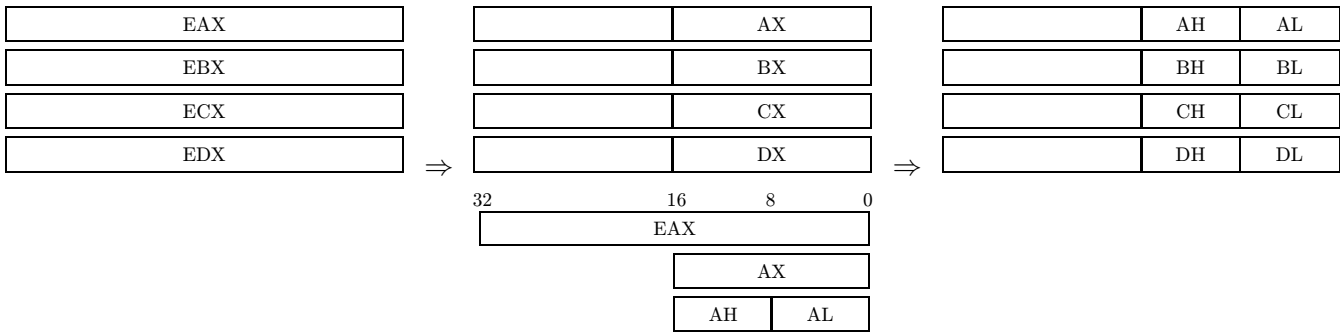
5.2.6 Registry procesorů Intel a AMD

Pro lepší představu si ukážeme, jaké registry existují na platformách Intel a AMD.

Obecné registry procesorů Intel jsou

- RAX, EAX, AX, AH, AL – používá se pro násobení a dělení, a také pro I/O operace (akumulátor),
- RBX, EBX, BX, BH, BL – používá se pro nepřímou adresaci paměti (k proměnným) (base),
- RCX, ECX, CX, CH, CL – používá se jako čítač při cyklech, posuvech a rotacích (counter),
- RDX, EDX, DX, DH, DL – používá se pro nepřímou adresaci I/O (data).

Dají se chápat jako datové registry, ale některé lze použít i pro jiné účely včetně ukládání adres, proto je nazýváme spíše obecné.



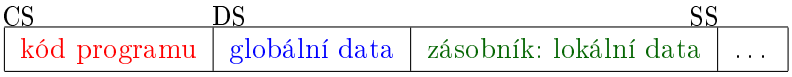
Obrázek 5.9: Vztah mezi registry Intelu o různých šířkách

Význam označení je na obrázku 5.9 (kromě nejširších forem začínajících písmenem R). Nejstarší intelovské procesory měly pouze registry s dvoupísmenným označením. Poslední písmeno označení je buď L (Lower, nižší byte), H (Higher, vyšší byte) u 8bitových částí, nebo X (zahrnuje nižší byte vpravo i vyšší byte vlevo). Předposlední písmeno (A, B, C, D) udává konkrétní typ registru.

16 bitů však brzy přestalo stačit, tak byly registry rozšířeny nejdřív na dvojnásobek (32 bitů, tříznakové označení, na začátku E), později znovu na dvojnásobek (64 bitů, na začátku R) s tím, že původní označení lze dále používat pro přístup k nižším oblastem téhož registru.

Adresové registry procesorů Intel se používají k uložení adres. Na těchto adresách mohou být například proměnné (tedy data), ale také může jít o adresy instrukcí nebo speciálních datových struktur (typicky zásobník).

Již několik desítek let se při správě paměti používají tzv. *segmenty*, třebaže na 64bitové architektuře spíše jen z důvodu zpětné kompatibility. Každý proces má k dispozici několik segmentů paměti, každý z nich k určitému konkrétnímu účelu. Obvykle je jeden segment pro kód programu (tam se načte obsah spustitelného souboru, ze kterého proces vznikl), dále jeden nebo dva segmenty pro globální data, jeden či dva segmenty pro zásobník (tam se ukládají lokální proměnné a cokoliv, co souvisí se spouštěnými funkcemi).



Obrázek 5.10: Segmentové registry procesu

Výhodou je, že když je známa adresa začátku segmentu, můžeme pro adresu konkrétního místa v daném segmentu (třeba adresu právě zpracovávané instrukce, která je rozhodně v segmentu s kódem a nikde jinde) používat kratší lokální (relativní) adresu (tzv. offset) – tyto adresy začínají číslem 0 na začátku segmentu a pro jejich uložení stačí méně bitů, než kdybychom je ukládali jako absolutní adresy (začínající na samotném začátku celého paměťového prostoru).

Ke každému segmentu je třeba si zapamatovat adresu jeho začátku, k tomu slouží *segmentové registry*. 16bitové segmentové registry:

- CS – segment s kódem programu (Code Segment),
- DS – první segment pro data (Data Segment),
- ES – pomocný segment pro data (Extended Data Segment),
- SS – segment pro zásobník programu,
- FS, GS – novější segmentové registry, jejich využití určuje operační systém.

Důležitý adresový registr je *čítač instrukcí* (Instruction Pointer) – označuje se IP, EIP, RIP (podle velikosti registru). V tomto registru je adresa instrukce, která má být v příštím kroku zpracována, obsahuje tedy relativní adresu uvnitř segmentu CS (tedy segmentu s kódem programu).

Další adresové registry (v pořadí 16bitové, 32bitové a 64bitové) jsou následující:

- BP, EBP, RBP – bázový registr (lokální proměnné),
- SP, ESP, RSP – ukazatel vrcholu zásobníku (Stack Pointer),
- SI, ESI, RSI – index v poli použitém jako zdroj dat (Source Index),
- DI, EDI, RDI – index v poli použitém jako cíl dat (Destination Index).

Postup

Podíváme se na několik příkladů pro obsah a používání registrů. Tyto příklady jsou vesměs použitelné při programování v Assembleru.

- CS:IP – plná adresa instrukce v kódu, která je momentálně vykonávána (zadáme adresu segmentu kódu a pak čítač instrukcí, který je lokální adresou v segmentu kódu),

- SS:SP – plná adresa vrcholu zásobníku (adresa zásobníkového segmentu a v něm lokální adresa vrcholu zásobníku),
- ES:[$15A0] – adresa v segmentu ES,
- DS:[BX][$20D8] – adresa v segmentu DS, relativní je obsah registru BX plus zadané číslo,
- DS:SI – tato absolutní adresa se použije, když využíváme registr SI v řetězcových instrukcích, pracujeme v datovém segmentu DS,
- ES:DI – tato absolutní adresa se použije, když využíváme registr DI v řetězcových instrukcích, pracujeme v datovém segmentu ES,
- MOV AX, [BX][SI] – chceme pracovat s prvky pole, adresu začátku pole jsme předem načetli do BX (třeba pomocí MOV), do registru AX načteme daný prvek pole (jehož index jsme předem uložili do registru SI).

Registr eFlags je speciální typ registru, ve kterém na rozdíl od ostatních máme (nepřímý) přístup k jeho jednotlivým bitům (tzv. *příznakům*). Každý významový bit má své vlastní označení. Příznaky nám většinou indikují stav ukončení právě zpracované instrukce. Například po provedení aritmetické instrukce (sčítání, násobení apod.) můžeme zjistit, zda její výsledek je či není nula, jestli je výsledek kladné nebo záporné číslo, apod.

Příznaky se používají především u porovnávání hodnot a následného větvení kódu. Procesorové instrukce pro porovnávání obvykle pracují tak, že dvě porovnávaná čísla (nebo indexy znaků v ASCII tabulce) od sebe odečtou a podle toho, zda je výsledek odčítání nulový, kladný nebo záporný, je možné poznat, která hodnota byla větší nebo jestli byly stejné.

Některé příznaky z registru eFlags nám pomáhají optimálně (bez dalšího složitého testování) určit typ hodnoty, která je výsledkem předchozí proběhlé instrukce. Například:

- ZF (Zero Flag) – nastaven na 1, pokud je výsledek operace nula,
- SF (Sign Flag) – nastaven na 1, pokud je výsledek záporný,
- OF (Overflow Flag) – nastaven na 1, jestliže došlo k přetečení výsledku (výsledek se nevejde do vyhrazeného počtu bitů).

V registru eFlags a dalších příznakových registrech k dispozici i další příznaky, například:

- CF (Carry Flag) – nastaven na 1, pokud při právě dokončené operaci došlo k přenosu 1 do vyššího řádu,
- IF (Interrupt Flag) – procesor smí vykonávat přerušení,
- DF (Direction Flag) – řídí směr zpracování při řetězcových operacích ,
- PF (Parity Flag) – nastaven na 1, pokud dolních 8 bitů výsledku obsahuje sudý počet jedniček,
- AF (Auxiliary CF) – nastaven na 1 při přenosu 1 ze spodní poloviny nižší slabiky (1 B) do vyšší,
- IOPL (IO Priority Level) – dvoubitové pole, udává prioritu potřebnou k provádění I/O. Pokud je priorita programu vyšší (tj. s nižším číslem) než obsah tohoto pole, operace I/O se neprovede a vyvolá se výjimka GPF (General Protection Fault).

Je taky rozdíl, jestli procesor (ne)pracuje v režimu se zapnutou ochranou paměti. Intelovské procesory mají ještě další registry příznaků:

- řídicí 32bitový registr příznaků CR0 – důležité příznaky:
  - PG – nastaven na 1, když je zapnuto stránkování,

  - PE – nastaven na 1, když je zapnut chráněný mód,
- registry CR2 a CR3 pro správu stránkování,
- ladicí registry DR0, DR1, DR2, DR3, DR6, DR7 jsou používány při ladění (debugging) programů,
- registry tabulek deskriptorů GDTR (adresa globální tabulky deskriptorů), LDTR (lokální), IDTR (deskriptory přerušení), o deskriptorech se budeme učit v předmětu Operační systémy,
- registr úlohy TR,
- registry pro testování TR3, TR4, TR5, TR6, TR7.

Poznámka

Pokud máte dojem, že Assembler se už dávno nepoužívá, jste na omylu. Assembler se dodnes využívá například při optimalizaci kódu, který má běžet hodně „svižně“, mnoho knihoven je takto optimalizováno. Obvykle se v Assembleru nepíše celý kód, ale jen jeho „kritické“ úseky, vkládá se do kódu vyššího programovacího jazyka.

Postup

Následuje krátký kód Assembleru vložený do pascalovského souboru:

```
...
asm
  mov ax,[x1]     { do registru AX načteme hodnotu proměnné x1 }
  mov bx,[x2]     { do registru BX načteme hodnotu proměnné x2 }
  add bx,255      { k tomu, co je v registru BX, přičteme číslo 255 }
  cmp ax,bx       { porovnáme obsah registrů AX a BX, výsledek je v eFlags }
  jnz @neni_nula  { pracujeme s příznakem ZF, Jump if not zero }
  jmp @je_nula    { když se nerovnají, odskočíme (jump) na návěští @je_nula }
  ...
@neni_nula:       { na toto návější jsme odskočili, pokud ax není rovno BX }
  sub bx,ax       { odečteme BX = BX - AX }
  mov ax,[prom]   { do registru AX načteme hodnotu proměnné prom }
  div bx          { dělíme AX = AX / BX }
  mul ax          { násobíme AX = AX * AX }
  jo @preteceni   { jestliže došlo k přetečení dostupného paměťového místa při }
  ...             {   předchozí operaci, odskočíme na @preteceni (jump if odd, }
end;              {   příznak OF) }
```

Další informace

Informace o registrech Intelu: https://software.intel.com/en-us/articles/introduction-to-x64-assembly

O použití assembleru vnořeného v kódu jazyka C:

- https://mojefedora.cz/pouziti-assembleru-v-linuxu-assembler-a-jazyk-c/
- https://www.root.cz/clanky/kdyz-cecko-nestaci-gas/
- http://phoenix.inf.upol.cz/~outrata/courses/os1/texts/asm1.html

5.2.7 Registry procesorů ARM

Procesory ARM používají jiný typ intrukčních sad než například Intel, od toho se odvíjí odlišné značení a používání registrů. V případě 32bitových procesorů je základní označení obecných registrů prostě R0,

R1, R2,..., R15, tedy máme 16 obecných registrů.

Kromě těchto označení existují i jakési substituční názvy, například:

- A1–A4 jsou registry používané pro argumenty instrukcí (hovorově „scratch registers“, něco jako pracovní registry), je to jiný název pro registry R0–R3,
- SP (Stack Pointer, totéž jako R13) ukazuje na aktuálně používaný zásobník procesu,
- LR (Link Register, totéž jako R14) se používá v případě, že v assembleru voláme subrutinu (něco jako funkci) a sem se uloží adresa instrukce následující po volání subrutiny (tj. kam se máme vrátit, když je subrutina ukončena),
- PC (Program Counter, totéž jako R15) ukazuje na instrukci, která je „v plánu“, tedy má být zpracovávána v příštím kroku.

Dále se používá například registr příznaků CPSR (Current Program Status Register), který je obdobou registru eFlags u Intelu a AMD, tedy z jednotlivých bitů se dozvíme například to, jestli výsledek předchozí instrukce je nula, záporné číslo, zda došlo k přenosu do vyššího řádu, apod.

Příklad

Jak bylo výše uvedeno, některé registry se fyzicky překrývají. Například pokud v kódu ukončujeme subrutinu (funkci) a vracíme se do původního kódu, je bod návratu uložen v registru LR (resp. R14) a je třeba ho dostat do registru PC (resp. R15). Jsou dvě možnosti, jak to provést:

```
MOV PC,LR
```

nebo

```
MOV R15,R14
```

Ve skutečnosti 32bitové procesory ARM mají 48 „fyzických“ registrů (některé se částečně překrývají, takže fyzicky o něco méně), přičemž konkrétní použitý registr závisí na tom, v jakém režimu procesor zrovna běží (uživatelském, systémovém, ošetření IRQ,...). Například výše zmíněný registr SP má variantu SP_usr pro uživatelský režim, SP_svc pro režim zpracování softwarového přerušení, SP_irq pro režim ošetření IRQ, atd.

Registry 64bitových procesorů ARM jsou pojmenovány trochu jinak: obecné jsou X0–X30 (takže 31 obecných registrů). Všechny jsou 64bitové, a pokud je třeba pracovat pouze s jejich spodními polovinami (na těchto procesorech existují 64bitové i 32bitové instrukce, 32bitové používají právě jen spodní poloviny obecných registrů), pak se používají názvy W0–W30. Registry se stejným číslem odkazují na totéž místo v paměti, jen písmena „X“ a „W“ určují jiný počet bitů, které se použijí.

Překrývání názvů už není tak rozsáhlé jako u 32bitové architektury. Registr SP se nekryje s žádným obecným registrem a má opět několik fyzických variant pro různé režimy procesoru. Osamostatnily se i další specifické registry, včetně PC. Ovšem registr LR se kryje s X30.

U ARM se můžeme setkat s názvem *register bank*. Pokud má mít určitý registr potenciálně jiný obsah v různých módech procesoru, pak o něm hovoříme jako o „banked“ registru. Každý mód má svůj register bank, v něm jsou fyzická umístění právě těchto registrů. Například u 32bitové ARM architektury jsou registry R0–R7 stejné (a přístupné) ve všech módech, tj. nikdy nejsou bankovány. Registry R8–R12 jsou stejné ve skoro všech módech, kromě módu FIQ, tj. tento mód má ve svém banku vlastní instance těchto registrů.

Registry R13 (SP) a R14 (LR) jsou bankovány ve všech privilegovaných módech (tj. všech kromě uživatelského, jinými slovy – každý mód má svou instanci, uživatelský má „originál“). Registry R15 (PC) a CPSR (příznaky) nejsou bankovány, ale v uživatelském módu je do nich omezený přístup.

V tabulce 5.2 je přehled adresních módů (tedy způsobů přístupu instrukcí k datům) pro procesory ARM, ke každému příklad instrukce, význam a druhotný efekt. Například první uvedený mód znamená,

že pracujeme přímo s číselnou konstantou, druhý s proměnnou (v paměti), třetí s obsahem registru, čtvrtý s registrem, ve kterém je adresa proměnné z paměti (tedy registr slouží jako pointer), atd.

#n	Mód	Příklad	Význam	Druhotný efekt
1	Immediate	`MOV R1, #25`	`R1=25`	žádný
2	Direct	`LOAD R1, Variable`	`R1=Variable`	žádný
3	Register direct	`ADD R1, R2, R3`	`R1=R2+R3`	žádný
4	Indirect	`LDR R1, [Variable]` or `LDR R1, (Variable)`	`R1=*Variable`	žádný
5	Register indirect	`LDR R1, [R2]` or `LDR R1, (R2)`	`R1=*R2`	žádný
6	Index addressing	`LDR R1, 24[R2]` or `LDR R1, 24(R2)`	`R1=*(R2+24)`	žádný
7	Base register	`LDR R1, [R2, R3]` or `LDR R1, (R2, R3)` `LDR R1, [R2, #12]` or `LDR R1, (R2, #12)`	`R1=*(R2+R3)` `R1=*(R2+12)`	žádný
8	PC-relative	`LDR R1, [PC, #-8]` or `LDR R1, (PC, #-8)`	`R1=*(PC-8)`	žádný
	Pre-index mode with base updated	`LDR R1, [R2, #8]!` or `LDR R1, (R2, #8)!`	`R1=*(R2+8)`	`R2=R2+8`
	Post-index mode with base updated	`LDR R1, R2, #8`	`R1=*(R2)`	`R2=R2+8`

Tabulka 5.2: Adresní módy ARM procesorů

Další informace

- https://wiki.cdot.senecacollege.ca/wiki/Aarch64_Register_and_Instruction_Quick_Start
- http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.100076_0100_00_en/gln1530702607451.html
- http://simplemachines.it/doc/arm_inst.pdf
- ftp://www.cs.uregina.ca/pub/class/301/ARM-addressing/lecture.html

5.3 Procesory podle instrukčních sad

Délka zpracování programu přibližně odpovídá tomuto vzorci:

$$\text{délka zpracování programu} \approx \text{délka cyklu procesoru} \times \text{počet cyklů na instrukci} \times \text{počet instrukcí programu}$$

Následující dělení má vliv především na parametry *počet cyklů na instrukci* a *počet instrukcí programu*.

Poznámka

Dále sice budeme hovořit o CISC a RISC procesorech, ale přesnější by bylo hovořit o CISC a RISC instrukčních sadách, protože, jak víme, procesory běžně mají více různých instrukčních sad. Tato modulární strategie je výhodná – umožňuje do dalších generací procesorů adaptivně implementovat další sady instrukcí podle potřeby, přičemž procesor může zůstat zpětně kompatibilní (půjdou na něm spustit starší programy využívající starší instrukční sady).

Stručně si popíšeme architektury typu CISC a RISC, podrobnější srovnání najdete v tabulce 5.3 na straně 89.

CISC (Complete Instruction Set Computing) sady se vyznačují velkým počtem instrukcí, pro mnoho nejběžnějších úloh je stanovena vlastní instrukce. Instrukce jsou reprezentovány pomocí *mikroprogramů* uložených ve speciální paměti v procesoru (variaci typu ROM paměti, dnes flash ROM, drahou). Každá instrukce zabírá určitý počet bitů, u CISC sad mívají různé instrukce různý počet bitů (nejsou stejně dlouhé) – kdybychom trvali na statické délce reprezentace instrukcí, bylo by potřeba vyhradit pro uložení jedné instrukce zbytečně mnoho bitů.

Výhodou je, program naprogramovaný v CISC sadě má celkově menší počet instrukcí, než kdyby byl naprogramován v RISC sadě, ale na druhou stranu samotná instrukce zabere více místa v paměti a její vyhodnocení trvá typicky více cyklů.

RISC (Reduced Instruction Set Computing) sady obsahují jen základní instrukce (obecně několik desítek), které jsou velmi malé (důležitější než malý počet je jednoduchost struktury instrukcí) a téměř všechny stejně dlouhé (výjimkou obvykle bývají instrukce přístupu do paměti). Pro instrukce nepotřebujeme žádnou speciální paměť, jsou reprezentovány pomocí obvodů. Důsledkem je, že zpracování jedné instrukce trvá obvykle jeden cyklus procesoru (to se vztahuje k frekvenci, na které procesor pracuje).

Program napsaný RISC instrukcemi obsahuje mnohem víc instrukcí, ale protože každou z těchto instrukcí zvládne procesor na jeden takt, netrvá provedení programu déle.

CISC	**RISC**
velmi mnoho instrukcí, pro každou běžnou úlohu je vlastní instrukce	jen základní instrukce (málo), složitější úlohy se skládají z více jednoduchých instrukcí
instrukce jsou dlouhé (navíc různě dlouhé) a mají víc operandů, vyhodnocení – u každé různý počet cyklů, podle složitosti	instrukce jsou krátké (obvykle všechny stejně dlouhé) a jednoduché, vyhodnocení – (téměř) každá jeden cyklus procesoru
protože u každé instrukce může trvat zpracování různě dlouho, nevíme předem, jak dlouho bude trvat zpracování určité posloupnosti instrukcí	(téměř) všechny instrukce se zpracovávají v jednom taktu procesoru, takže je předem známo, jak dlouho bude trvat zpracování sekvence instrukcí
každá instrukce je reprezentována mikroprogramem, tedy v procesoru potřebujeme místo na mikroprogramy (zabírají velkou část procesoru)	instrukce jsou reprezentovány v obvodech (hardwired), obvody s instrukcemi zabírají jen málo místa, nevytěžují prostorově procesor
načítání instrukce je pomalé, vysoká režie	načítání instrukce je rychlé, nezdržuje
dekodér instrukcí je složitější, instrukce mají různorodou strukturu s různými příznaky	dekodér je jednoduchý, jen pracuje s obvody
problémy při zvyšování frekvence – procesor při vyšších frekvencích nestíhá	snadnější zvyšování frekvence – pracuje se převážně s obvody
inovace procesoru je jednodušší, především se přeprogramovávají mikroprogramy nebo se do paměti ukládají nové	inovace procesoru je hardwarová, je nutné zasahovat do obvodů

Tabulka 5.3: Srovnání architektur typu CISC a RISC

Další typickou vlastností RISC sad je nedestruktivnost operací – zatímco u CISC instrukcí je často výsledek operace načten do jednoho z operandů, u RISC bývá pro výsledek použit další operand.

Rozdíl mezi CISC a RISC sadou si můžeme představit třeba na analogii ze stavebnictví: RISC znamená,

že máme několik základních stavebních prvků (cihly, tašky na střechu apod.), kdežto CISC znamená, že místo jednotlivých cihel máme předpřipravené stavební bloky pro postavení rovné stěny, stěny s výklenkem, stěny s komínem, s krbem, rohový blok, blok pro balkon, atd.

Příklad

Pro CISC sady je například typická takováto operace sčítání (obsah registru `EAX` a obsah proměnné `prom` jsou sečteny, výsledek se uloží do registru `EAX`):

```
ADD EAX, prom
```

Tedy původní obsah registru `EAX` byl přepsán, „zničen“. Oproti tomu pro RISC sady je typické následující:

```
ADD prom1, prom2, vysl
```

Instrukce sečte obsah proměnných `ADD prom1` a `prom2`, výsledek uloží do proměnné `vysl`. Původní obsah prvních dvou operandů zůstává zachován.

RISC sady se objevily v procesorech řady PowerPC, dále u některých serverových platforem, dnes jsou především čistě RISC procesory v embedded a malých mobilních zařízeních (procesory ARM).

„Čisté“ CISC sady se používaly ve starších procesorech od Intelu (například stará Pentia), ale v současné době se u Intelu a AMD jako základní instrukční sady používají spíše CISC s RISC rysy (takže jakási kombinace). Jednoduché instrukce (jako jsou třeba běžné aritmetické operace) jsou implementovány „RISCově“ v obvodech (aby se šetřilo místo v procesoru), složité instrukce mají formu mikroprogramů. Pokud je třeba v novějším procesoru implementovat starší instrukční sady (které jsou většinou CISCové), dělá se to pomocí emulace (tedy mikroprogramově).

K RISC procesorům řadíme zejména procesory ARM. Z dalších třeba PowerPC od IBM nebo MIPS, obojí dnes najdeme spíše v embedded systémech, procesor PowerPC má například vozítko Mars rover Curiosity. Existuje také otevřená specifikace OpenRISC.

Další informace

- Na http://www.fit.vutbr.cz/study/courses/ITP/public/itp07/architektury01.pdf je zajímavé a podrobné srovnání CISC a RISC architektur.
- https://www.root.cz/clanky/architektura-mikroprocesoru-openrisc/
- https://www.root.cz/clanky/riscova-architektura-powerpc/

5.4 Vlastnosti procesorů

U procesorů nás zajímají především tyto vlastnosti:

- pracovní frekvence (MHz, GHz)
  - vnitřní – jak rychle pracuje procesor,
  - vnější – jak rychle komunikuje se svým okolím,

  vnitřní frekvence je některým násobkem vnější frekvence,
- MIPS (Million Integer operations Per Second, celočíselných operací za sekundu) – není příliš vypovídající, protože procesory s různými typy instrukčních sad jsou takto nesrovnatelné,
- MFLOPS (Million FLoating-point Operations Per Second) – pro čísla v plovoucí řádové čárce,
- technologie výroby, počet jader, velikost cache, podporované instrukce, jak ho lze uchladit, apod.,
- patice, do které patří, se kterými čipovými sadami spolupracuje, atd.,
- ekologie – materiály, výroba, recyklace.

5.5 Techniky zvyšování propustnosti a výkonu procesorů

5.5.1 Pipelining a . . . skalární architektury

Pipelining (zřetězené zpracování, překrývání instrukcí) je rozdělení zpracování jedné instrukce mezi různé části procesoru (případně jádra), zpracovává se více instrukcí najednou (každá instrukce je v jiné fázi zpracování). Představme si to jako výrobní linku, na které se kompletuje složitější výrobek.

Příklad

Pipelining může to vypadat takto (dvě možnosti, první je dvoumístná fronta, druhá pětimístná):

1. načtení a dekódování instrukce
2. provedení instrukce a uložení výsledku

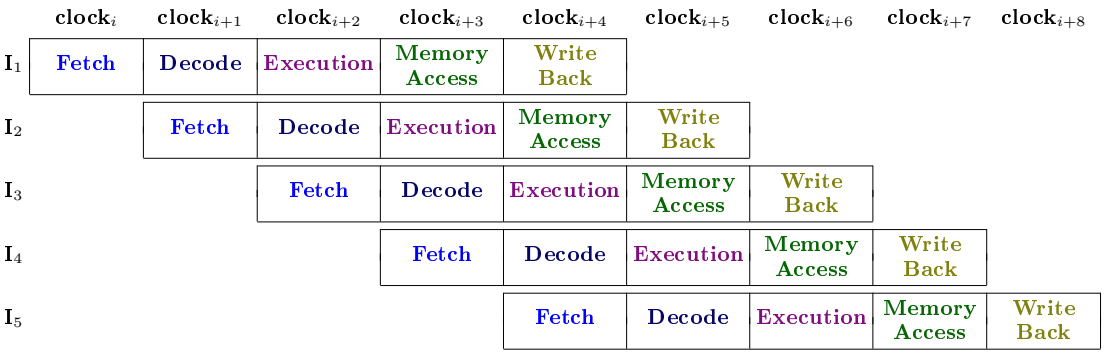
1. načtení instrukce
2. dekódování a načtení obsahu registrů
3. vyhodnocení instrukce
4. přístup do paměti
5. uložení výsledku do registrů

V současných procesorech se cyklus zpracování instrukce může se například dělit do kroků (stage)

- *fetch* – získání instrukce z paměti (resp. L1i cache),
- *decode* – dekódování instrukce, určení, co vše je třeba udělat, aby byla instrukce zpracována,
- *indirect* – získání dat potřebných pro provedení instrukce, včetně přístupu do paměti, může být zařazeno do fáze decode,
- *execute* – samotné provedení instrukce,
- *memory access* – pokud je třeba, jsou výsledky zapsány do paměti,
- *write back* – provedení změn v registrech, včetně registru s adresou následující instrukce k provedení.

Příklad

Předpokládejme jednoduchý případ, kdy všechny instrukce trvají stejně dlouho, každá je provedena během pěti cyklů procesoru, tj. provedení instrukce je rozděleno do pěti fází. Obrázek 5.11 naznačuje, jak to může vypadat (všechny „řádky“ probíhají v čase zároveň, paralelně, tedy například v čase $\mathbf{clock}_{i+4}$ máme rozpracovaných pět instrukcí, z toho první už končí, pátá začíná).



Obrázek 5.11: Příklad 5fázové pipeline

To by byla jedna pipeline, v jádru procesoru jich můžeme mít několik (pro různé typy instrukcí – celočíselné v ALU, pro čísla s plovoucí řádovou čárkou v FPU, vektorové multimediální instrukce, atd.), přičemž každá z nich může předpokládat jinou délku zpracování instrukce, trochu jiné fáze, apod.

Například Intel používá pipelines s délkou 14 až 20, obvykle dvě pipelines v jádře plus FPU, podle konkrétní mikroarchitektury. ARM procesory mají řadu pipelines s různou délkou (1 nebo 2 fáze pro celočíselné operace, mnohem více pro další typy operací).

Používají se tyto pojmy:

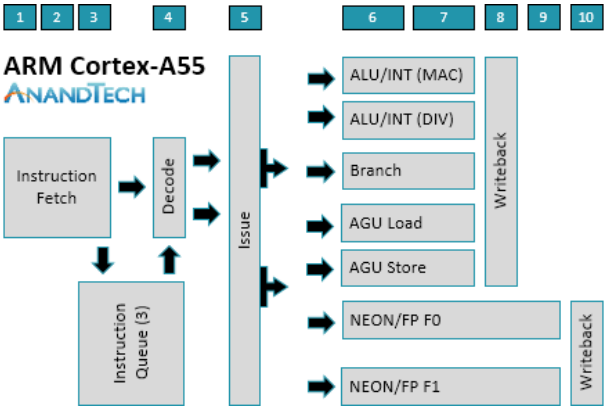
- *Skalární architektura* (pipelining) – jednu instrukci zpracovávají postupně různé části procesoru, které jsou navzájem zřetězeny (fronta),
- *Superskalární architektura* – takových zřetězení existuje více než jedno (více front),
- *Hyperskalární architektura* – prodlužování front (i několik desítek kroků).

Prakticky všechny současné procesory jsou superskalární, mnohé zároveň hyperskalární.

Příklad

Na obrázku 5.12 vidíme pipelines v procesoru ARM Cortex A55 (Cortex A je mainstreamová řada procesorů ARM). Uvnitř každého jádra máme několik pipelines pro různé druhy instrukcí.

Fáze Fetch (3 tiky), Decode (1 tik) a Issue (1 tik) jsou ve všech pipelines, stejně tak i fáze Writeback (1 tik). Fáze Issue slouží k načítání dat z registrů (tj. z paměťové struktury register file).



Obrázek 5.12: Pipelines v procesoru ARM Cortex A55[4]

Dvě pipelines jsou určeny pro celočíselné operace (obě umí jednoduché instrukce typu sčítání, bitové posuny apod.), ale jen první z nich zvládá násobení (MAC znamená Multiply Accumulate), a jen druhá pro změnu zvládá dělení (DIV). Poslední dvě pipelines (NEON/FP) jsou určeny pro operace s čísly v plovoucí řádové čárce.

Pipelines AGU (Address Generation Units) slouží ke zpracování operací `LOAD` (načítání z paměti do registrů) a `STORE` (ukládání z registrů do paměti), opět zde máme specializaci.

Další informace

- https://www.anandtech.com/show/11441/dynamiq-and-arms-new-cpus-cortex-a75-a55 (pět kapitol)
- https://www.arm.com/products/silicon-ip-cpu (přehled AMD procesorů)
- https://developer.arm.com/ip-products/processors/cortex-a (procesory Cortex-A)

[4] Zdroj: https://www.anandtech.com/show/11441/dynamiq-and-arms-new-cpus-cortex-a75-a55/4

5.5.2 HyperThreading, více jader, více procesorů

Více procesorů: Multiprocessing znamená více procesorů v jednom systému. Více procesorů na jedné základní desce je obvyklých spíše u serverových platforem (musí být víc patic na základní desce), na desktopech a v mobilních zařízeních se setkáváme s jedním vícejádrovým procesorem. Více procesorů (a vlastně také jader) by mělo být podporováno operačním systémem, aby vůbec mohly být využity:

- *jednoprocesorové* (monoprocesorové) – Windows s DOS jádrem (verze 9x, ME),
- *víceprocesorové* (multiprocesorové) – UNIXové systémy včetně Linuxu, Windows s NT jádrem (NT, 2000, XP, Vista, 7, 8, 10), dokážou rozplánovat alespoň některé úlohy tak, aby mohly být zpracovávány na více procesorech zároveň.

Rozlišujeme dva typy multiprocessingu podle toho, zda jednotlivé procesory jsou/nejsou využívány univerzálně kterýmkoliv procesem:

- asymetrický (ASMP) – jeden procesor je vyhrazen pro procesy systému a uživatelské procesy běží na ostatních procesorech,
- symetrický (SMP) – kterýkoliv proces může běžet na kterémkoliv procesoru.

Všechny současné operační systémy podporují SMP.

Více jader: Jádro procesoru je tvořeno především ALU, FPU, řadičem, registry dekodérem instrukcí a dalšími důležitými podpůrnými součástmi. Vícejádrový procesor je procesor, který obsahuje více (fyzických) jader integrovaných v jediném čipu a dále logiku určenou k jejich propojení.

Využívá se toho, že několik jednoduchých jader na jedné křemíkové destičce má vyšší výpočetní výkon s nižšími náklady než jedno složité a supervýkonné na téže ploše, nebo dokonce několik fyzických procesorů (ty by zabraly větší plochu).

Současné procesory bývají běžně vícejádrové. Obvykle jsou počty jader mocninou čísla 2 (2, 4, 8 apod.), ale můžeme se setkat i s lichým počtem, například 3. Typicky je některé jádro vadné, deaktivované, ale ostatní jsou v pořádku, tak proč takový procesor vyhazovat.

Poznámka

Pozor, není jádro jako jádro. Jsou velké odlišnosti mezi jádry procesorů různých platforem (např. ARM vs. Intel), ale také procesorů s podobnými instrukčními sadami (Intel vs. AMD) a dokonce mezi různými generacemi téhož typu procesoru.

U procesorů AMD se v architektuře Bulldozer a jejích následovnících objevila „odlehčená“ jádra – v procesoru jsou jádra spárována do dvojic, přičemž dvojice sdílí některé součásti, které by jinak mělo každé jádro vlastní. Důsledkem je zhoršení porovnatelnosti s procesory majícími jinou architekturu.

Hyperthreading: Jeden procesor (nebo jádro) je schopen provádět kód několika výpočetních vláken (threads, například kód několika procesů). Existuje více různých implementací hyperthreadingu.

V procesorech Intelu a AMD se používá varianta SMT (Simultaneous MultiThreading, Intel ji označuje jednoduše HT, HyperThreading, AMD přímo zkratkou SMT): jedno fyzické jádro se hlásí systému jako dvě logická jádra se sdílenými prostředky, resp. pokud v procesoru máme víc než jedno fyzické jádro, počet logických jader je dvojnásobný (takže například dvě fyzická jádra „vidí“ operační systém jako čtyři logická).

Procesor musí být superskalární (tj. více pipelines), dvě pipelines pracují paralelně. V každé pipeline může být zpracováváno výpočetní vlákno jiného procesu (případně u vícevláknového procesu jeho různá vlákna). Tato vlastnost musí být zapnuta v BIOSu (obvykle bývá). Nárůst výkonu je udáván především u vícevláknových aplikací v desítkách procent.

Je třeba si uvědomit, že pokud mají být na tomtéž jádru procesoru prováděna paralelně dvě různá výpočetní vlákna (nebo více), třeba i z různých procesů, musí být jejich činnost kompletně oddělena – například používají oddělené adresové prostory, tj. pod toutéž adresou (jako číslem) každé z vláken „vidí“ něco jiného ze svého vlastního adresového prostoru, každé vlákno má svůj obsah registrů, který nesmí být přepsán jiným vláknem, atd. Takže vždy, když operační systém určí, že se v pipeline má přejít na jiné výpočetní vlákno (například jiný proces „dostane“ procesor/jádro), proběhne tzv. *přepnutí kontextu* – obsah registrů včetně těch adresových od původního vlákna je uschován a načte se obsah registrů následujícího vlákna (to se samozřejmě děje i bez multithreadingu, ale pouze souhrnně za celé fyzické jádro; s multithreadingem zvlášť pro logická jádra).

Společnost ARM představila první procesor s podporou SMT, jde o ARM Cortex-A65AE, který se má na trhu objevit v roce 2020. V segmentu mobilních a embedded zařízení zatím tato technologie nebyla považována za přínosnou, což se může změnit.

Trochu jinak funguje *interleaved hyperthreading*, který (na rozdíl od SMT) má smysl i v případě, že v jádře máme jen jednu pipeline. Existují dvě základní formy:

- *Fine-grained multithreading* (s jemnou granularitou) – v rámci pipeline se přepíná mezi několika výpočetními vlákny, a to i v každém tiku (ale záleží na architektuře, může to být po jednotkách až desítkách tiků). Každé výpočetní vlákno, jehož instrukce jsou zpracovávány, musí mít místo pro uložení svého stavu, tj. vlastní instanci register file. Tento typ hyperthreadingu se objevil u některých serverových architektur, např. Sun Ultra-SPARC T1 (v každém jádru 4 vlákna zároveň, přepínání mezi vlákny v každém tiku), v pozdějších generacích se přešlo na SMT.
- *coarse-grained multithreading* (s hrubou granularitou) – podobně jako předchozí, ale mezi vlákny se přepíná jen při „hardwarových událostech“, například cache miss, TLB miss, synchronizační události, výpadek stránky v paměti, apod. Tento typ hyperthreadingu byl implementován například u některých serverových procesorů Intel Itanium (Montecito), ty už dávno nejsou vyráběny.

5.5.3 Řešení datových závislostí

Pipelining sice pomáhá co nejlépe využívat kapacitu procesoru, ale přináší s sebou taky problémy, protože po sobě jdoucí instrukce mohou být navzájem datově závislé: například následující instrukce potřebuje výsledek předchozí instrukce, ale když se jejich výpočet překrývá, tak následující instrukce musí počkat.

	$\mathbf{clock}_i$	$\mathbf{clock}_{i+1}$	$\mathbf{clock}_{i+2}$	$\mathbf{clock}_{i+3}$	$\mathbf{clock}_{i+4}$	$\mathbf{clock}_{i+5}$	$\mathbf{clock}_{i+6}$	$\mathbf{clock}_{i+7}$	$\mathbf{clock}_{i+8}$
$\mathbf{I}_m$	Fetch	Decode	Execution	Memory Access	Write Back				
$\mathbf{I}_{m+1}$		Fetch	Stall	Stall	Stall	Decode	Execution	Memory Access	Write Back

Obrázek 5.13: Instrukce s datovou závislostí na předchozí

Situaci vidíme na obrázku 5.13 (instrukce $\mathbf{I}_{m+1}$ je datově závislá na předchozí instrukci). Takovou situaci nazýváme *hazard* (v češtině se dá říct „riziko“). Rozlišujeme

- *data hazard* – nezbytná data opravdu nejsou ve správnou dobu přístupná,
- *structure hazard* – konflikt hardwarových zdrojů, například hardwarová komponenta neumožňuje určitou kombinaci těchto za sebou jdoucích instrucí:

```
MUL R1, R2, R3          // znamená R1 = R2*R3
MUL R4, R5, R6          // znamená R4 = R5*R6
```

Zdánlivě je vše v pořádku. Jenže násobič v této architektuře potřebuje dva kroky k vynásobení dvou čísel, mezitím nedokáže násobit další čísla, tedy druhá násobící instrukce musí jeden krok počkat,

- *control hazard* – způsoben problémem s podmíněným větvením kódu (například ve zdrojovém kódu jsme měli if, switch, smyčku while nebo něco podobného), kdy jsou alespoň dvě různé cesty v kódu, kterými lze jít, a procesor se při odhadu správné cesty „netrefí“.

Situace „data hazard“ je právě výsledkem datové závislosti. Například výše naznačená situace by mohla vzniknout posloupností těchto instrukcí (kód je pro procesor ARM):

```
ADD R2, R4, #150        // znamená R2 = R4+150
ADD R3, R2, #84         // znamená R3 = R2+84
```

První instrukce uloží výsledek do `R2`, další instrukce ho potřebuje pro svůj výpočet. Ovšem fáze Decode druhé instrukce (která načítá hodnotu z R2) to může provést až po fázi Write Back předchozí instrukce.

Částečné řešení je používání mechanismu *interstage buffer*, což jsou paměťová místa (obdoba registrů) využívaná na začátku či konci jednotlivých fází, sloužící k „předčasnému transferu“ hodnot mezi instrukcemi. Například podle obrázku 5.13 by na konci fáze Execution první instrukce došlo k předání obsahu registru R2 přes interstage buffer následující instrukci, počet prázdných kroků by se snížil ze tří na jeden.

Graf závislostí (Dependence graph, Data-flow graph) je graf znázorňující datové závislosti mezi instrukcemi, jak vidíme na následujícím příkladu. Uzly grafu jsou instrukce, hrany ukazují závislosti.

Počet hran v grafu (stejně jako zachycených instrukcí) závisí na délce a struktuře dané pipeline, množina hran se dynamicky mění podle toho, jak instrukce procházejí přes pipeline. Instrukce zůstává závislá, dokud její parametry jsou v *nekonzistentním stavu* (tj. čeká na jejich dodání od jiné instrukce), potom k ní vede nejméně jedna hrana. V okamžiku, kdy závislost končí, je příslušná hrana odstraněna, a pokud k instrukci nevedou žádné hrany (tj. všechny parametry jsou v *konzistentním stavu*), může instrukce pokročit ve své pipeline.

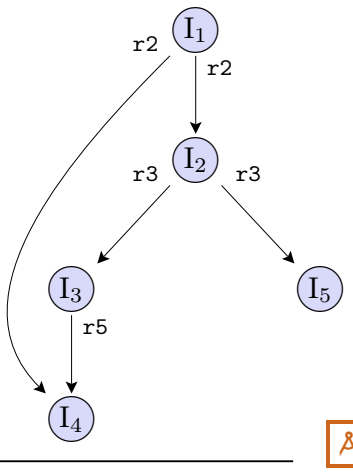
Příklad

Předpokládejme tuto posloupnost ARM instrukcí:

I_1: `LDR R2, [R4]`
I_2: `ADD R3, R2, #150`
I_3: `SUB R5, R3, #80`
I_4: `ADD R4, R5, R2`
I_5: `LDR R3, =280`

První instrukce změní hodnotu registru R2, a pak instrukce I_2 a I_4 tuto hodnotu používají. Takže dvě instrukce jsou závislé na instrukci I_1.

Podobné vztahy jsou mezi dalšími instrukcemi, všechny tyto vztahy můžeme vidět na grafu vpravo.



Datové závislosti se řeší následovně (nechceme, aby pipeline zůstávala zbytečně prázdná):

- stalling – přidáváme „prázdné“ intervaly (NOP operace, taky se jim říká „bubble“ – bublina), což ovšem není ideální řešení, provádí buď procesor nebo se použije softwarové řešení,
- používání interstage buffers – popsáno výše před příkladem,
- operand forwarding – obcházíme registry a buffery včetně interstage buffers, hodnoty se předávají mezi instrukcemi uvnitř pracující komponenty (například ALU, FPU), je to rychlejší a v rámci probíhajícího kroku (tiku),

- instructions reordering (přeuspořádání instrukcí) – často (ne vždy) je možné změnit pořadí čekajících instrukcí tak, aby navzájem závislé instrukce nenásledovaly moc rychle za sebou,
- softwarové řešení – obdoba předchozího, přeuspořádání (optimalizaci) provede už překladač.

Příklad

Jak bylo uvedeno u první možnosti, vkládání prázdných cyklů se dá řešit i softwarově během překladu do spustitelného kódu, a to pomocí „prázdné“ instrukce NOP (Not-operation).

Původní instrukce:

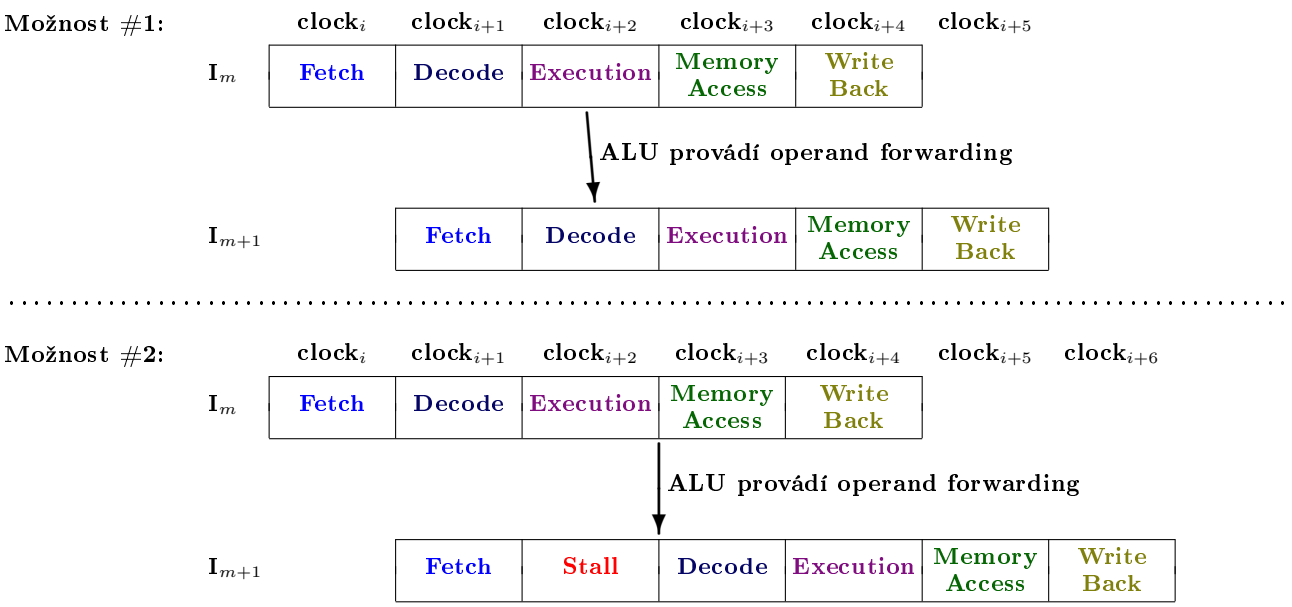
```
ADD R2, R4, #150
ADD R3, R2, #84
```

Pozměněné instrukce:

```
ADD R2, R4, #150
NOP
NOP
NOP
ADD R3, R2, #84
```

Příklad

Implementace třetí možnosti (operand forwarding, register bypassing) závisí na konkrétní architektuře. Na obrázku 5.14 jsou znázorněny dvě možnosti, z nichž první je zjevně optimálnější a je použitelná, pokud ALU (která zrovna tento výpočet provádí) dokáže v jediném kroku vypočíst výsledek první instrukce a zároveň použít při přípravě parametrů druhé instrukce.



Obrázek 5.14: Operand forwarding

Typy datových závislostí. Příklad na straně 95 ukazuje graf závislostí. Tento graf (vhodně implementovaný) může být použit pro detekci závislostí. Účelem je mít v grafu co nejméně hran a zároveň co nejméně bublin v pipeline. Postupy při přeuspořádání instrukcí se liší podle toho, o jaký typ problému „data hazard“ se jedná (následují ARM instrukce):

- *RaW* (Read after Write, True Data Dependence, Flow Dependence) – například:

  I_1: `LDR R2, [R4]`
  I_2: `ADD R3, R2, #150`

pokud tyto dvě instrukce nejsou ve správném pořadí nebo instrukce I_2 je provedena příliš rychle, je výsledek výpočtu chybný,

- *WaR* (Write after Read, Anti-dependence) – posloupnost operací je opačná než v předchozím příkladu:

  I_1: `ADD R4, R2, R3`
  I_2: `SUB R2, R5, #50`

  operace `SUB` (odečítání) musí být provedena po I_1 (nejdřív `R2` je načten instrukcí I_1, potom je přepsán instrukcí I_2), jinak dostaneme špatný výsledek,
- *WaW* (Write after Write, Output Dependence) – obě instrukce zapisují do téhož místa, například

  I_1: `ADD R3, R2, #10`
  I_2: `ADD R3, R5, R2`

  tato posloupnost může vypadat i jinak, z praktického hlediska pravděpodobněji:

  I_1: `ADD R3, R2, #10`
  I_2: `ADD R3, R3, R2`

  v druhém případě máme dokonce dva typy závislostí zároveň: WaW a RaW.

Při řešení problému „data hazard“ přeuspořádáním instrukcí by se hlavně nemělo stát, že po odstranění jednoho typu závislosti vznikne jiný typ závislosti.

5.5.4 Nepodmíněné a podmíněné skoky

Nepodmíněné skoky. V programech bývá velké množství větvení, cyklů, volání instrukcí typu exit/break/continue, skoků do jiných částí kódu (včetně spuštění funkce a návratu po ukončení funkce), atd., což nám poněkud ztěžuje situaci v pipeliningu. Nepodmíněné skoky v kódu vznikají buď použitím `break` uvnitř cyklu nebo větvení, nebo voláním funkce či naopak návratem po provedení funkce.

Příklad

Následující dva úseky kódu jsou podobné, jeden v ARM, druhý u Intelu. Nejdřív je instrukce sčítání, druhá instrukce skoku (ta nás má přenést na návěští někde dál v kódu), v posloupnosti pak máme instrukci, kterou bychom správně měli „přeskočit“.

Máme tento kód pro ARM:

```
        ADD R2, R3, #50
        B navesti
jineNa: SUB R4, R5, [R6]
        ...
navesti:
```

Máme tento kód pro Intel:

```
        ADD EAX, 50
        JMP navesti
jineNa: SUB EBX, [EDI]
        ...
navesti:
```

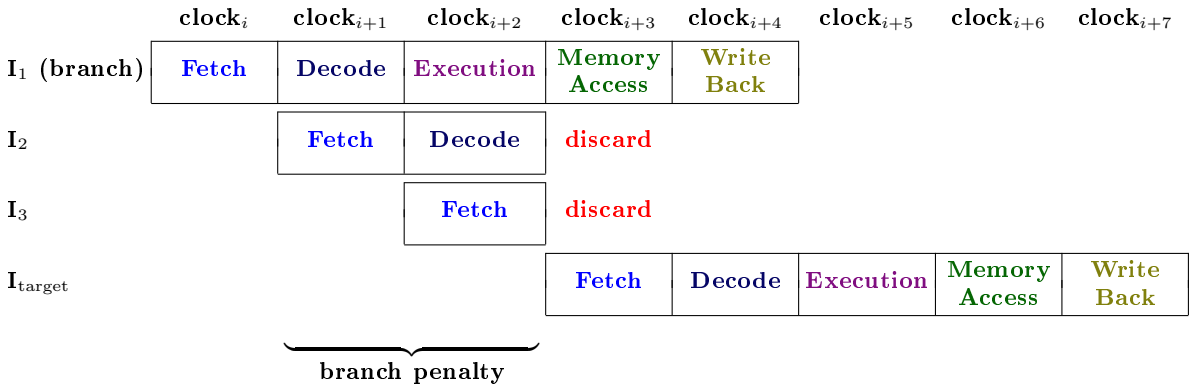
Při provádění druhé instrukce v pořadí (`B`/`JMP`) je cíl skoku známý až po několika krocích v pipeline (nejdřív je fetch, pak decode, a až v třetí fázi první instrukce zjistíme cíl). Jenže další instrukce jde do pipeline jen o krok později (protože „nějaká“ instrukce tam holt jít musí), stejně tak třetí v dalším kroku (skok nás má poslat úplně jinam, ale zatím nevíme kam, takže provedeme `SUB`). Kvůli zpoždění při vyhodnocování jsou další dvě instrukce v pipeline opožděny/zahozeny.

Počet provedených kroků, které ve skutečnosti nebudou využity, se nazývá *branch penalty*. Příliš vysoká branch penalty snižuje výkon procesoru, tedy je dobré tento parametr řešit. Jsou dvě možnosti:

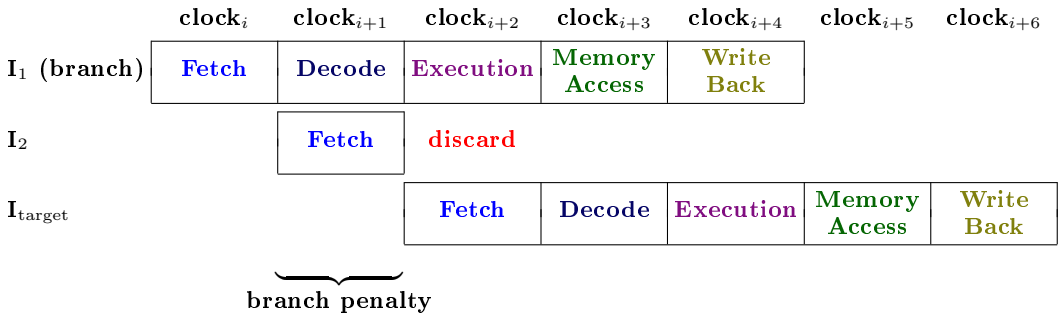
- výpočet cíle skoku v dřívějším kroku v pipeline a přenos v rámci komponenty,

- přeuspořádání instrukcí tak, že instrukce před skokem (v příkladu `ADD`) se přesune až za instrukci skoku, tedy nejdřív vyhodnocujeme cíl skoku, o krok dále začneme sčítat (tu instrukci provedeme určitě, nezahodíme), mezitím bude zjištěn cíl skoku, takže zjistíme, kam odskočit.

První možnost předpokládá aktualizaci registru PC nebo IP (podle architektury – ARM nebo Intel) u instrukce skoku už ve fázi decoding. Srovnejte obrázky 5.15 a 5.16. První obrázek odpovídá architektuře, kdy je registr PC/IP (má obsahovat adresu následující instrukce, která má být provedena) aktualizován až po fázi execute instrukce `B`, resp. `JMP`, druhý obrázek zachycuje „rychlejší“ architekturu, kdy je registr aktualizován už ke konci fáze decode.



Obrázek 5.15: Branch penalty, adresa cíle skoku je předána do registru PC/IP po fázi Execution



Obrázek 5.16: Branch penalty, adresa cíle skoku je předána do registru PC/IP během fáze Decode

Pokud zároveň použijeme i druhou možnost řešení (vyměníme pořadí u instrukce skoku a původní předchozí instrukce), nebudeme muset u „rychlejší“ architektury dokonce vůbec nic zahazovat. Samozřejmě podmínkou je, že vyměňované instrukce nesmí být navzájem závislé.

Pokud řízení toku instrukcí závisí na výsledku instrukce skoku, která ještě neopustila pipeline, tuto situaci nazýváme *control hazard*.

Podmíněné větvení kódu (podmíněné skoky) je důsledkem buď jednoduchého větvení (strojový kód vzniklý přeložením příkazů `if`, `switch`, apod.), nebo implementace smyček (příkazy `while`,...).

Příklad

Opět následuje úsek kódu pro procesory Intel (pro ARM by byl podobný, až na registry a formát čísla). První instrukce násobí číslem 25 obsah registru, druhá instrukce srovnává obsah dvou registrů a ukládá výsledek do registru příznaků. Třetí instrukce si v registru příznaků zjistí, jestli u srovnávaných hodnot je první větší nebo rovná druhé (JGE – Jump if Greater or Equal), a když ano, odskočí na zadané návěští.

```
            SUB ebx, 25
            CMP eax, edx
            JGE someLABEL
            ADD eax, 32
            ...
someLABEL:  MOV edx, eax
```

Je zřejmé, že nejdřív musí být provedena srovnávací instrukce `CMP` a až potom `JGE` (Jump if greater or equal), protože tu máme závislost. Porovnávací instrukce ukládá hodnotu do registru, ze kterého následně instrukce skoku čte. Jenže zápis do registru příznaků se opět musí stihnout dřív než z něj bude číst instrukce skoku. Na rozdíl od nepodmíněného skoku, zde nemůžeme tyto dvě instrukce přehodit.

Ovšem je tady ještě instrukce `SUB` před testováním. Když nemůžeme přehodit `CMP` a `JGE`, tak vložíme `SUB` mezi tyto vázané instrukce a budeme mít posloupnost `CMP`, `SUB`, `JGE`. Instrukce `SUB` sice taky zapisuje do registru příznaků, ale v pozdějším kroku, než ve kterém `JGE` z tohoto registru čte výsledek porovnávání. Takže zase jsme ušetřili minimálně jednu fázi.

Pokud by před `CMP` byla taková instrukce, kterou by nešlo přehodit, museli bychom za `CMP` místo toho dát prázdný krok Stall, resp. při softwarovém řešení instrukci NOP.

Zbývá vyřešit stejný problém jako u nepodmíněného skoku, tedy „kolizi“ přístupů pro zápis a čtení u registru PC/IP s adresou následující instrukce k provádění. Následující instrukce `ADD` buď bude přeskočena (pokud testování dopadne úspěšně) nebo provedena (pokud testování skončí hodnotou false). Takže máme přibližně 50procentní šanci, že když tuto instrukci začneme provádět, nebudeme ji pak muset zahazovat. 50procentní šance je pořád lepší než zbytečně vkládat prázdné intervaly do pipeline.

Takže v případě, že testování skončí false, je branch delay nulová, a v případě, že testování skončí true, zahodíme jeden nebo několik kroků a pokračujeme instrukcí `MOV` v cíli skoku.

5.5.5 Branch Prediction

V předchozím příkladu je naznačeno, že u podmíněných skoků se zřejmě bude pracovat s pravděpodobnostmi. Pravděpodobnost, že skok nepřijde, je sice přibližně kolem 50 procent, ale hýbe se nahoru nebo dolů podle určitých pravidel:

- když už jsme uvnitř smyčky (třeba `while`), obvykle ji provádíme více než jednou, i více než tisíckrát,
- stává se ovšem, že podmínka není splněna ani při prvním testování, tedy dovnitř smyčky se vůbec nedostaneme,
- pokud jsou smyčky vnořené, předchozí dva faktory se kombinují.

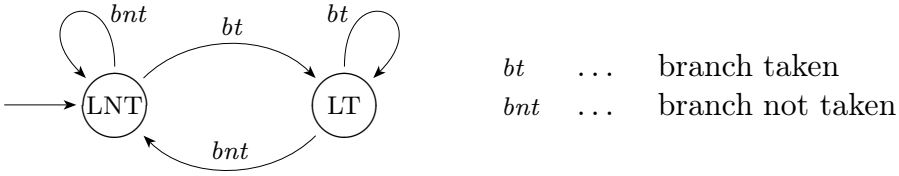
Jednodušší algoritmus pro předpovídání výsledku větvení (branch prediction) spočívá v tom, že instrukce používá bit (takže dvě hodnoty) nabývající buď hodnoty „pravděpodobně bude skok“ nebo hodnoty „pravděpodobně nebude skok“. Nejdřív je ve stavu „nebude“ (pravděpodobnost je menší než 50 %), ale když už se v kódu dostaneme dovnitř smyčky, nabyde bit hodnoty „bude“, protože pravděpodobnost, že zůstaneme uvnitř, vzroste nad 50 %.

Teď trochu předběhneme, s následujícími strukturami se setkáte až v předmětu o teoretické informatice. Náš systém se pohybuje mezi dvěma stavy:

- pravděpodobnost skoku větší než 50 %, tedy pokračování v aktuální větvi nízké, přeskočíme smyčku (LNT – branch Likely Not Taken),
- pravděpodobnost skoku menší než 50 %, tedy zřejmě setrváme v té větvi, ve které jsme a jdeme dovnitř smyčky (LT – branch Likely Taken).

Přechody mezi stavy můžeme zachytit jako šipky, šipku označíme popiskem určujícím příčinu přechodu. Například než se dostaneme do smyčky, jsme ve stavu LNT (pravděpodobně smyčku přeskočíme). Pokud poprvé byla podmínka vyhodnocena jako true a je třeba jít dovnitř, je pravděpodobné, že uvnitř smyčky nějakou dobu zůstaneme, tedy se přesuneme do stavu LT.

Výsledné schéma zakreslujeme jako graf, kterému říkáme konečný automat (protože má konečný počet stavů) a vidíme ho na obrázku 5.17.



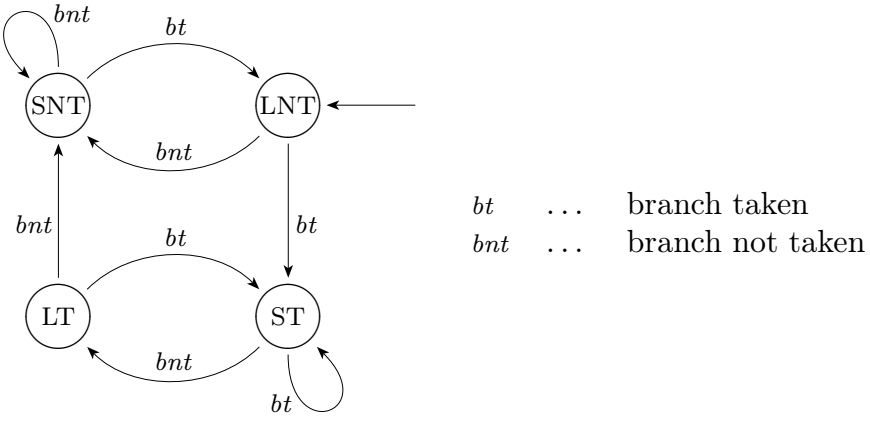
Obrázek 5.17: Konečný automat pro jednoduchý algoritmus předvídání větvení

Pokročilejší algoritmus branch prediction je již složitější, počítá s vnořováním smyček a místo dvou stavů (menší než 50 %, větší než 50 %) máme stavů více.

Pokud podle předchozího jednoduchého algoritmu opakovaně přicházíme k téže smyčce (protože opakovaně provádíme „nadřízenou“ smyčku), pak vlastně opakovaně „skáčeme v pravděpodobnostech“. Takže místo dvou stavů budeme mít čtyři pro různé pravděpodobnosti toho, že v současné větvi (posloupnosti kódu) zůstaneme, jak vidíme na obrázku 5.18.

- LT – branch is likely to be taken (něco nad 50 %),
- ST – branch is strongly likely to be taken (o dost víc než 50 %),
- LNT – branch is likely not to be taken (něco pod 50 %),
- SNT – strongly likely not to be taken (o dost míň než 50 %).

Při implementaci potřebujeme místo jednoho bitu dva bity, což dává čtyři stavy.



Obrázek 5.18: Konečný automat pro komplexnější algoritmus předvídání větvení

No dobře, ale ten jeden nebo dva bity potřebujeme mít někde uložené, navíc zvlášť pro různé instrukce podmíněného větvení, protože právě u takové instrukce potřebujeme vědět, jestli už jsme u ní někdy byli nebo k ní přicházíme poprvé (popřípadě v kombinaci s informací o nadřízené smyčce). Takovému místu se říká *branch target buffer* (tedy buffer cílů větvení), a pro každou větvící instrukci, kterou jsme v poslední době prošli nebo byla jiným způsobem zjištěna, zde máme tyto informace:

- adresa větvící instrukce,
- jeden nebo dva bity pro výše popsaný algoritmus,
- adresa cíle skoku, abychom v případě, že musíme skočit, mohli hned reagovat.

Takže když se provádí taková instrukce, ve fázi 1 (fetch) v pipeline je zkontrolováno, jestli pro instrukci existuje záznam v branch target bufferu. Když ne, v průběhu provádění instrukce se vytvoří (cíl skoku

se doplní až jako poslední) a nastaví se počáteční stav LNT. Když už záznam existuje, použije se podle algoritmu. Ovšem buffer není bezedný, záznamy se průběžně promazávají.

Další informace

- https://slideplayer.com/slide/5019089/
- http://www2.cs.siu.edu/~cs401/Textbook/ch3.pdf
- https://en.wikibooks.org/wiki/X86_Disassembly/Branches

5.5.6 Mechanismy optimalizace superskalárních procesorů

Speculative Execution. Při překladu příkazů `if`, `switch`, `while` a dalších příkazů s podmínkami se běžně do kódu dostává velké množství větvení. Speculative Execution (spekulativní vyhodnocování kódu) má právě zefektivnit provádění větveného kódu (branching).

Díky pipeliningu procesorová jádra běžně zpracovávají více instrukcí zároveň a je škoda nechávat pipeline zahálet jenom proto, že nemáme jistotu, kterou větví se má jít a jestli má dojít ke skoku. Proto procesory „odhadují“ předpokládaný další průběh výpočtu a spekulativně (tj. bez jistoty využití) provádějí úseky kódu ve větvích. Jde vlastně o jakousi nástavbu algoritmu Branch Prediction.

Spekulativní (nejisté) instrukce se vyhodnocují stejným způsobem jako ty „jisté“, jen mají nastavený speciální příznak „speculative“. Závěrečné fáze, které by jinak zapisovaly výsledek do paměti a registrů, zapisují do speciální cache, aby skutečné výpočty zůstaly konzistentní. V případě, že tyto instrukce jsou potvrzeny (přestanou být spekulativními), je jejich výsledek přenesen z cache do požadovaných cílových umístění (paměti a registrů), ale pokud budou zamítnuty (spekulace nevyšla), bude jejich výsledek zahozen.

Spekulativní vyhodnocování kódu sice vyžaduje přídavné hardwarové komponenty a mikrokód v procesoru a zvyšuje spotřebu energie, ale má velmi podstatný pozitivní vliv na výkon procesoru.

Out-of-Order Execution. V části textu o pipeliningu a datových závislostech jsme se dozvěděli, že některé kroky vyhodnocování instrukce musejí počkat, vkládají se intervaly Stall (bubliny). Určitou pomocí může být přeuspořádání instrukcí, o kterém již taky byla zmínka.

Tento koncept používá další typ bufferu, kterému říkáme *fronta instrukcí* (instruction queue). Instrukce přicházející ke zpracování čekají v tomto bufferu, k nim se zaznamenává informace o tom, jestli jsou všechny jejich vstupní operandy dostupné (konzistentní). Pokud má instrukce všechny operandy připravené (není nutno čekat na jinou instrukci, než je vypočte či zapíše), může ve frontě postoupit kupředu.

První fáze zpracování instrukce se obvykle nazývá *fetch*. Současné procesory mají pro tuto fázi sdílenou jednotku přes všechny pipelines, označuje se *parallel fetch unit* a dokáže v jediném kroku (nebo pár krocích) najednou načíst několik instrukcí a přenést je do fronty instrukcí. Další komponenta pracující přes všechny pipelines je *dispatch unit* provádějící fázi *decode* (včetně přípravy operandů) a obsluhující frontu instrukcí. Příklad pro ARM Cortex A55 jsme viděli na obrázku 5.12 (strana 92).

A právě ve frontě instrukcí může docházet k jejich přeuspořádání, protože zde máme také informace o závislostech mezi instrukcemi. Přeuspořádání může (staticky) provádět buď překladač během překladu do strojového kódu, nebo (dynamicky) procesor při provádění kódu.

Out-of-Order Execution (vyhodnocování mimo pořadí) znamená provádění instrukcí v jiném pořadí než jak přicházejí do procesoru, dynamické přeuspořádání. Pokud má instrukce zkompletovány operandy, může být přiřazena do příslušné pipeline. Až je provedena, mohou být operandy, které mění, ještě před finálním

uložením do paměti a registrů předány také čekajícím instrukcím ve frontě instrukcí, aby se urychlilo jejich předání do pipelines.

První pokusy s prováděním instrukcí mimo pořadí byly již v roce 1964, první mikroprocesor používající Out-of-Order byl Power1 od IBM (pouze pro operace s čísly v pohyblivé řádové čárce), později se tento mechanismus stává běžným, protože se ukázalo, že takováto optimalizace činnosti procesoru (jádra) může jeho činnost hodně urychlit.

Poznámka

Procesory Intel Atom od začátku nebyly vybaveny podporou Out-of-Order. Důsledkem byly špatné výsledky ve výkonnostních testech (a obecně špatný výkon) i při poměrně vysokém taktování a více jádrech. Tato funkce v Atomech nebyla implementována, protože vyžaduje místo pro umístění příslušných integrovaných obvodů (Atomy mají být co nejmenší) a procesor se více zahřívá, potřebuje lepší chlazení (což by taky vadilo).

Mechanismus Out-of-Order Execution do značné míry závisí na následujícím algoritmu.

Register Renaming. Algoritmus přejmenovávání registrů (Register Renaming) slouží k odbourání některých datových závislostí. U registrů rozlišujeme *názvy* (names) a *umístění* (locations), každé umístění může mít více názvů. Procesor si udržuje mapovací tabulku zvanou *register map table* a používá dynamickou mapovací funkci, aby zachoval vztah mezi umístěními a názvy registrů.

V instrukcích jsou „fyzické“ názvy registrů (tedy vlastně lokace) nahrazeny virtuálními názvy, a vztah mezi virtuálním názvem a umístěním je uložen do mapovací tabulky. Se vstupními a výstupními registry se zachází odlišně. Pokud pro určité (vstupní) umístění existuje více názvů, pak u instrukce volíme poslední existující virtuální název. Pro výstupní registr instrukce se vytvoří nový název.

Postup

Předpokládejme tuto posloupnost instrukcí (ARM kód):

```
ADD R1, R2, R3
MUL R3, R1, R2
SUB R1, R2, R3
ADD R2, R1, R3
```

Postupně si ukážeme transformaci všech těchto instrukcí během zařazování do fronty instrukcí. První instrukce je zpracována následovně (pro cílový registr, tedy výstup, je zvolen nový název):

Původní instrukce	**R1**	**R2**	**R3**	...	**Transformovaná instrukce**
počáteční názvy:	V1	V2	V3	...	
ADD R1, R2, R3	V4				ADD V4, V2, V3

Druhá instrukce používá registr R1, který byl přepsán předchozí instrukcí, takže musíme použít nový virtuální název, abychom neztratili vazbu na zdroj dat. Jako cílový registr opět dosadíme nové jméno.

Původní instrukce	**R1**	**R2**	**R3**	...	**Transformovaná instrukce**
počáteční názvy:	V1	V2	V3	...	
ADD R1, R2, R3	V4				ADD V4, V2, V3
MUL R3, R1, R2			V5		MUL V5, V4, V2

Zrychlíme. Doplníme zbývající instrukce:

Původní instrukce	**R1**	**R2**	**R3**	...	**Transformovaná instrukce**
počáteční názvy:	V1	V2	V3	...	
ADD R1, R2, R3	V4				ADD V4, V2, V3
MUL R3, R1, R2			V5		MUL V5, V4, V2
SUB R1, R2, R3	V6				SUB V6, V2, V5
ADD R2, R1, R3		V7			ADD V7, V6, V5

Starší virtuální názvy jsou odstraněny hned poté, co z fronty zmizí instrukce, které tyto názvy používaly, a mohou být recyklovány pro pojemnování cílových operandů dalších instrukcí, tedy nehrozí, že by názvy „došly“. Takto uvolněné názvy jsou vedeny ve speciální tabulce volných názvů.

5.5.7 Zneužití mechanismů superskalárních architektur

Myšlenka superskalárních architektur, přeuspořádání instrukcí, spekulativního vyhodnocování apod. je dobrá, ale bylo zjištěno, že tyto mechanismy, zejména Speculative Execution, se dají zneužít ke spuštění nechtěného či škodlivého kódu (proniknutí bariéry mezi uživatelským a privilegovaným módem, proniknutí bariéry mezi jednotlivými aplikacemi) nebo k proniknutí přes bariéru mezi virtualizovaným a hostitelským operačním systémem. Tyto informace se objevily i v tisku a mnoho lidí zná názvy těchto zranitelností:

- Meltdown,
- Spectre,
- Spoiler.

Zranitelnostmi Meltdown a Spectre trpí všechny současné procesory od Intelu a AMD a některé procesory ARM. Spoiler je pouze na procesorech Intel.

Přímá oprava není dost dobře možná, to by znamenalo přepsání mikrokódu, což je vlastně jakýsi firmware procesoru, nebo dokonce pozměnění tištěných obvodů, což je ještě problematičtější. Nicméně pro Meltdown a Spectre existují záplaty v operačních systémech, které sice zranitelnosti neodstraní, ale znemožní jejich zneužití. Pro Spoiler neexistuje záplata, není řešitelný softwarově.

Další informace

Superskalární architektury:

- http://people.ee.duke.edu/~sorin/ece252/lectures/4.2-tomasulo.pdf
- https://www.d.umn.edu/~gshute/arch/register-renaming.xhtml
- http://hpca23.cse.tamu.edu/taco/utsa-www/cs5513-fall07/lecture5.html

Zranitelnosti superskalárních architektur:

- https://meltdownattack.com/
- https://diit.cz/clanek/intel-spoiler-bezpecnostni-diru-lze-vyuzit-i-javascriptem
- https://www.extremetech.com/computing/261439-spectre-meltdown-new-critical-security-flaws-explored-explained
- https://arxiv.org/pdf/1903.00446.pdf
- https://www.techrepublic.com/article/spectre-and-meltdown-explained-a-comprehensive-guide-for-professionals/

5.6 Další vlastnosti procesorů

5.6.1 Podpora virtualizace

Virtuální stroj je součást systému, která má přidělenu část zdrojů systému (paměť, čas procesoru apod.), slouží k izolování konkrétních procesů od systému (proces běžící ve virtuálním stroji vidí jen tento virtuální stroj a nic jiného včetně skutečného systému). Virtuální počítače se používají tehdy, když v systému chceme spustit „cizí“ aplikaci programovanou pro jiný systém (jinak by ta aplikace nebyla schopná v tomto systému pracovat), a také to může být jakási ochrana (chceme zabránit izolované aplikaci zasahovat do hlavního systému).

Existuje více typů virtualizace, pro ten jednodušší (provozování jiného operačního systému a jeho aplikací ve Windows, Linuxu, MacOS apod.) máme například virtuální stroje VirtualBox, VMWare Workstation, Parallels Desktop, KVM, Xen.

Virtualizace si postupně našla cestu i k procesorům. Pokud procesor podporuje virtualizaci, pak virtuální stroj běží mnohem efektivněji, rychlost běhu virtualizovaného systému se téměř rovná rychlosti běhu nativních nevirtualizovaných procesů. Podpora u výrobců procesorů:

- AMD: technologie Pacifica (AMD-V).
- Intel: technologie Vanderpool (Intel VT).
- ARM: pro hardwarovou virtualizaci existuje speciální režim: hypervisor mode.

To, zda konkrétní procesor umí hardwarovou virtualizaci, zjistíme na webu výrobce.

Kromě základní hardwarové virtualizace mohou procesory podporovat přídavné virtualizační funkce, například k intelovské technologii VT se přidává VT-d, což je virtualizace komunikace s I/O zařízeními.

Poznámka

Kromě procesoru musí hardwarovou virtualizaci podporovat BIOS (tj. základní deska) a chipset. V BIOSu je virtualizace někdy vypnuta, protože ji zneužíval rootkit Blue Pill.

Aby mohla hardwarová virtualizace fungovat, musí být podporována také u softwarových virtualizačních nástrojů (jinak tyto nástroje sice fungují, ale pomaleji), což je už dnes běžné.

Další informace

Pod Windows potřebujeme pro zjištění virtualizace například CPU-Z nebo jiný podobný nástroj, v Linuxu se obvykle stačí podívat do některých souborů. Souhrn o detekci hardwarové virtualizace v Linuxu je na http://www.cyberciti.biz/faq/linux-xen-vmware-kvm-intel-vt-amd-v-support/.

Virtualizace na procesorech ARM: https://genode.org/documentation/articles/arm_virtualization.

5.6.2 Energetická náročnost a správa energie

V současné době je jedním z nejdůležitějších parametrů množství spotřebované energie a prakticky všichni výrobci procesorů nabízejí ekologické varianty svých procesorů s nízkou spotřebou.

Hodnota, podle které se posuzuje energetická náročnost procesoru, se označuje *TDP* (Thermal Design Power) – u Intelu, AMD původně používal pojem *Wattage*, teď také TDP. Udává se ve Watech.

Uvedená hodnota je důležitá nejen pro odhad energetické spotřeby procesoru, ale především pro určení vhodné úrovně chlazení (protože se vzrůstající spotřebou roste množství produkovaného tepla, které je nutno od procesoru odvést), takže je tím určeno, jaký chladič vlastně potřebujeme.

U obou společností je tím míněno něco podobného, ale každý výrobce „svůj“ pojem definuje trochu jinak. Souhrnně se dá říct, že TDP odpovídá maximálnímu množství energie, kterou procesor odebírá v standardním nastavení. Ale pozor, i to standardní nastavení nemusí odpovídat tomu, s jakými parametry reálně procesor kupujeme. Navíc základní deska (resp. její BIOS) do tohoto nastavení také může zasáhnout (a u dražších komponent také běžně zasahuje).

Poznámka

Vzhledem k tomu, že Intel a AMD tento pojem definují odlišně, nelze podle tohoto parametru jejich produkty srovnávat. Navíc jejich procesory mají různě implementovány i technologie řízení spotřeby, tedy spotřeba a vyzařování tepla za běžného provozu bude odlišná.

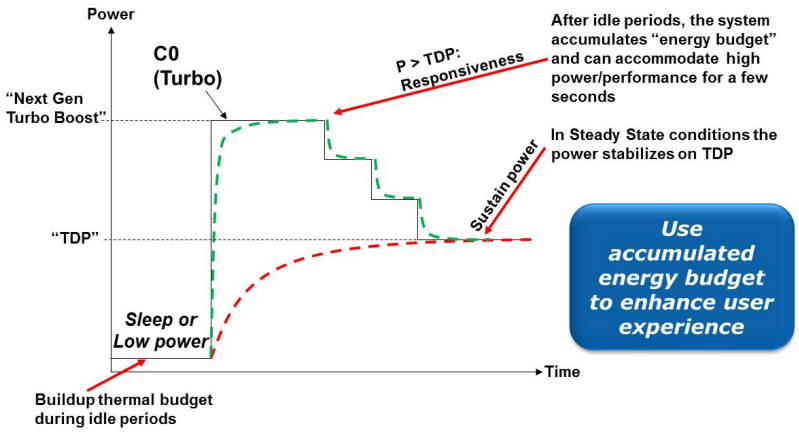
Zatímco běžné procesory mají TDP běžně v desítkách W (někdy i přes 100), procesory určené do slabých notebooků (či netbooků) mívají TDP obvykle kolem 10 W nebo v jednotkách Wattů. Procesory typu ULV (Ultra Low Voltage) mají TDP také nízké, jsou určeny do ultrabooků.

Jak bylo výše zmíněno, existují různé technologie správy energie. Na nejdůležitější se nyní podíváme.

Úspora energie. Intel SpeedStep, AMD PowerNow! (obojí pro notebooky), AMD Cool'n'Quiet (pro desktopy) a PowerTune (pro grafické čipy) jsou technologie správy energie, kde hlavním cílem je spořit energii. Tyto technologie obvykle fungují tak, že frekvence výpočetního čipu se snižuje, pokud není moc vytížen, a tedy frekvence se dynamicky mění. Důsledkem je, že při menším zatížení se snižuje spotřeba energie, zařízení se méně zahřívá a aktivní chladiče (větráčky) se mohou otáčet pomaleji (nebo se zastaví).

Dočasné zvyšování výkonu. Pro tento účel se používá technologie Turbo Boost (Intel) nebo Turbo Core (AMD). Účelem je vyždímat z procesoru co největší výkon tak, abychom se ještě „vešli do TDP“, tedy tak, aby při navýšení frekvence teplota nestoupla až moc. Představují možnost *krátkodobého* (cca 25 s) několikanásobného navýšení frekvence jader při zátěži.

Frekvence nesmí být vysoká moc dlouho, protože procesor začne vyzařovat více tepla a při překročení hranice stanovené hodnotou TDP by nebylo možné procesor uchladit (důsledkem by byla nestabilita ve výpočtech, pád systému, v horším případě „uvaření“ nebo vypnutí procesoru), proto se po krátkodobém navýšení frekvence postupně snižuje tak, aby chlazení pořád zůstalo pod hranicí danou TDP (teplota roste postupně, tedy i frekvence klesá „synchronizovaně“).



Obrázek 5.19: Průběh změny frekvence u Intel Turbo Boost 2.0[5]

[5] Zdroj: https://software.intel.com/en-us/articles/intel-xeon-processor-e5-26004600-product-family-technical-overview

Tato technologie je přínosná také tehdy, když je jedno jádro vytíženo více než ostatní jádra (je zvýšena jeho frekvence) – technologii lze uplatňovat zvlášť na jednotlivá jádra.

Intel Turbo Boost verze 2.0 tuto funkci ještě vylepšuje, navíc přibírá do party i grafické jádro (pokud je integrováno v procesoru), tedy i jeho frekvenci přizpůsobuje. AMD Turbo Core funguje podobně, včetně zapojení grafických jader.

Na obrázku 5.19 je graficky znázorněna závislost mezi hodnotou TDP a dynamicky měněnou frekvencí. Frekvence (zeleně) zároveň s odběrem energie rychle stoupá, zároveň s tím začne pomalu stoupat teplota (protože procesor zvyšuje odběr energie, a tedy se víc energie přemění na teplo). Teplota sice stoupá pomaleji než frekvence, ale těsně před tím, než se dostane k hranici TDP, je potřeba jednat – rychle o něco snížíme frekvenci, čímž se zpomalí nárůst teploty. Stejný skok ve frekvenci se provede ještě nekolikrát (teplota pořád stoupá, ale čím dál pomaleji), až se dostaneme do víceméně stabilního stavu (teplota je na hranici TDP, ale už nestoupá). Podobné korekce se provádějí i dále, ale „skoky“ už nejsou tak divoké. Pokud procesor (vlastně jádro) zrovna nemá co dělat, frekvence naopak klesne.

Poznámka

U procesorů se rozlišuje běžná základní frekvence (Base Frequency) a turbo frekvence (po zvýšení pomocí Turbo Boost/Turbo Core), takže ve specifikaci procesorů v tom případě najdeme dva údaje o frekvencích – jeden pro základní režim, druhý pro turbo režim. Pokud ve specifikaci vidíme jen základní frekvenci, pak tento procesor zřejmě nepodporuje Turbo režim.

Později si pro jednotlivé výrobce ukážeme, jak zjistit TDP a další vlastnosti procesorů.

Intel přišel s podobně nazvanou technologií: Turbo Boost Max (TBMT – TurboBoost Max Technology). Jde o to, že jádra procesoru mohou být z výroby různě kvalitní, třebaže jsou vyrobena na jediné desce. V procesoru s TBMT jsou jádra seřazena do posloupnosti od „nejlepšího“ (superior) po „nejhorší“, přičemž ta nejlepší mohou bez problémů běžet na vyšších frekvencích, tedy například jsou přednostně přetaktovávána a jsou na ně předávána nejnáročnější výpočetní vlákna. Na těchto jádrech může dočasně vzrůst frekvence nad rámec Turbo Boost frekvence (cca o 200 MHz vyšší hranice).

5.6.3 Modularita procesorů

Některé současné procesory jsou modulární (opravdu si nemůžeme poskládat procesor podle svého uvážení) – ovšem nikoliv z pohledu uživatele, spíše z pohledu výroby a návrhu.

S modulárními procesory přišla jako první společnost AMD, šlo o architekturu Bulldozer. Procesor byl poskládán z modulů obsahujících vždy dvojici jader pro celočíselné výpočty (každé obsahuje několik pipelines) a jedno sdílené vlákno FPU pro výpočty v pohyblivé řádové čárce (FPU je v samostatné části čipu a je vícevláknová, s několika pipelines).

Každé jádro má vlastní L1 cache, jádra v rámci jednoho modulu sdílejí společnou L2 cache a dále existuje L3 cache sdílená jádry přes všechny moduly procesoru. Jádra v jednom modulu sdílejí také dekodér instrukcí. A to právě byl kámen úrazu: pokud má jádro začít zpracovávat novou instrukci, musí ji dekódovat. Jenže pokud je v téže fázi zpracování instrukcí i druhé jádro, jedno z nich musí počkat. Taktéž sdílení jednoho vlákna FPU v rámci modulu bylo až příliš limitující. Proto nebyl nárůst výkonu až tak velký, jak se očekávalo.

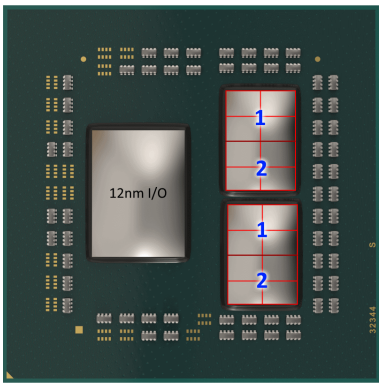
Nicméně společnost AMD to nevzdala. V nové architektuře Zen byly taky použity moduly, ale už se tolik nešetřilo. Zcela se mění architektura procesoru. Procesor se skládá z jednoho nebo více *CCX*

(Core CompleX), jakýchsi modulů, ale tentokrát v jiném smyslu než v předchozích generacích. Například osmijádrový procesor architektury Zen se skládá ze dvou CCX. Jednotka CCX obsahuje 4 výpočetní jádra, která mají každé svou L1 a L2 cache, L3 cache je v rámci jednotky sdílená. Každé jádro má vlastní ALU a vlastní vlákno do FPU (resp. CCX obsahuje 4 ALU a 4 vlákna FPU).

Následná architektura Zen 2 přišla s další změnou v modularitě – procesor je poskládán z relativně samostatných modulů, kterým se říká *čiplet* (chiplet, přesněji Core Chiplet Die, CCD). Jde o samostatné malé desky tištěných spojů, na každé z nich najdeme dva moduly CCX (ty jsou o polovinu menší než v předchozí generaci). Protože CCX obsahuje čtyři jádra, je v jednom čipletu osm jader, nebo méně, protože některá jádra mohou být deaktivovaná.

Procesor může obsahovat jeden nebo více čipletů s běžnými procesorovými jádry (7nm litografie), a dále tam najdeme speciální I/O čiplet, který může být vyroben jinou litografií (u Zen 2 se používá I/O čiplet vyrobený 14nm nebo 12nm technologií, a to od GlobalFoundries, třebaže 7nm výpočetní čiplety vyrábí TSMC).

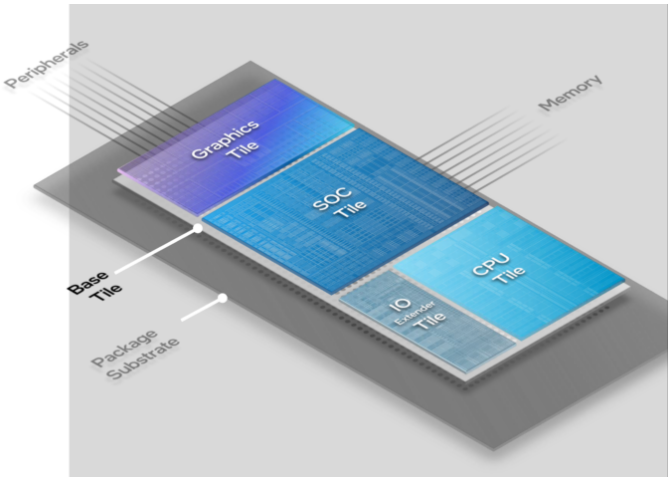
V procesorech architektury Zen 3 a 4 také najdeme čiplety obsahující CCX modul, a to pouze jeden, ovšem tento CCX modul obsahuje osm jader místo čtyř. Takže celkový počet jader v jednom čipletu se nemění, jen jsou ze dvou CCX modulů shrnuta do jednoho. Důvodem je pravděpodobně to, že AMD do budoucna počítá pouze s výrobou mnohajádrových procesorů a segmentování po 4 jádrech se mu zdálo zbytečné a neefektivní.



Obrázek 5.20: Čiplety v procesoru Ryzen třetí generace (Zen 2)[6]

AMD používá modulární strukturu spíše u desktopových procesorů, kdežto procesory pro mobilní zařízení (notebooky apod.) zůstávají monolitické. Výhodou modulárního procesoru je možnost „poskládat“ různě výkonné a různě vybavené procesory tak, že se zkombinuje vhodný I/O čiplet (původní severní most čipsetu) a několik čipletů s výpočetními jádry.

Později přišel s modulárními procesory také Intel. Poprvé se tak stalo u procesorů mikroarchitektury Meteor Lake, kde najdeme výpočetní část, část s grafickým jádrem, SoC čiplet s funkčností původního severního mostu a dále I/O čiplet obsahující řadiče některých komunikačních protokolů, které by jinak byly v jižním mostě (včetně podpory Thunderbolt 4, další PCIe linky apod.).



Obrázek 5.21: Modulární struktura procesorů Intel Meteor Lake[7]

[6] Zdroj: https://wccftech.com/ryzen-3000-series-ccx-and-core-layout-quick-guide/

5.6.4 Little a Big Endian

Předpokládejme, že ukládáme velké číslo (více B) do paměti. V jakém pořadí se jednotlivé Byty celého datového typu ukládají? Existují tyto tři základní postupy:

- *Little Endian* – nejnižší Byte se zapisuje na nižší adresu (procesory Intel, DEC Alpha)
- *Big Endian* – nejvyšší Byte se zapisuje na nižší adresu (procesory Motorola, SPARC)
- *Mixed Endian* – pomíchaně (procesor PDP)

Power PC od IBM dokáže pracovat oběma postupy. Rozlišuje dva módy: big a little, mezi nimi se přepíná podle momentálního nastavení.

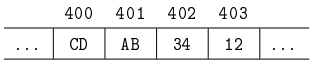
Příklad

Na adresu 400 (hexadecimálně) uložíme hexadecimální číslo 1234ABCD:

```
mov EAX, 1234ABCDh
mov [400h], EAX     ; EAX je 32-bitový
```

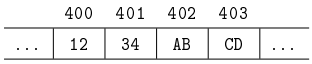
Little Endian:

- adresa 400h = CDh
- adresa 401h = ABh
- adresa 402h = 34h
- adresa 403h = 12h



Big Endian:

- adresa 400h = 12h
- adresa 401h = 34h
- adresa 402h = ABh
- adresa 403h = CDh



Problémy s endians mohou nastat při komunikaci s jiným počítačem v síti, který používá jiné pořadí, anebo při komunikaci se zařízením, které používá jiné pořadí (ovladače). Většina zařízení, která se připojují k počítačům, je Little Endian, I/O funkce operačních systémů s tím počítají a Big Endian systém automaticky provádí konverze. V UNIXových systémech navíc existují Big Endian varianty I/O funkcí pro případ komunikace s „výjimkami“, a pokud zrovna taková funkce neexistuje, zbývá ruční konverze.

5.6.5 Vícevláknové aplikace

Výpočetní vlákno je vlastně podproces, s tímto pojmem jsme se setkali v sekci o pipelines. Ve ve vícevláknových systémech se každý proces skládá nejméně z jednoho vlákna. Každé vlákno má svůj vlastní programový kód a další prostředky, vlákna téhož procesu jsou navzájem téměř nezávislá. Podle typu operačního systému se obvykle procesor přiděluje jednotlivým vláknům, nikoliv procesům. Rozlišujeme vlákna aplikační, systémová a hardwarová.

Aplikační vlákna jsou vlákna běžící v uživatelském režimu. Každý proces má nejméně jedno takové vlákno (hlavní vlákno) a pak případně další, každé vlákno má své identifikační údaje, kód, programové zásobníky a datový prostor.

Systémová vlákna jsou vlákna operačního systému. Jádro operačního systému (kernel) může mít výpočetní vlákna pro svou potřebu, a dále vlákna využívaná aplikačními vlákny při systémových voláních (aplikační vlákno se napojí na systémové vlákno a přes ně využívá služby operačního systému a hardwaru).

[7] Zdroj: https://wccftech.com/intel-lego-like-chiplet-design-for-next-gen-meteor-lake-arrow-lake-lunar-lake-3d-foveros-cpus-detailed/

Hardwarová vlákna jsou vlákna vykonávaná některým jádrem procesoru. Každé výpočetní jádro procesoru může zpracovávat kód nezávisle na ostatních, tedy představuje jedno hardwarové vlákno, a pokud je procesor superskalární (tj. používá hyperthreading/SMT), může být v jednom jádře zpracováváno několik vláken zároveň. Operační systém plánuje (přiděluje) hardwarová vlákna jednotlivým systémovým a aplikačním vláknům.

Poznámka

Pokud programujeme vícevláknovou aplikaci, je třeba si rozmyslet, který kód bude které vlákno vykonávat. Většinou přiřazujeme vláknu konkrétní funkci, kterou má provádět, a především se staráme o to, aby jednotlivá vlákna mohla pracovat na sobě nezávisle. Pokud jedno vlákno musí počkat, až jiné vlákno něco uloží do sdílené proměnné, pak to není vhodná struktura pro vícevláknovou aplikaci.

5.7 Patice procesorů

Procesor se na základní desce zasazuje do *patice*. Dnes se výhradně používají patice typu *socket* (čtvercové nebo obdélníkové, piny jsou uspořádány přibližně do matice). V minulosti se krátce používaly patice typu *slot* (podobné jako sloty různých sběrnic na základní desce, piny byly uspořádány do řady).

U socektů se můžeme setkat se těmito typy provedení:

1. *PGA* (Pin Grid Array) – piny na procesoru jsou uspořádány v matici, jim odpovídají prohlubně na socketu.
2. *ZIF* (Zero Insertion Force) – socket je opatřen páčkou, která slouží k připevnění čipu procesoru, účelem je minimalizovat škody působené zasouváním a vyndáváním čipů ze socketu.
3. *BGA* (Ball Grid Array) – místo pinů jsou na procesoru napájené „kuličky“, prohlubně (spíše měděné plošky) na socketu jsou rozmístěny v matici podle kuliček na procesoru (piny se občas ulamovaly, zde je snaha se jich zbavit); výhodou je lepší odvod tepla, vyšší hustota „kuliček“, menší el. odpor; nevýhody jsou spíše mechanického rázu a souvisejí s nutností důkladnějšího připevnění procesoru.
4. *LGA* (Land Grid Array) – na procesoru jsou plošky (napájená měď, obvykle pozlacená), na socketu jsou piny.

ZIF je „přídavný“ mechanismus, který se typicky kombinuje s něčím jiným. Například současné patice typu PGA jsou většinou PGA-ZIF (například AM4 pro některé procesory AMD), tedy k patici PGA se přidá páčka pro lepší uchycení procesoru, nebo LGA-ZIF (novější patice AMD a všechny současné patice Intelu). Přídavek ZIF se obvykle už ani nepíše. Patice BGA najdeme především v noteboocích.

Další informace

Pěkný článek o rozdílech mezi LGA a BGA najdeme na
https://www.lioncircuits.com/blog/posts/lga-vs-bga

Na konci 90. let se objevilo několik procesorů do patice typu *slot* (piny nebyly v matici, ale v jedné řadě), například Intel Pentium III nebo AMD Athlon dané generace. Hodně staré procesory patřily do patice typu DIP („švábi“ s nožičkami na dvou protilehlých stranách, např. Intel 8086) nebo podobné PLCC (nožičky na všech čtyřech stranách, např. Intel 80386).

Intel. Desktopové procesory vyráběné Intelem používají výhradně patice typu LGA, označují se číslem odpovídajícím počtu pinů/kuliček. Například LGA 1200 a LGA 1700 jsou patice používané pro běžné desktopové procesory, LGA 2066 je momentálně pro desktopové high-end procesory (Core-X Series). Pro notebookové procesory Intel používá patici BGA nebo v některých případech je procesor přímo připájen na základní desku.

Starší desktopové procesory patří do patic LGA 1151, LGA 1150, LGA 1155, LGA 1156, atd. pro nižší a střední třídu, LGA 2011-v3, LGA 2011, LGA 1366 pro high-end procesory. Seznamy jdou „proti proudu času“.

Poznámka

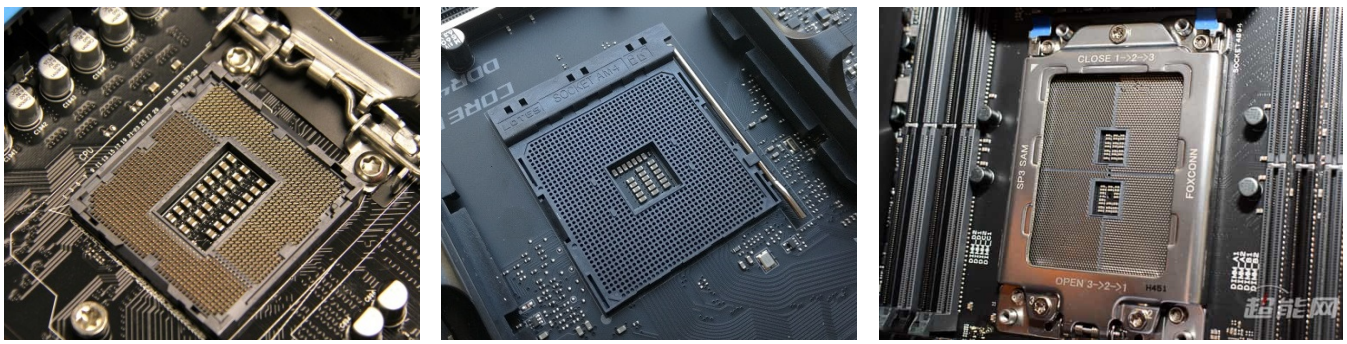
Jak bylo výše uvedeno, číslo u označení intelovských patic znamená počet pinů. Přes piny se procesor propojuje se základní deskou, tedy jejich počet i rozmístění musí přesně sedět. Všimněte si, že například mezi LGA 1150 a LGA 1151 je rozdíl jediného pinu – nejsou hardwarově kompatibilní (procesor určený pro jednu patici nejde zapojit do druhé).

Nekompatibilita však nemusí být jenom fyzická, ale taky se může stát, že novější procesor nebude fungovat v patici na starší desce, do které by fyzicky pasoval – třeba tehdy, když mezi vyrobením základní desky a procesoru uběhne víc než rok, technologie se totiž mění i v rámci generace a novější procesor může vyžadovat novější čipset. Proto je dobré si prověřit, jestli si nový procesor bude rozumět se starší základní deskou.

Pokud novější procesor by měl umět spolupracovat s novějším čipsetem, ale přesto to nefunguje, může pomoci upgrade BIOSu/UEFI.

AMD. Desktopové procesory nižší a střední třídy vyráběné společností AMD patří většinou do patice AM4, v roce 2022 přichází novinka AM5. Současné high-end procesory (Threadripper) používají patice TR4, sTRX4 nebo nejnovější sTR5. Zatímco patice AM4 je typu PGA-ZIF, patice AM5, TR4, sTRX4 a sTR5 jsou typu LGA-ZIF. Notebookové procesory AMD nepočítají s nějakými výměnami, tedy připevnění je na výrobci notebooku, často bývají připájené nebo v patici BGA.

Patice TR4 a její novější varianty jsou zvláštní nejen tím, že na rozdíl od AM2–4 jsou typu LGA, ale navíc jejich upínací mechanismus je šroubovací (potřebujete šrouby Torx). Je to zřejmě tím, že jsou ve skutečnosti odvozeny od patice SP3 určené pro serverové procesory Epyc (serverové patice mají běžně šroubovací upínací mechanismus).



Obrázek 5.22: Sockety Intel LGA 1150, AMD AM4 a AMD TR4[8]

[8] Zdroje: https://www.pcmag.com/article/361381/buying-a-motherboard-20-terms-you-need-to-know, https://www.cnews.cz/desky-platformy-amd-x399-socket-tr4-chladice-pro-procesory-threadripper-computex/

Serverové procesory Epyc patří do patice SP3, která je typu LGA. Starší desktopové procesory AMD patří do jednoho z těchto typů patic:

- AM2, AM2+, AM3, AM3+ – do těchto patic patří procesory bez integrované grafiky (AMD FX a starší Athlony),
- FM1, FM2, FM2+ – pro procesory s integrovanou grafikou, starší zástupci řady „A-Series“.

Patice FMx nejsou navzájem kompatibilní (například procesor FM2+ nezasuneme do patice FM2). Patice AM2, AM2+, AM3, AM3+ jsou do určité míry zpětně kompatibilní (vždy vedlejší generace – procesor AM3+ například můžeme zasunout do patice AM3 a bude to fungovat, i když ne zcela optimálně).

Poznámka

O kompatibilitě platí u patic AMD totéž jako v případě Intelu – to, že procesor do dané patice fyzicky patří, ještě neznamená, že v ní bude bez problémů fungovat. Pokud vyžaduje novější čipset, nebude fungovat v základní desce se starším čipsetem. V některých případech stačí upgrade BIOSu. Pokud kupujete novější procesor do starší desky, ověřte si, za jakých okolností bude možné ho použít. Pokud je nutný upgrade BIOSu, musí být proveden s takovým procesorem, který je v desce schopen fungovat před upgradem.

5.8 Přehled procesorů

Podíváme se na procesory společností Intel a AMD, ke konci pak stručně na procesory ARM. Každý výrobce používá svou vlastní terminologii, má své vlastní tempo uvádění nových typů procesorů a taky své vlastní názvy používaných technologií.

5.8.1 Architektura a mikroarchitektura u Intelu

Architektura procesoru určuje, co a na kterých místech je v procesoru zahrnuto (komponenty), jak co s čím komunikuje (návrh interních datových, adresových a řídicích sběrnic): koncept ALU a FPU, struktura jader, pipelines, registry, adresování paměti, přístup do cache, speciální druhy cache jako je např. TLB, správa napájení, instrukční sady, jejich implementace, skalární architektury, branch prediction, atd.

Mikroarchitektura procesoru je dána těmito parametry:

- architektura,
- výrobní proces, zejména litografie (64nm, 45nm, 32nm, 22nm, 14nm, atd.).

Tedy mohou existovat procesory, které mají stejnou architekturu, ale jsou vyráběny odlišnými výrobními procesy, mají proto jinou mikroarchitekturu. Podobně je to i v případě procesorů, které jsou vyráběny tímtéž výrobním procesem, ale liší se v architektuře (jak jsou uvnitř uspořádány).

5.8.2 Typy procesorů Intel a systém Tick-Tock

U Intelu se dříve používal systém „Tick-Tock“: stejná architektura se nejdřív implementuje pomocí „ověřené“ výrobní technologie, pak pomocí nové technologie výroby; na této nové technologii výroby, která se tak stane ověřenou, se potom postaví nová architektura, atd. Tedy se střídají fáze změny výrobní technologie (tick) a fáze změny architektury (tock). Průniky jsou mikroarchitektury.

Například mikroarchitektury Nehalem a Westmere patří k architektuře Nehalem (ale každá mají jiný výrobní proces), následující dvě patří k architektuře Sandy Bridge (taky se liší výrobním procesem). U pozdějších procesorů se tento systém změnil.

	Core		*Nehalem*		*Sandy Bridge*		*Haswell*		*Skylake*	
	65 nm		**45 nm**		**32 nm**		**22 nm**		**14 nm**	
	Tick	*Tock*	*Tick*	*Tock*	*Tick*	*Tock*	*Tick*	*Tock*	*Tick*	*Tock*
	NetBurst	Merom	Penryn	Nehalem	Westmere	Sandy Bridge	Ivy Bridge	Haswell	Broadwell	Skylake
	2005–2006		2007–2008		2009–2010		2011–2012		2014–2015	

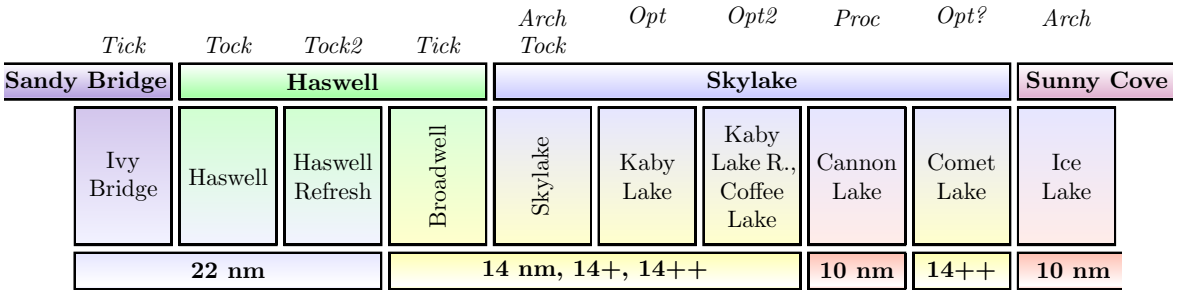
Tabulka 5.4: Tick-Tock u procesorů Intel (shora architektura, výrobní proces, mikroarchitektura)

Poznámka

Všimněte si, že systém Tick-Tock do roku 2015 přibližně zachovává Mooreův zákon – co dva roky se změní výrobní proces, zmenší se tranzistory (třebaže ne zcela na polovinu, s těmi čísly to je stejně trochu složitější), a tedy do procesoru o zhruba stejném rozměru se vejde mnohem víc tranzistorů.

V rámci architektury Skylake se původně podle Tick-Tock mělo přecházet na 10nm výrobní proces, ale nepovedlo se, Intel začíná mít s miniaturizací vážné problémy. Pro přechod na 10nm litografii si tedy nechal delší čas a místo toho svůj dvoufázový model Tick-Tock upravil na třífázový: *Tick-Tock-Optimization*, který označuje *PAO* (Process-Architecture-Optimization).

Od té doby už není až tak přesně určeno, kdy se který typ parametru změní, nicméně ke každému procesoru se tato informace dá dohledat.



Obrázek 5.23: Souhrn aktuálních mikroarchitektur Intelu, stav k roku 2019

U posledních mikroarchitektur Intel postupně přestává používat rozdělení architektura/mikroarchitektura, ale spíše používá rozdělení mikroarchitektura/codename (kódové jméno). Takže například u procesorů s výrobním procesem Intel 7 se dočteme o mikroarchitektuře Golden Cove a codename Alder Lake, následně mikroarchitektuře Raptor Cove a codename Raptor Lake.

Aby těch názvů nebylo málo, vězte, že mikroarchitektur je ve skutečnosti ještě více. Například pro podobnou (ne úplně stejnou) architekturu a výrobní proces ještě máme například odlehčené mikroarchitektury (taky hovoříme o *typech jader*), například Braswell (obdoba Broadwellu), Apollo Lake (obdoba mikroarchitektury Skylake) a Gemini Lake (obdoba Kyby Lake) jsou mikroarchitektury SoC čipů určené pro slabší procesory typu Pentium Silver a Celeron. Gemini Lake používá architekturu Goldmont+, která je odvozena od Skylaku. Zároveň s Coffee Lake se objevila mikroarchitektura Whiskey Lake určená pro mobilní zařízení s velmi malou spotřebou, a mikroarchitektura Amber Lake, která je ještě úspornější. A to jsme se ještě moc nezdrželi u serverů...

Generace	Mikroarchitektura	Litografie	Patice na desktopu	Fáze
1. generace	Nehalem Westmere	45 nm 32 nm	LGA 1156, LGA 1366	Tock Tick
2. generace	Sandy Bridge	32 nm	LGA 1155, LGA 2011	Tock
3. generace	Ivy Bridge	22 nm	LGA 1155, LGA 2011, LGA 2011-1	Tick
4. generace	Haswell, Haswell Refresh	22 nm	LGA 1150, LGA 2011-v3	Tock
5. generace	Broadwell	14 nm	LGA 1150, LGA 2011-v3	Tick
6. generace	Skylake	14 nm	LGA 1151, LGA 2066	Tock, Arch
7. generace	Kaby Lake	14+	LGA 1151, LGA 2066	Opt
8. generace	Kaby Lake Refresh, Coffee Lake Cannon Lake	14++ 10 nm	LGA 1151 mobilní	Opt2 Proc
9. generace	Coffee Lake	14++	LGA 1151	Opt?
10. generace	Comet Lake Ice Lake	14++ 10nm	LGA 1200 mobilní	Opt? Arch
11. generace	Tiger Lake	10 nm	BGA	Opt
12. generace	Alder Lake	Intel 7	LGA 1700	Proc
13. generace	Raptor Lake	Intel 7	LGA1700	Opt

Tabulka 5.5: Generace Core i a odpovídající mikroarchitektury

Další informace

Podrobnější informace o intelovském mikroarchitekturním chaosu najdeme například na
https://en.wikipedia.org/wiki/List_of_Intel_CPU_microarchitectures

5.8.3 Přehled procesorů Intelu

Dále se podíváme na typy procesorů. Od architektury Nehalem označuje Intel většinu svých procesorů jako „Core i“, nověji pouze „Core“, přičemž k označení dodává informaci o škálování podle výkonu. Dále najdeme slabší procesory pojmenované Pentium, Celeron a Atom. Na opačném spektru výkonnosti jsou procesory Intel Xeon určené pro servery nebo hodně výkonné počítače, a varianty pro IoT zařízení (Internet of Things).

U procesorů řady Core se můžeme setkat s tímto názvoslovím:

- Core 3, Core 5, Core 7 pro běžné procesory,
- Core Ulta 5, Core Ultra 7, Core Ultra 9 pro výkonnější high-endové procesory,
- Core i3, Core i5, Core i7, Core i9 je starší označení procesorů patřících do prvních dvou skupin.

Čím vyšší číslo, tím lepší parametry (frekvence, počet jader apod.). Jde vlastně o výkonnostní index.

Součástí označení bývá generace, dále číslo upřesňující výkonnostní index, a pak tam mohou být písmena určující specifické parametry. Například písmeno „K“ znamená odemčený násobič. Níže v seznamu odkazů najdete adresu stránky s přehledem používaných písmen.

Procesory *Pentium* a *Celeron* ještě pořád najdeme na trhu, ale Intel je pomalu vytěsňuje ze své nabídky a nahrazuje slabšími procesory z rodiny Core. Jde o spíše slabší procesory (na úrovni nebo těsně pod úrovní

Core 3) – Pentia jsou o něco výkonnější než Celerony. Mají spíše nízký výkon, ale také mají nízkou spotřebu a méně se zahřívají. Jsou levnější, šetří se u nich například na množství cache paměti a na obvodech pro akceleraci výkonu (většina z těchto procesorů neumí Turbo Boost). Většinou mají integrované grafické jádro, spíše méně výkonné.

Procesory *Atom* jsou na nejnižší příčce žebříčku výkonnosti. V porovnání s předchozími jsou výkonově nejslabší, často jsou přiletovány na základní desce (tedy žádná patice). Atomy většinou nepodporují Out-of-Order execution, protože tato technologie navyšuje velikost čipu a je výpočetně náročnější. Některé Atomy míří do IoT segmentu.

Procesory *Xeon* jsou primárně určeny pro servery a případně výkonné grafické stanice, ale existují i varianty pro úplně jiné způsoby využití. V posledních generacích u nich Intel používá označování podle kovů (Platinum, Gold, Silver apod.) – čím dražší kov, tím výkonnější procesor...

Úkol

Především je dobré vědět, kde to všechno najdeme, a kde případně zjistíme další konkrétní parametry. Procesory Intelu (a vlastně taky další produkty, Intel toho vyrábí víc) jsou vcelku podrobně zdokumentovány na https://intel.com, v menu najdeme *Products*.

Po proletění marketingových informací se na této stránce dají najít technické údaje o jednotlivých procesorech. Vyberte si několik procesorů různého typu a najděte možnost srovnání jejich parametrů.

Další informace

- Čísla a písmena v označení procesorů: https://www.intel.com/content/www/us/en/processors/processor-numbers.html
- Výroba procesorů Ivy Bridge: https://www.youtube.com/watch?v=d9SWNLZvA8g
- Pohled dovnitř továrny: https://www.youtube.com/watch?v=-KTKg0Y1snQ
- Typy procesorů Intel: https://www.which.co.uk/reviews/laptops/article/intel-processors-explained-what-is-core-i3-i5-i7-and-pentium-av6235O66lQP
- https://en.wikichip.org/wiki/intel/process-architecture-optimization

5.8.4 Intel vs. AMD

Společnost AMD vyrábí kromě jiného procesory x86/amd64 kompatibilní, tj. kompatibilní s Intelem. Roku 1982 společnosti Intel a AMD uzavřely smlouvu o poskytnutí licencí na procesory, zejména i8086 a i8088. Společnost Intel byla k této smlouvě víceméně dotlačena svým odběratelem, společností IBM, která chtěla mít pro své počítače alespoň dva ekonomicky nezávislé dodavatele procesorů. Poté, co procesory od AMD začaly zákazníky zajímat pomalu více než procesory od samotného Intelu, se Intel pokusil smlouvu zrušit, ale neúspěšně. Společnost AMD dále vytvářela procesory, které byly víceméně klony těch intelovských, ale čím dál víc začala přidávat vlastní technologie a vylepšení.

Jako první například přišla s NX bitem pro označování paměťových stránek jako čistě datových, jako první prodávala procesory s odemčeným násobičem (až později s tím přišel i Intel, když bylo zřejmé, že zákazníci o něco takového opravdu stojí), jako první přešla na 64bitovou instrukční sadu (taky proto se 64bitové procesory s touto architekturou označují amd64), jako první přenesla část funkcionality čipsetu do procesoru (paměťový řadič v architektuře K8, celý severní most v mikroarchitektuře K9, Intel s obojím

opět přišel později). Navíc procesory AMD byly většinou o něco levnější než srovnatelné procesory od Intelu.

Dlouhá léta byl konkurenční boj mezi Intelem a AMD hnací silou inovací, tyto společnosti se navzájem pokoušely „trumfnout“. Později však AMD nestíhala přecházet na „menší tranzistory“, nedařilo se miniaturizovat výrobní proces podle Mooreova zákona, takže Intel dlouho vedl – procesory od AMD používaly starší výrobní proces, což znamenalo horší poměr výkonu a spotřeby.

Společnost AMD si bohužel podrážela nohy i jinak – postupně se objevilo několik případů, kdy AMD uvedla novou řadu procesorů příliš brzy – nová technologie nebyla řádně odladěná, vyskytovaly se chyby ve firmwaru (výpočetní, ve stabilitě apod.), od toho se odvíjely velké náklady na vyřízení reklamací a dodatečné změny ve výrobě.

Jenže potom přišla společnost AMD s architekturou Zen, která se ukázala opět v mnohém inovativní. Modulární architekturu sice uvedla už dříve, ale v procesorech architektury Zen už moduly fungují velmi dobře.

Další informace

- Nárůst výkonu CPU – AMD vs. Intel:
  https://pctuning.cz/article/narust-vykonu-cpu-za-poslednich-osm-let-amd-vs-intel?chapter=3
- https://www.digitaltrends.com/computing/amd-vs-intel/

5.8.5 Typy procesorů AMD

Než se podíváme na architektury a konkrétní procesory, podíváme se na typy procesorů od AMD. Novější procesorové řady jsou následující:

- *Ryzen* – hlavní řada (pro mainstream), na desktopu patří do patic AM4 nebo novější AM5. Právě s Ryzeny přišlo „zmrtvýchvstání“ procesorů AMD. U části Ryzenů najdeme integrovanou grafiku.
- *Ryzen Threadripper* – high-endové procesory patřící do LGA patice sTR5 nebo některé starší (TR4, sTRX4). Velmi výkonné, dražší.
- *Athlon* – „odlehčené“ procesory pro nenáročné použití. Některé z nich jsou APU, jiné neobsahují grafické jádro. Desktopové varianty patří do stejné patice jako Ryzeny.
- *Epyc* – většinou určeny pro servery, patří do patice SP6 nebo starší.
- Pokud za označením najdete „PRO“, znamená to, že jde o firemní procesory, které jsou sice výkonné, ale nepočítá se u nich s přetaktováváním (nemají odemčený násobič).

U AMD sice také rozlišujeme architektury, ale v mnohem obecnějším smyslu slova. Zhruba od roku 2003 se drží architektura *x86-64* (podle toho, že jde o 64bitové procesory kompatibilní s intelovskými), členíme ji na rodiny. Nicméně v textech o procesorech AMD se jako architektura často označuje to, co dále nazýváme rodina.

Rodiny patřící do téže architektury se navzájem liší především architektonickými změnami (přidání nové funkce, navýšení určitého parametru, změny ve struktuře jader, změnami v instrukčních sadách a jejich implementaci apod.), v některých případech i přechodem na lepší výrobní proces.

Momentálně se setkáváme především s rodinou Zen (17h), do které patří jak první generace procesorů Zen (přišla roku 2017 s 14nm výrobním procesem), tak i její vylepšení Zen+ (12nm litografie), druhá generace Zen 2 (z roku 2019, přechod na 7nm litografii), následuje třetí generace Zen 3 (litografie 7nm+). Do těchto „zenových“ rodin řadíme procesory Ryzen a Threadripper.

Na podzim roku 2022 se objevuje Zen 4 určená pro Ryzeny 7. generace (pozor, je zde čím dál větší nesoulad mezi generacemi architektury a generacemi procesorů).

Pokud zahrneme i historii, vypadá to následovně:

- K8 (Hammer) – zde AMD poprvé přišla s přesunem paměťového řadiče do procesoru, podřízená rodina K9 integrovala do některých procesorů celý severní most čipsetu,
- K10 – AMD přichází s APU, dělí procesory do řad podle toho, jestli (většinou) obsahují či neobsahují grafické jádro,
- Bulldozer (15h) – nástupce K10 pro výkonnější stroje (střední a vyšší třída),
- Bobcat (14h) – ve stejné době jako Bulldozer, nástupce K10 s úsporným provozem,
- Jaguar (16h) – nástupce Bobcatu, přechod na 28nm výrobní proces,
- Zen (17h).

Bulldozer a Bobcat přišly zhruba ve stejné době. Obě rodiny jsou nástupci K10, ale každá je založena na jiném principu – Bulldozer je pro výkonnější procesory, u Bobcatu je zase kladen důraz na úspornost a nízkou cenu.

Mikroarchitektury vždy určitým směrem rozvíjejí danou architekturu, liší se většinou drobnějšími architektonickými změnami. Do určité míry se pojmy rodina a mikroarchitektura kryjí, ale do jedné rodiny může patřit i více mikroarchitektur na sebe navazujících.

Poznámka

Společnost AMD jako první přišla s procesory, které mají odemčený násobič. U novějších procesorů řady Ryzen bývá násobič obvykle odemčený a lze je tedy přetaktovávat změnou hodnoty násobiče, ale ve variantách PRO používaných ve firmách obvykle násobič není odemčený.

5.8.6 Přehled procesorů AMD

V Bulldozerech se objevila nová struktura jader, přesněji byla značně pozměněna komunikace mezi běžnými jádry (CPU) a grafickým jádrem (GPU). Tato komunikační struktura se označuje jako unifikovaná architektura – *Heterogeneous System Architecture* (HSA), přičemž o všech těchto jádrech hovoří jako o *výpočetních jádrech* (Compute Core), tedy v tomto pojmu shrnuje běžná i grafická jádra. Odlišnost je především v tom, že CPU a GPU jádra na rozdíl od předchozího řešení komunikují s okolím naprosto stejně, stejně přistupují k paměti, například už se nedělí běžná operační paměť (pro CPU) a videopaměť (pro GPU).

Oficiální definice výpočetního jádra od AMD hovoří o tom, že jde o jakékoliv jádro, na kterém může běžet výpočetní proces ve svém vlastním kontextu a přidělené virtuální paměti, nezávisle na jiných jádrech (v předchozím řešení bylo GPU úkolováno některým CPU). Pojem „proces“ zde znamená jednoduše souvislou posloupnost instrukcí posílanou ke zpracování operačním systémem.

Poznámka

HSA ve skutečnosti není jen záležitostí procesorů AMD, byla vyvinuta konsorciem, jehož členem je také ARM. Takže s touto architekturou se setkáváme také v malých mobilních zařízeních se SoC čipy ARM.

Zen ze začátku roku 2017 je nová rodina procesorů, přičemž hlavní změnou oproti předchozím je přechod na 14nm výrobní proces. Konečně, Intel vyráběl 14nm procesory už o dva roky dřív.

Jak jsme se dozvěděli v jedné z předchozích sekcí o modularitě, desktopové procesory rodiny Zen se skládají z jednoho nebo více *CCX modulů*. Další změnou je zavedení *SMT (Simultaneous Multi-Threading)*, což je prakticky totéž jako u Intelu HT (HyperThreading) v superskalárním systému.

Procesory *Ryzen* vzbudily velkou pozornost. Výkonově se prakticky vyrovnají intelovským procesorům řady Core i. Na desktopu patří do patice AM4 nebo AM5. AMD rozlišuje výkonnostní třídy:

- Ryzen 3 (tj. R3) jsou slabší procesory,
- Ryzen 5 (tj. R5) je střední třída,
- Ryzen 7 (tj. R7) jsou výkonnější procesory,
- Ryzen 9 (tj. R9) je nejnovější počin (R9 se objevily brzy poté, co Intel přišel s Core i9).

Podobnost se značením Intelu je samozřejmě čistě náhodná...

Ryzeny se objevily v únoru 2017. Na podzim téhož roku přišly high-endové procesory *Threadripper* určené do nové patice TR4 (a pak novějších variant). Postupně se objevily varianty „PRO“ určené pro firmy. Většinu procesorů bez tohoto označení lze přetaktovávat změnou hodnoty násobiče, „PRO“ varianty ne.

Ryzeny a Threadrippery se v rámci rodiny dají rozdělit do generací, které jsou patrné i v označení procesoru. Pozor, generace Ryzenu a Threadripperu nejsou totéž jako generace rodiny Zen.

Architektura Zen 2 navíc přišla s opravou zranitelnosti Spectre přímo v hardwaru, takže nejsou nutné záplaty operačního systému, které by znesnadňovaly zneužití tohoto bezpečnostního problému.

Úkol

Podíváme se na procesory AMD. Informace o nich najdeme na https://amd.com, v menu najdeme Products, pak Processors. Jednotlivé odkazy v zobrazeném menu vedou k webovým stránkám, kde pod velkým množstvím marketingových informací najdeme obvykle tabulku s vlastnostmi jednotlivých procesorů.

V části menu Resources je odkaz Product Specifications vedoucí k přehledové tabulce s většinou prodávaných procesorů, ve které můžeme také filtrovat a vyhledávat.

Projděte si uvedený web. Najděte několik procesorů různého typu (alespoň dva různé typy v desktopových, něco i v dalších kategoriích). Vyzkoušejte porovnání procesorů.

5.8.7 Procesory ARM

Instrukční sada procesorů ARM je typická RISC sada. Instrukce mají víceméně pevnou délku (32 nebo 64 bitů). Tyto procesory mají především velmi nízkou spotřebu, proto je dnes najdeme především v malých mobilních (telefony, tablety) a embedded zařízeních a v zařízeních Internetu věcí (IoT), ale používají se i na serverech (pole ARM procesorů může mít při stejném výkonu nižší spotřebu a méně se zahřívá než Intel/AMD).

Dnes se u procesorů ARM setkáváme nejčastěji s jádry typu *Cortex*, přičemž jde obvykle o odvozená jádra – výrobci si mohou původní návrh upravit, přidat další moduly (grafickou část, různé řadiče apod.) a dodat nové funkce. Velice často jde o úpravy vedoucí k vytvoření SoC čipů (přesněji: naprostá většina procesorů ARM jsou SoC čipy).

Je zajímavé, že pro ARM procesory není jen jediný výrobce, ale licence jsou poskytnuty více různým výrobcům. Tedy vývojem se zabývá společnost ARM Holdings, výrobou pak jiné společnosti. Licence mohou

být poskytnuty buď na celá předpřipravená jádra (tj. výrobce používá jádra ARM tak jak jsou navržená od ARM), nebo výrobce spolupracuje na vývoji jádra s ARM (specifické vlastnosti z tohoto vývoje jsou pak určeny jen danému výrobci), nebo může být licence na architekturu ARM (výrobce si navrhne svá vlastní jádra, jen musí být kompatibilní s ARM včetně podpory instrukčních sad), a nejvyšší typ je ARM Flexible Access, kdy výrobce má plný přístup k veškerým specifikacím daného typu procesoru.

ARM tedy licencuje výrobu svých procesorů různým výrobcům. Na ty nejznámější se podíváme.

- *Snapdragon* jsou procesory ARM vyráběné společností Qualcomm. Najdeme je ve smartphonech a tabletech mnoha různých výrobců. Jako prvním se těmto procesorům podařilo splnit požadavky Microsoftu na procesor pro tzv. Copilot PC (dostatečně výkonná jednotka NPU v procesoru, podpora instrukcí SSE 4.2, atd.).
- *Apple ARM* jsou ARM procesory vyráběné společností Apple pro vlastní zařízení. Nejdřív šlo o mobilní procesory pro chytré telefony a tablety (název začíná písmenem „A“), později se přidávají procesory pro notebooky a desktopy (začínají písmenem „M“). Další písmena jsou vyhrazena speciálnímu využití, například písmeno „H“ je určeno pro AirPods a písmeno „R“ pro senzory.
- *Exynos* jsou ARM procesory vyráběné společností Samsung. Některé z těchto procesorů najdeme také ve wearables, tedy nositelné elektronice.
- *Nvidia Tegra* jsou starší ARM čipy pro Nintendo a některá mobilní zařízení, novější *Nvidia Xavier NX* je čip pro roboty, drony, chytré kamery, autonomní vozidla.
- *Kyrin* jsou ARM procesory vyráběné společností Huawei (přesněji – výrobcem je čínská společnost HiSilicon, jejím vlastníkem je Huawei). Huawei je používá také pro svou odvozenou značku Honor.
- *Broadcom* vyrábí ARM procesory určené zejména pro síťová zařízení, ale najdeme je například také v populárních zařízeních Raspberry Pi.
- *Atmel* kromě vlastních řešení vyrábí procesory založené na ARM (zejm. Cortex), často ve formě MCU či SoC do embedded zařízení. Také některé modely Arduina používají procesor od Atmelu.
- *Marvel* vyrábí SoC založené na ARM, najdeme je ve smartphonech, embedded systémech, síťových zařízeních. Je zajímavé, že první licenci na ARM procesory koupil Marvel od Microsoftu.

Čipy ARM se používají buď jako plnohodnotné procesory nebo v roli „lepších“ řadičů. Řadiče SSD například bývají ARM čipy, protože běžné řadiče používané v klasických pevných discích by měly příliš nízký výkon.

Další výrobci procesorů (nejen ARM): Transmeta, IBM PowerPC, Motorola, MIPS, Alpha, Cyrix, SPARC, UltraSPARC, DEC (Digital Equipment Corporation), NEC (Nippon Electronic Company).

Další informace

- https://www.arm.com/products/silicon-ip-cpu
- Přehled desktop. procesorů: https://www.svethardware.cz/prehled-desktopovych-procesoru/22566
- https://www.root.cz/serialy/co-se-deje-v-pocitaci/

5.9 Komunikace se zařízeními

V této sekci se zabýváme nízkoúrovňovými možnostmi komunikace probíhající mezi komponentou a procesorem (nebo jinou komponentou).

5.9.1 IRQ

IRQ (Interrupt Request Line/Level) je požadavek přerušení, který vysílá zařízení v případě, že je třeba ošetřit událost se zařízením související. Je určen procesoru, který po obdržení IRQ přeruší svou činnost, aby mohl požadavek obsloužit. Účelem je co nejúspornějším způsobem (defacto předáním určitého čísla na určité místo) informovat procesor, že „se stalo něco, co vyžaduje pozornost“.

Příklad

Pokud stiskneme klávesu na klávesnici, očekáváme, že na tuto událost počítač okamžitě zareaguje (například vypíše na obrazovku příslušné písmeno nebo zareaguje na klávesovou zkratku). V klávesnici bývá malý pomocný procesor, který detekuje stisknuté klávesy, vytvoří k nim *kód klávesy* (ten určuje, která klávesa byla stisknuta, jestli zároveň byla stisknuta některá pomocná klávesa – Ctrl, Alt apod., jestli se jedná o stisknutí nebo uvolnění, atd.). Tento kód uloží na určené místo, aby mohl být zpracován, a pomocí IRQ „dá vědět“ procesoru, že je nutné věnovat pozornost klávesnici.

Staré klávesnice používaly IRQ 1 (takže když procesor dostal přerušení s číslem 1, bylo jasné, že je od klávesnice), současné klávesnice využívají některé IRQ vyhrazené pro rozhraní USB.

Další čísla IRQ jsou využívána třeba myší (toto IRQ je generováno při pohybu myší, stisku či uvolnění některého tlačítka apod.), zvukovou kartou (i pro mikrofon), časovačem, různými zařízeními připojenými k USB rozhraní, dokonce samotný procesor využívá IRQ (i když vlastně přerušuje sám sebe) – například když nastane událost dělení nulou, je třeba výpočet okamžitě přerušit a vypsat chybové hlášení.

Přerušení má na starosti obvod, kterému říkáme *řadič přerušení* (PIC – Programmed Interrupt Controller). Protože přerušení lze předávat různým jádrům procesoru, případně ve víceprocesorovém systému různým procesorům a zároveň na přerušeních stojí prakticky veškerá obsluha různých komponent, je řadič přerušení poměrně důležitým obvodem, jakýmsi „hlavním plánovačem“.

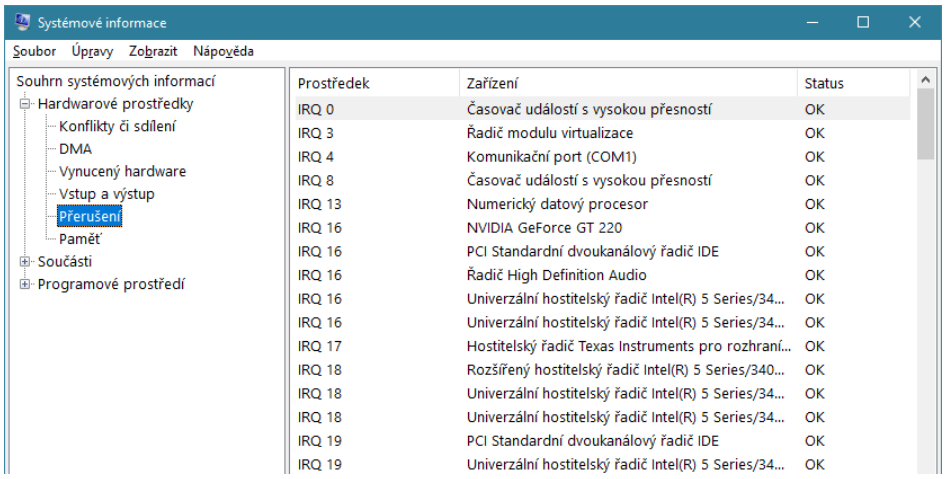
V současných architekturách se obvykle setkáváme s APIC (Advanced PIC) od Intelu, také existuje OpenPIC (dnes se už nepoužívá) a z něj odvozený MPIC (Multiprocessor PIC) od IBM. Na architektuře ARM se používá GIC (Generic Interrupt Controller), nebo příslušný výrobce použije vlastní obvod. Obvykle jsou lokální řadiče v jednotlivých procesorech a pak hlavní řadič obvykle v jižním mostě, což znamená, že přerušení budou korektně obsloužena i ve víceprocesorovém systému.

Zařízení, které chce využívat IRQ k informování procesoru, se předně musí *registrovat* k určitému číslu IRQ a v rámci registrace sděluje adresu *obslužné rutiny přerušení*, tedy jaký kód se má provést, pokud procesor dostane IRQ s daným číslem. Například u klávesnice bude na příslušné číslo IRQ navázána funkce, která umí zjistit, kterému procesu bylo stisknutí klávesy určeno (to může být třeba textový editor, když píšeme text), dokáže pracovat s kódem stisknuté klávesy a dotyčnému procesu tento kód předá.

Poznámka

Je samozřejmě možné (a naprosto běžné), že k témuž číslu IRQ se registruje více komponent. Pak hovoříme o *sdílení IRQ*. Zatímco v jednodušším případě procesor při příchodu IRQ požadavku jednoduše přejde na obslužnou rutinu, která byla k danému číslu IRQ registrována, při sdílení je třeba nejdřív určit, která rutina vlastně má být provedena (tj. od které komponenty sdílející totéž IRQ vlastně požadavek přišel). Například na obrázku 5.24 vidíme, že přerušení číslo 16 je sdíleno pěti zařízeními.

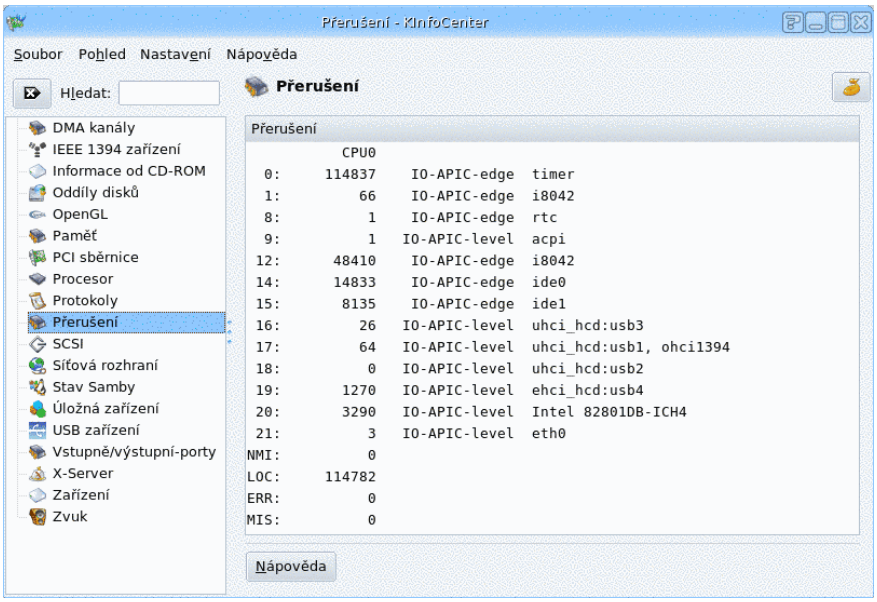
Za určitých okolností dochází místo sdílení ke *kolizi v IRQ*. Tento případ nastane, pokud totéž IRQ registuje více zařízení, ale některé z nich se sdílením „nepočítá“. Důsledkem je, že systém reaguje poněkud



Obrázek 5.24: Seznam IRQ ve Windows 10

neočekávaně – zařízení s tímto IRQ se chovají jinak než by měla, protože reagují na signály, které jsou určeny jinému zařízení.

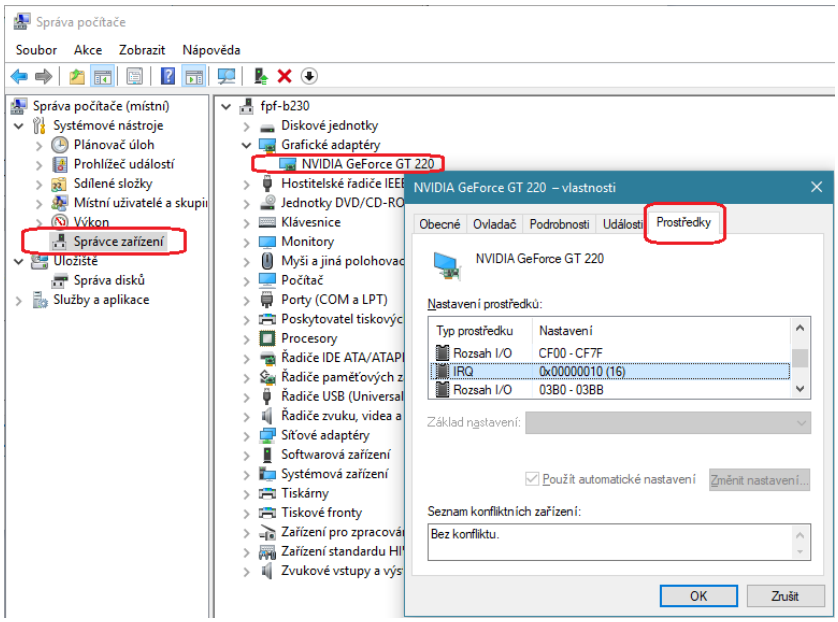
K přerušením se dostaneme ve Windows v nástroji *Systémové informace* (spustíme příkazem `msinfo32`, případně najdeme v *Systémových nástrojích*, obrázek 5.24). Je zde také část *Konflikty či sdílení*, kde můžeme odhalit případný problém s konflikty v IRQ. V UNIXových systémech obvykle najdeme nějaký nástroj přímo v grafickém prostředí (obrázek 5.25), případně si v textovém režimu vypíšeme obsah souboru `/proc/interrupts`.



Obrázek 5.25: Seznam IRQ v Linuxu (KInfoCenter)

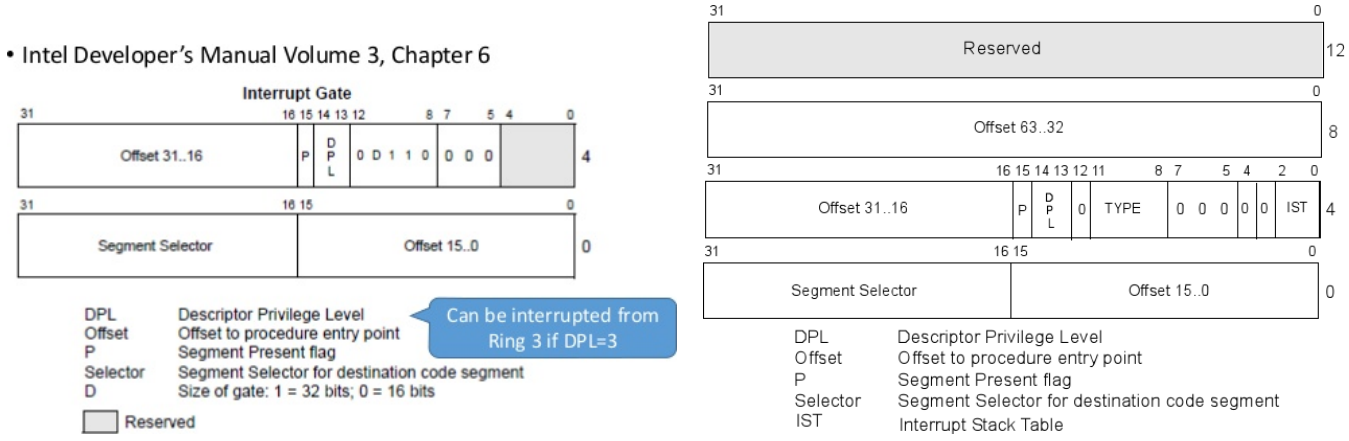
Pokud chceme ve Windows najít zdroje příslušející konkrétnímu zařízení, najdeme je přes *Správce zařízení*, v něm vybereme dotyčné zařízení, zobrazíme jeho vlastnosti, přejdeme na kartu *Prostředky* (obrázek 5.26). Jestliže dané zařízení v seznamu nenajdeme, zapneme v menu okna zobrazování skrytých zařízení.

Informace k jednotlivým IRQ jsou uloženy v poli, které se nazývá *Interrupt Descriptor Table* (IDT). Tato tabulka se používá na všech hardwarových architekturách, ale může mít různou podobu. Ve skutečnosti



Obrázek 5.26: Kde najdeme obecně zdroje ve Windows 10

ji vede operační systém, ale adresa jejího začátku je v jednom z registrů. Položky („řádky“) této tabulky jsou *vektory přerušení*, tedy jakési záznamy o probíhajícím přerušení s daným číslem (pro každé číslo IRQ je zde jeden vektor přerušení).



Obrázek 5.27: Formát položky v IDT pro architektury x86 a x64[9]

Formát vektorů přerušení pro architektury x86 a x64 je na obrázku 5.27. Nejdůležitější části jsou:

- DPL (Descriptor Privilege Level) – číslo od 0 do 3 určující úroveň oprávnění subjektu, který způsobil přerušení (0 pro zařízení nebo jádro – Ring0, 3 pro běžné procesy – Ring3), v současných operačních systémech se typicky povoluje jen 0,
- P (Present) – 1bitová informace, zda se tento konkrétní vektor používá nebo jestli je prázdný,
- Segment – určení segmentu (bloku v paměti), ve kterém se nachází obslužná rutina přerušení,
- Offset – offset (relativní adresa v segmentu) obslužné rutiny.

Identifikace obslužné rutiny (segment a offset) vede buď přímo k obslužné rutině, nebo k poli adres (či jiných identifikací) obslužných rutin pro všechna zařízení sdílející toto IRQ.

[9] Zdroje: https://www.slideshare.net/inaz2/abusing-interrupts-for-reliable-windows-kernel-exploitation-en, http://ethv.net/workshops/osdev/notes/notes-3

Většinu přerušení lze „maskovat“, tedy v době maskování přerušení procesor tato přerušení ignoruje. Například v době, kdy procesor zpracovává jedno přerušení, by nemělo nastat další; příp. pokud je zpracováván nějaký „citlivý“ kód jádra (například synchronizační úlohy), opět bývá použito maskování přerušení. Přerušení jsou maskována tehdy, když procesor z nějakého důvodu „nechce být rušen“. Některá přerušení nelze maskovat (např. dělení nulou).

Kód, který je obslužnou rutinou přerušení nebo vyžaduje maskování přerušení, by měl být co nejkratší, protože při jeho vyhodnocování systém nereaguje na žádná IRQ.

Implementace závisí na konkrétní hardwarové architektuře. Obvykle jde o jeden nebo několik bitů v některém registru takových, že nastavení na 1 znamená maskování.

Co se děje, když je vyvoláno přerušení (například po stisku klávesy):

- IRQ je po vygenerování komponentou nejdřív posláno do *řadiče přerušení* (PIC), který tato přerušení eviduje a informuje procesor, že došlo k přerušení (u Intelu to je signál INTR, sděluje „pozor, přišlo přerušení“),
- procesor zkontroluje, zda nemají být přerušení maskována, pokud ano, ignoruje je, pokud ne, postupujeme dále,
- procesor odpoví řadiči přerušení (u Intelu signálem INTA), tedy se ptá „o co jde?“,
- PIC najde a pošle procesoru záznam o přerušení z IDT,
- procesor uloží kontext (hlavně obsah registrů) předchozího výpočtu, ve spolupráci s operačním systémem určí obslužnou rutinu a informuje zařízení o tom, že rutina bude spuštěna,
- spustí obslužnou rutinu daného přerušení.

Řadič přerušení má omezenou kapacitu paměti, do které ukládá záznamy o přerušení. Proto pokud procesor dostatečně rychle „neodebírá“ tyto záznamy, novější přerušení přepisují ta dříve uložená. Důsledkem je ztrácení přerušení (uživatel má pak dojem, že je „ignorován“). Ovšem ne vždy se přerušení opravdu ztrácejí, jejich odebírání může být pouze zpomaleno.

Problematika IRQ je poněkud složitější. Existují různé možnosti hlášení IRQ, řešení souběhu žádostí v čase, priorit, paralelní obsluhy přerušení apod.

Další informace

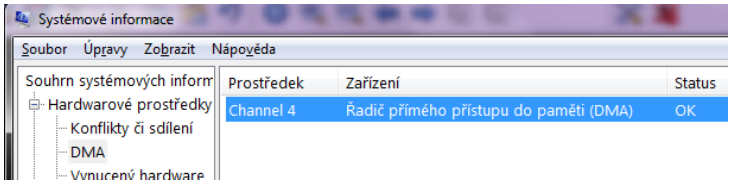
- https://www.electronicshub.org/arm-interrupt-tutorial/
- https://www.geeksforgeeks.org/priority-interrupts-sw-polling-daisy-chaining/
- http://ethv.net/workshops/osdev/notes/notes-3
- https://mcuoneclipse.com/2016/08/14/arm-cortex-m-interrupts-and-freertos-part-1/
- https://www.elprocus.com/types-of-interrupts-in-8051-microcontroller-and-interrupt-programming/
- https://fpgauniversity.intel.com/redirect/materials?id=/pub/Intel_Material/16.1/Tutorials/Using_GIC.pdf
- https://electronics.stackexchange.com/questions/76867/what-do-the-different-interrupts-in-pcie-do-i-referring-to-msi-msi-x-and-intx

5.9.2 DMA

DMA (Direct Memory Access) – jedná se o kanály přímého přístupu do paměti. Používají se pro přenosy většího množství dat mezi komponentami tak, aby jimi nebyl zbytečně zdržován procesor. Princip je takový, že DMA přenos je zahájen procesorem, ale zbytek přenosu zajistí DMA řadič, takže procesor se

v době přenosu může věnovat „jiným záležitostem“. DMA řadič bývá obvykle integrován v jižním mostě čipsetu, případně v SoC čipu.

Tuto technologii využívají obvykle paměťová média nebo grafické, zvukové či síťové karty. Aby mohl být DMA přenos použit, musí být podporován oběma komunikujícími stranami, někdy je třeba ho i aktivovat v BIOSu či rozhraní dané komponenty.



Obrázek 5.28: Seznam DMA kanálů ve Windows 7

Při komunikaci se využívá konkrétní *DMA kanál*, každý kanál má své číslo, podobně jako IRQ (jejich počet je 4 nebo 8, číslované od 0). Kanál 4 je vyhrazen pro DMA řadič. Na starších základních deskách byl kanál 2 vyhrazen pro disketovou mechaniku, to v současné době už není potřeba.

Pokud má být DMA použito například pro přenos dat z pevného disku do operační paměti, řadič disku zaregistruje DMA kanál a vyjedná s DMA řadičem přenos. Ten může probíhat synchronně (proces žádající o přenos si počká na provedení) nebo asynchronně (žádající proces během přenosu provádí jiný kód).

Přiřazení DMA kanálů najdeme ve Windows opět v *Systémových informacích*, v UNIXových systémech v příslušném nástroji s GUI nebo v souboru `/proc/dma`.

```
[sarka@localhost ~]$ cd /proc
[sarka@localhost proc]$ cat dma
4: cascade
```

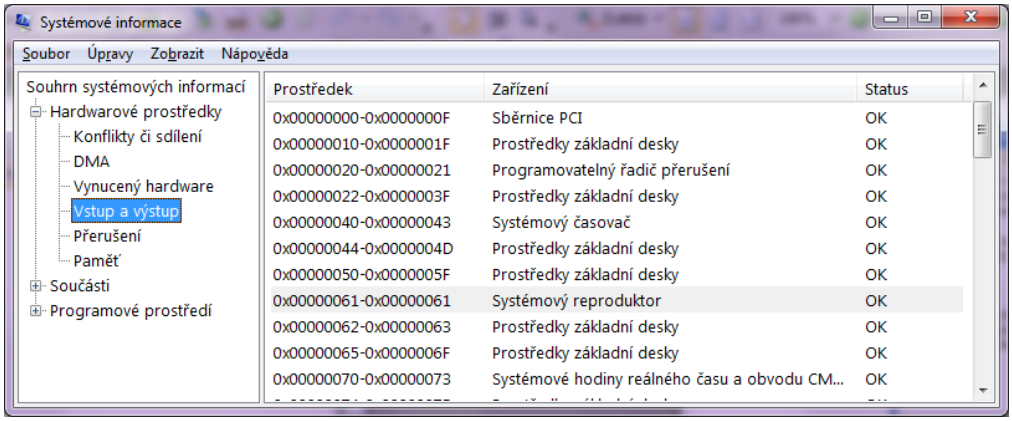
Obrázek 5.29: Výpis se seznamem DMA kanálů v Linuxu (konzola)

Na každém kanálu je sada dvou signálních linek:

1. DRQ (DMA Request) – pro DMA žádost,
2. DACK (DMA Acknowledgement) – pro DMA potvrzení.

5.9.3 I/O adresy a adresy paměti zařízení

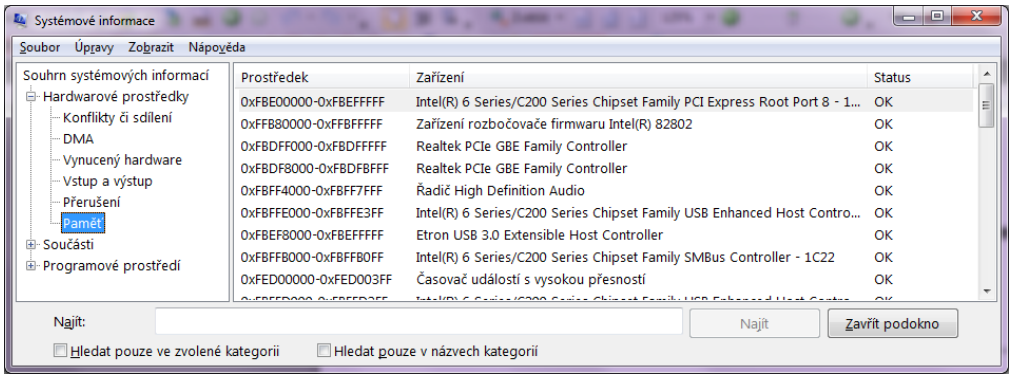
I/O porty (I/O adresy) jsou číselné reprezentace skutečných umístění v paměti, která jsou používána pro nízkoúrovňovou komunikaci s různými zařízeními. Určují adresu, na kterou lze zapsat data určená pro dané zařízení (resp. číst data odeslaná daným zařízením) včetně řídicích signálů. Obvyklá velikost těchto bloků paměti je 16, 32, ... kB, ale ne vždy. Jedno zařízení může mít i více takových adresových bloků, nemusejí nutně následovat za sebou.



Obrázek 5.30: Seznam I/O portů ve Windows 7

Velice často tuto možnost používá grafická karta, zvuková karta, operační paměť, reproduktory, časovač, hodiny reálného času (pozor, to není totéž), sběrnice USB, různé další sběrnice a zařízení na ně připojená,

mechanismus DMA, atd. Seznam I/O adres ve Windows najdeme v Systémových informacích pod položkou Vstup a výstup, jak je vidět na obrázku 5.30.



Obrázek 5.31: Adresy zařízení ve Windows

```
[sarka@localhost ~]$ cd /proc
[sarka@localhost proc]$ cat ioports
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-006f : keyboard
0070-0077 : rtc
0080-008f : dma page reg
00a0-00a1 : pic2
...
```

```
[sarka@localhost ~]$ cd /proc
[sarka@localhost proc]$ cat iomem
00000000-0009fbff : System RAM
00000000-00000000 : Crash Kernel
0009fc00-0009ffff : reserved
000a0000-000bffff : Video RAM area
000c0000-000ccbff : Video ROM
000f0000-000fffff : System ROM
...
d8000000-dfffffff : PCI Bus #1
...
```

Obrázek 5.32: Seznam I/O portů (vlevo) a seznam adres zařízení (vpravo) v Linuxu (část výpisů)

Adresa paměti zařízení (resp. rozsah adres) je paměť fyzicky se nacházející jinde než na modulech operačních pamětí namapovaná do adresového prostoru. Mnoho zařízení má vlastní BIOS, který také musí být přístupný v rámci společného adresového prostoru, aby se k němu mohl dostat operační systém nebo procesy (tj. potřebuje přiřadit adresu). BIOS daného zařízení či komponenty samzřejmě není umístěn na základní desce ani v modulech operační paměti, je v některém čipu přímo na zařízení/komponentě.

Hodně bloků těchto adres je přiřazeno především samotné základní desce (protože ta má nejrozsáhlejší BIOS a možnosti konfigurace), další jsou pro grafickou, zvukovou a síťovou kartu, ostatní adresové bloky pak většinou pro různé sběrnice, které je dále distribuují připojeným zařízením. Všechna jmenovaná zařízení/komponenty na těchto adresách zpřístupňují svůj BIOS či parametry pro konfiguraci.

Poznámka

Takže jaký je rozdíl mezi I/O adresami a adresami paměti zařízení? I/O adresy slouží čistě pro data – je to nástroj pro nízkoúrovňové přenosy dat. Adresy paměti zařízení naproti tomu slouží pro nízkoúrovňovou práci s konfigurací zařízení/komponenty (tedy pro přístup k funkcím) a využívají je například i diagnostické nástroje.

Seznam I/O portů (I/O adres) najdeme na podobných místech jako předchozí probírané položky, v textovém rozhraní UNIXových systémů jde o soubor `/proc/iomem`.

Kapitola 6

Vnitřní paměti

Rychlý náhled: V této kapitole se budeme zabývat vnitřními paměťmi – jejich typy a vlastnostmi. Zaměříme se na mechanismus, na kterém pracují, různá provedení (pouzdra, moduly), se kterými se můžeme setkat, možnosti jejich testování a ovlivnění jejich vlastností.

Klíčová slova: Vnitřní paměť, vnější paměť, energeticky závislá (volatilní) a energeticky nezávislá (nevolatilní) paměť, dynamická a statická paměť, destruktivní při čtení, paměťová buňka, dekodér, ROM, PROM, EPROM, EEPROM, Flash PROM, RAM, SRAM, DRAM, CMOS RAM, FPM RAM, SDRAM, DDR SDRAM, DDR2, DDR3, DDR3L, DDR4, DDR4L, DDR5, modul, SIMM, DIMM, SPD čip, profily XMP, slot, bank, Dual Channel, Triple Channel, Quad Channel, latence.

Cíle studia: Po prostudování této kapitoly budete vědět, jak v základu fungují vnitřní paměti, získáte přehled o různých druzích těchto pamětí a jejich vlastnostech a využití, budete umět otestovat operační paměť.

6.1 Nejdřív obecně o pamětech

Rozlišujeme paměti vnitřní (memory) a vnější (storage, úložiště). Vnější paměti slouží k dlouhodobějšímu ukládání dat, jsou to například pevné disky, SSD, optické disky, USB flash disky, atd. Slovo „vnější“ neznamená, že nutně mají být vně počítače (ostatně – pevný disk většinou bývá uvnitř), ale většinou to znamená, že není až takový problém je odpojit a případně připojit k jinému počítači (i pevný disk můžeme vyndat a zapojit do jiného počítače).

Vnitřní paměti se tímto způsobem nepoužívají, třebaže by to teoreticky nebylo úplně vyloučeno. Jsou zapojeny do slotu na základní desce nebo integrovány v některém čipu (případně přímo v procesoru).

Základní charakteristiky, které nás u pamětí mohou zajímat:

- kapacita – kolik dat je možné do paměti uložit,
- přístupová doba – doba čekání na splnění požadavku (rozlišujeme pro čtení/zápis),
- přenosová rychlost – kolik dat je možné přečíst (zapsat) za danou časovou jednotku,
- spolehlivost – střední doba mezi dvěma poruchami,
- cena/bit, případně cena/byte.

K uloženým datům na vnitřních i vnějších pamětech lze přistupovat dvěma způsoby:

- *sekvenční* přístup – data v paměti jsou zpřístupňována lineárně, nelze je „přeskakovat“, typické pro některé vnější paměti,
- *přímý* přístup – data lze číst přímo z dané adresy bez ohledu na předchozí, což je typické pro polovodičové (elektronické) paměti.

Poznámka

Sekvenčně tedy čteme například ze zálohovacích pásek a určitá forma sekvenčnosti je také u klasických pevných disků (obojí jsou vnější paměti), i když u pevných disků převládají vlastnosti přímého přístupu do paměti. Proč? Čtecí hlava se sice může volně pohybovat poměrně rychle a téměř přímo (po určité dráze) mezi různými místy na plotně disku, ale rychlost, s jakou se dostane na zadané místo, je ovlivněna jejím původním umístěním. Používají se techniky, které mají pohyb čtecích/zápisových hlav optimalizovat, kromě jiného i přeuspořádáním dat, která jsou čtena (pokud má být přečtena zadaná sekvence dat, hlava je nečte v daném pořadí, ale v tom pořadí, jaké má „po cestě“ – NCQ).

6.2 Vlastnosti vnitřních pamětí

6.2.1 Členění

Paměti také členíme podle *energetické závislosti*:

- *energeticky nezávislé* (nevolatilní) – obsah paměti se uchovává i po odpojení od zdroje,
- *energeticky závislé* (volatilní) – obsah paměti se po odpojení od zdroje energie ztratí, dále členíme podle doby uchování dat:
  - statické – data jsou v paměti uchovávána po celou dobu připojení ke zdroji, není třeba je obnovovat,
  - dynamické – uložená data se ztrácejí i v době, kdy je paměť připojena ke zdroji, je nutné je periodicky obnovovat.

Typický příklad energeticky závislých pamětí jsou moduly operačních pamětí.

Zajímá nás také *destruktivnost při čtení:*

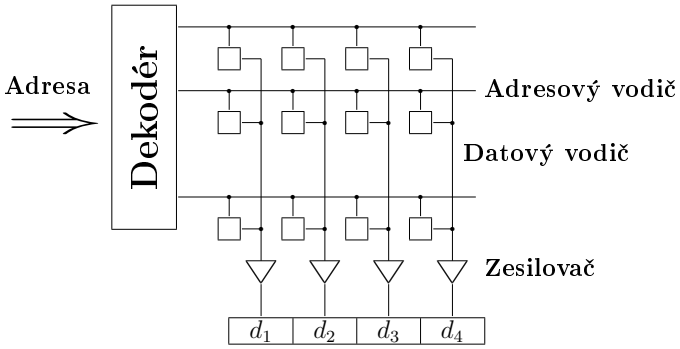
- nedestruktivní při čtení,
- destruktivní při čtení – proces čtení dat způsobí smazání těchto dat, tedy přečtená data musí být znovu zapsána.

Paměti destruktivní při čtení najdeme jak u starších typů vnitřních pamětí (bublinkové paměti), tak i u novějších typů (DRAM a její potomci, což dnes znamená operační paměti).

6.2.2 Obvyklá struktura

Vnitřní paměti se skládají z paměťových buněk. *Paměťová buňka* je místo pro uložení jednoho bitu, tedy uchovává hodnotu 0 nebo 1.

Dále obsahují *dekodér*, který provádí překlad mezi adresou v paměti (tu zná operační systém) a fyzickým umístěním dat z této adresy v paměťových modulech, tedy určuje, se kterými paměťovými buňkami se vlastně má pracovat.



Obrázek 6.1: Schéma vnitřních pamětí

Mezi dekodérem a paměťovými buňkami vedou vodiče pro adresy a data. *Adresové vodiče* určují, se kterými paměťovými buňkami se pracuje (na tom, který má být použit při dané operaci čtení/zápis, je nastavena 1), *datové vodiče* slouží k přenosu dat:

- *čtení z paměti:* k datovým vodičům jde signál od adresového vodiče přes paměťové buňky; pokud 1 projde k datovému vodiči, je na výstupu 1 a naopak,
- *zápis do paměti:* nastaví se bity b_x na hodnoty, které mají být uloženy, pak přes ně jde/nejde signál po datových vodičích k paměťovým buňkám.

Schéma vidíme na obrázku 6.1.

6.2.3 ROM paměti

Zkratka ROM sice znamená „Read Only Memory“ (pouze pro čtení), ale ve skutečnosti se z původních nezapisovatelných druhů časem vyvinuly paměti podobných vlastností a struktury, ale již zapisovatelné.

Velmi důležitou vlastností ROM pamětí je *energetická nezávislost* – pokud je přerušeno napájení (vypneme počítač, dojde ke zkratu nebo výpadku proudu apod.), obsah paměti není smazán.

ROM (Read Only Memory) je nejstarší z pamětí uvedených v této sekci. Je určena pouze pro čtení (permanentní), data jsou zapsána přímo při výrobě a nelze je změnit. Jedná se o kolmou síť vodivých cest (ve více vrstvách) propojených obvykle pojistkami či tranzistory. Tranzistor je buď otevřen (logická 1, „přepálená“ pojistka) nebo uzavřen (logická 0).

ROM paměti se používaly jako permanentní paměť v nejrůznějších zařízeních, například pro uložení kódu BIOSu. Jejich výhodou je velmi nízká cena a vysoká hustota zapsaných dat (díky tomu, že nemohou být přepisována a zapisují se již při výrobě čipu).

PROM (Programmable ROM) je další generace ROM pamětí. Jak název napovídá, je programovatelná, tedy jednou je možné provést zápis (ve speciálním zařízení – tedy neprovádí se při výrobě v továrně), pak již jen čtení, tedy po prvním zápisu funguje jako ROM paměť.

Při výrobě jsou všechny paměťové buňky nastaveny na 1, oproti ROM je mezi diodou a datovým vodičem pojistka, která je při zápisu hodnoty 0 přepálí, čímž je zajištěna možnost odlišení hodnot 0 a 1. Stejně jako ROM, i PROM uchovává data prakticky neomezeně dlouho.

EPROM (Eraseable PROM) lze kdykoliv vymazat (čip je obvykle nutné vyjmout z desky) a znovu zapsat. Je realizována pomocí unipolárních tranzistorů a dokáže uchovat data i několik desítek let (udává se 10 až 20 let).

Vymazání provádíme *UV zářením* (asi půl hodiny přes malé okénko v pouzdře obvodu, pro ochranu před náhodným vymazáním bývá přelepeno), v slunečném dni někdy stačí vystavit čip na dostatečně dlouhou

dobu přímému slunci. Mazání lze provádět i opakovaně, ale tím se zhoršují vlastnosti těchto pamětí.

Po vymazání jsou všechny paměťové buňky vybité (to zde znamená logickou hodnotu 1) a lze zapisovat (měnit jejich hodnotu na 0) pomocí silnějšího elektrického impulsu.

EEPROM (Electrically Eraseable PROM) na rozdíl od předchozí je smazatelná elektrickým impulsem (navíc téměř okamžitě). U starších čipů bylo nutné při mazání použít vyšší napětí, později byl čip vybaven nábojovou pumpou, tedy dokázal sám napětí zvýšit.

Flash PROM (mžiková paměť) je vylepšená EEPROM. Rozdíl oproti jejímu předchůdci – EEPROM – je v tom, že mazání i následný zápis se provádějí *po blocích* (nemusí se mazat celý čip). Mazáním a zápisem se paměťové buňky bohužel postupně ničí (navíc je vždy takto ovlivněn celý blok paměťových buněk najednou), proto je množství těchto operací limitováno, u kvalitnějších čipů jde o desítky až stovky tisíc zápisů a mazání (u méně kvalitních kolem tisíce mazání).

S flash technologií se setkáváme jak u vnitřních (Flash PROM), tak i u vnějších pamětí (například USB flash disk nebo SSD). Zatímco u vnitřních pamětí je obvykle použita technologie NOR, u vnějších pamětí technologie NAND.

Poznámka

Flash PROM čipy se dnes používají například pro čipy s kódem BIOSu. Je to především z toho důvodu, aby bylo možné provádět upgrade BIOSu, je tedy výhodné, když je kód BIOSu v zapisovatelné permanentní paměti.

Další informace

Podrobnosti o nevolatilních pamětech (celý seriál o pamětech) najdeme například na webu http://www.root.cz/clanky/nevolatilni-pameti/.

6.2.4 RAM paměti

RAM je zkratka z „Random Access Momory“, tedy „Paměť s náhodným přístupem“. Většinou se jedná o energeticky závislé paměti (statické nebo dynamické) umožňující čtení i zápis.

Existují dva základní druhy RAM pamětí:

- *dynamická (DRAM)* – hlavními elektronickými součástkami jsou zde jednoduché (a levné) kondenzátory, které mají tendenci se vybíjet i v případě, že je paměť neustále pod proudem, proto se musí pravidelně obnovovat (provádí se refresh),
- *statická (SRAM)* – data se uchovávají po celou dobu připojení k napájení bez nutnosti pravidelné obnovy (proto statická), ale po odpojení napájení se vymaže (proto volatilní).

Výhodou SRAM je především velká rychlost (velmi nízké přístupové doby), ovšem její výroba je složitější a dražší (vzpomeňte si na paměťovou pyramidu). Obvykle je používána pro *cache* (vyrovnávací paměti), protože právě zde potřebujeme větší rychlost, i za cenu dražší technologie.

Výhodou dynamických pamětí je zejména nižší cena, proto se používají například pro operační paměť. Nevýhodou je tedy nižší propustnost a nutnost pravidelného refreshe (obnovujeme obsah paměťových buněk). Refresh provádí řadič paměti. Při čtení dat je vždy vybit příslušný kondenzátor, tj. jde o *destruktivní čtení*, po přečtení je nutný opětovný zápis buňky (což je jedním z důvodů horší přístupové doby).

SRAM je realizována pomocí bistabilních klopných obvodů (tj. obvod, který se nachází vždy v jednom ze dvou určených stavů – dva spojené tranzistory), stavy jsou interpretovány jako 0 nebo 1. Používají se dva datové vodiče – jeden pro zápis a jeden pro čtení (na tom je výstup opačný než v paměťových buňkách, musí se negovat).

Dále se podíváme na různé druhy pamětí – jak šel čas, tedy od starších druhů po momentálně používané.

CMOS RAM (Complementary Metal Oxide Silicon RAM) je vyrobena technologií CMOS, která především znamená velmi malou spotřebu. Stejně jako jiné RAM je volatilní, a tedy pro zachování svého obsahu vyžaduje neustálý přísun energie, ovšem jen malý, proto vystačí napájení z baterie (na několik let až několik desítek let).

Jedná se vlastně o SRAM vyrobenou CMOS technologií. Výrobní technologie CMOS znamená velmi malý příkon, přičemž je napájena z baterie umístěné na základní desce.

Obvykle se používá pro uložení konfigurace počítače (k ní se dostaneme přes BIOS Setup) a pro obvod hodin reálného času. Proto tyto čipy většinou najdeme na základní desce v blízkosti baterie.

FPM RAM (Fast Page Mode RAM) je nástupce DRAM, kde buňky jsou v matici (page), adresuje se zvlášť řádek (signál RAS) a sloupec (signál CAS). Taktéž se jedná o dynamickou volatilní paměť.

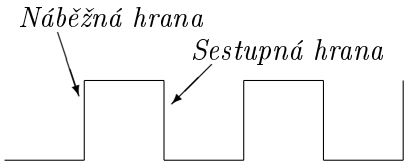
Přístupové doby jsou nižší než u původní DRAM. Využívá se předpokladu, že následující data se budou číst na stejném řádku, tedy jedna z adres se mění s menší pravdobností.

FPM RAM se sama o sobě již dnes nepoužívá, ale pro nás důležitá především proto, že se s ní objevily technologie a metody, které se u pamětí používají dodnes (například způsob adresace).

EDO RAM (Extended Data Output RAM) je vylepšená FPM RAM s rychlejším obnovováním dat – každé 2 hodinové cykly (u FPM 3 cykly). Umožňuje delší přidržení dat na výstupu, tedy je lepší možnost synchronizace s procesorem a je možné překrývání čtecích impulsů (tj. během čtení obsahu jedné adresy se připravuje další adresa). Důsledkem těchto změn je zrychlení přístupu až o 20 %. Ve variantě *BEDO RAM (Burst EDO RAM)* jsou navíc data čtena po dávkách.

SDRAM (Synchronous Dynamic RAM, SDR) je již předzvěstí dnes používaných pamětí. Je to dynamická volatilní paměť, a synchronní se nazývá proto, že pracuje synchronizovaně se systémovými hodinami (stejný takt jako paměťová sběrnice, během jednoho taktu provede právě jednu operaci). Přenáší data pouze po náběžné hraně řídicího signálu systémového časovače.

Přístupové doby závisejí na rychlosti sběrnice, mohou být i pod 10 ns (obvykle 8, 10 nebo 12 ns).



Obrázek 6.2: Náběžná a sestupná hrana řídicího signálu (směr zleva)

Poznámka

Poprvé se objevuje značení podle organizace *JEDEC* (Join Electron Device Engeneering Council): PCxxx, kde xxx je *frekvence* v MHz (například PC100, PC133).

Od frekvence se samozřejmě odvíjí také přenosová rychlost. Ta se vypočítá podle vzorce

$$\text{rychlost} = \text{frekvence} \times \text{šířka sběrnice}$$

Např. při frekvenci 133 MHz a šířce sběrnice 32 bitů je $133 \times 32 = 4\,256\,\text{Mb/s} = 532\,\text{MB/s}$.

SDRAM dnes najdeme jen na starých počítačích, zvláště u základních desek pro procesory AMD byla

velmi brzy nahrazena následujícím typem pamětí.

DDR SDRAM (Double Data Rating SDRAM) je konstrukčně podobná SDRAM, ale na rozdíl od ní je schopná přenášet data na náběžné i sestupné hraně signálu generovaného systémovým časovačem, čímž zdvojnásobuje datovou propustnost (pouze data, nikoliv ostatní signály), během jednoho taktu sběrnice provede dvě operace.

Rychlosti a označení jsou opět standardizovány mezinárodní organizací *JEDEC*.

Tyto paměti se označují PCyyyy, kde yyyy je *přenosová rychlost* v MB/s (například PC2100, PC3200, apod.). Také se běžně používá označení DDRxxx, kde xxx je dvojnásobek frekvence paměťové sběrnice (např. DDR400 jede na frekvenci 200 MHz), taky hovoříme o jednotce MT/s (Megatransferů za sekundu) – u pamětí DDR je poměr frekvence a T/s dvojnásobný, tedy například DDR400 mají propustnost 400 MT/s a paměťová sběrnice komunikuje na frekvenci 200 MHz. První z těchto způsobů označení se vztahuje k typu paměťového modulu (těm se budeme věnovat v následující sekci), druhý k standardu DDR.

Například: PC1600 (DDR200), PC2100 (DDR266), PC3200 (DDR400).

Poznámka

Všimněte si, že číslo v označení PCyyyy je osminásobek čísla u označení DDRxxx. Z toho vyplývá, že u pamětí DDR je jeden T/s (transfer za sekundu) roven osmi B/s, tj. najednou se přenese 8 B. Takže od varianty pamětí DDR si budeme pamatovat, že existují dva druhy označení:

- PCyyyy, kde yyyy je přenosová rychlost v MB/s,
- DDRxxx, kde xxx je přenosová kapacita v MT/s.

V některých případech nemusí násobnost 8× mezi těmito dvěma hodnotami úplně sedět – záleží i na jiných parametrech, které se mohou mírně „hýbat“. Navíc se zaokrouhluje.

DDR2 jsou novější generace s vyšší frekvencí, dnes jsou tyto paměti nejpoužívanější. Oproti DDR SDRAM mají vnitřní frekvenci poloviční oproti vnější vedoucí ke sběrnici (neboli vnější sběrnice je taktována na dvojnásobnou frekvenci než vnitřní) při nižším napětí, tj. při stejné frekvenci jako DDR se méně zahřívají. Dvojnásobná rychlost zmíněná u DDR platí nejen data, ale také pro všechny signály.

Podle konsorcia JEDEC používáme označení PC2-yyyy, kde yyyy je přenosová *rychlost* v MB/s stejně jako u původních DDR. Setkáváme se také s označením DDR2-xxx, kde xxx je dvojnásobek taktu paměťové sběrnice, resp. T/s.

Například: PC2-4200 (DDR2-533), PC2-6400 (DDR2-800), PC2-8500 (DDR2-1066).

DDR3 je další generace, musí být podporována čipsetem (tj. základní deskou) a procesorem. Má vlastnosti podobné jako DDR2, navíc potřebuje ještě nižší napětí pro stejné frekvence (napětí je obvykle 1,5 V). Označují se PC3-yyyy, kde yyyy je přenosová rychlost v MB/s, resp. DDR3-xxxx, kde xxxx je kapacita v MT/s. Například: PC3-6400 (DDR3-800), PC3-8500 (DDR3-1066), PC3-12800 (DDR3-1600).

Poznámka

Od verze DDR3 může být ve specifikaci přidáno písmeno „L“, například DDR3L. Je to zkratka z „Low Voltage“, tedy se jedná o paměti pracující na nižší než běžné frekvenci – u DDR3L jde o 1,35 V místo běžných 1,5 V. Tyto paměti jsou užitečné v zařízeních běžících alespoň občas na baterii (mají nižší spotřebu), obvykle jsou níže taktovány, jsou celkově pomalejší. Z toho vyplývá, že hlavním účelem „L“ pamětí je snížit energetickou spotřebu zařízení.

Označení obsahuje totéž písmeno, například PC3L-8500 (DDR3L-1066).

Musíme si dát pozor na to, že tyto moduly se nedají kombinovat s moduly pracujícími na běžné frekvenci, taky je dobré zkontrolovat si, jestli si s nimi poradí základní deska a procesor (paměťový řadič, který je dnes obvykle integrován v procesoru).

Existuje taky ještě úspornější varianta označovaná písmenem „U“ (Ultra Low Voltage) pracující při napětí 1,25 V. Písmeno „U“ se pak také vyskytuje v označení paměti.

DDR4 je momentálně nejnovější generace dynamických pamětí. U DDR4 pamětí se mění výrobní proces (na 30nm litografii, postupně se má přejít na 20nm litografii), snížilo se provozní napětí (na 1,2 V), zvýšily se pracovní frekvence (v naprosté většině případů nad 2000 MHz) a tím i přenosové rychlosti, negativní změnou je zvýšení časování na CL12 (budeme probírat později). Je zajímavé, že díky těmto změnám se snížila spotřeba energie až o 40 %.

Používá se způsob označování DDR4-xxxx (s přenosovou kapacitou), například DDR4-2400.

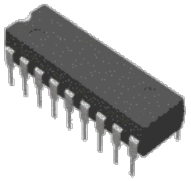
Varianta DDR4L obvykle běží na nižším provozním napětí, jen 1,05 V, a naopak se můžeme setkat s trochu vyššími frekvencemi. Použití je podobné jako u DDR3L – potřebujeme dostatečný výkon při nižší spotřebě a stabilním chodu, čehož se právě dociluje nižším provozním napětím.

DDR5 se začaly objevovat v průběhu roku 2022. Jako první si s nimi začaly rozumět procesory Intel Alder Lake, následně procesory AMD rodiny Zen 4 (tj. 7. generace procesorů Ryzen). Jako obvykle se setkáváme s vyšší frekvencí (vpodstatě tam, kde DDR4 končí, tam DDR5 sotva začínají) i kapacitou (jeden modul může jít až na 128 GB), pracuje se na nižším napětí (1,1 V, ale může být vyšší: 1,25 V), ovšem latence není zrovna ideální. Cena se může zdát také neúměrně vyšší – je ovšem způsobena nejen tím, že jde o novinku, ale i tím, že část elektroniky, která byla původně na základní desce, se přesunula přímo na modul (regulátor napětí a související elektronika). Do základní výbavy se také přesouvají některé funkce, které u dřívější generace byly jen volitelné (především ECC, viz dále).

Používá se označení PC5-propustnost v MB/s, například PC5-44800 (zrovna tento modul pracuje na frekvenci 5600 MHz).

Existují také varianty LPDDR3, LPDDR4, LPDDR5 (Low Power...). Zatímco DDRxL jsou určeny pro úsporný chod desktopů, serverů a embedded zařízení, LPDDRx jsou obdobou pro mobilní zařízení. V každém případě jsou dražší než moduly DDRx. LPDDR je standard nezávislý na standardu DDR, a to včetně doby uvedení, například moduly LPDDR5 byly na trh uvedeny dřív než moduly DDR5.

6.3 Paměťové moduly



Obrázek 6.3: Paměť v DIP pouzdře

Zatím jsme se bavili o typech pamětí (dnes DDR2, DDR3, DDR4). V obchodech však kupujeme nikoliv tyto čipy, ale paměťové moduly, na kterých je sdruženo vždy několik takových čipů.

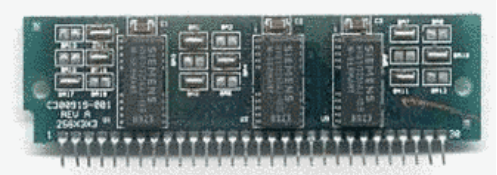
Paměti v DIP pouzdře – DIP (Dual In-Line Package) je běžný integrovaný obvod SRAM nebo DRAM pamětí v DIP pouzdře, šířka přenosu 1 nebo 4 bity.

Dnes se s nimi už prakticky nesetkáme, používaly se u počítačů do i286. Kapacita těchto čipů byla malá, a ani přístupová doba nebyla ideální. Hlavní nevýhodou bylo množství místa na základní desce, které tyto čipy zabíraly – takovou desku poznáme podle „matice černých hrobečků“.

Další informace

Zajímavé obrázky z historie najdeme na http://redhill.net.au/b/b-92.html.

SIPP (Single Inline Pin Package) obsahuje paměťové obvody (typu FPM RAM), ale na rozdíl od DIP pouzdra jsou všechny integrované na jedinou desku, která se zasouvá do patice na základní desce. Ovšem vývody (piny) jsou stejné jako u DIP – úzké tenké plíšky, které se často ohýbaly a lámaly.

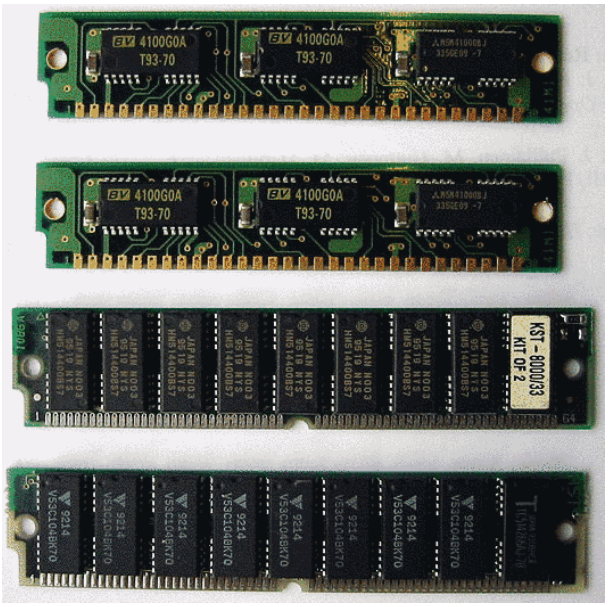


Obrázek 6.4: SIPP pouzdra[1]

Přínosem byla tedy integrace na jednu desku (paměťový modul), ale vzhledem k výše uvedené nevýhodě se moc neujaly.

SIMM (Single Inline Memory Module) zachovaly koncepci paměťového modulu zasouvaného do slotu (podobně jako SIPP), ale již praktičtější kontakty (plošky na části modulu zasouvané do slotu), kterážto základní koncepce se používá dodnes.

SIMM moduly byly ve dvou délkách – 30pinové nebo 72pinové. 30pinové obsahovaly obvody FPM RAM, 72pinové pak obvody EDO RAM.



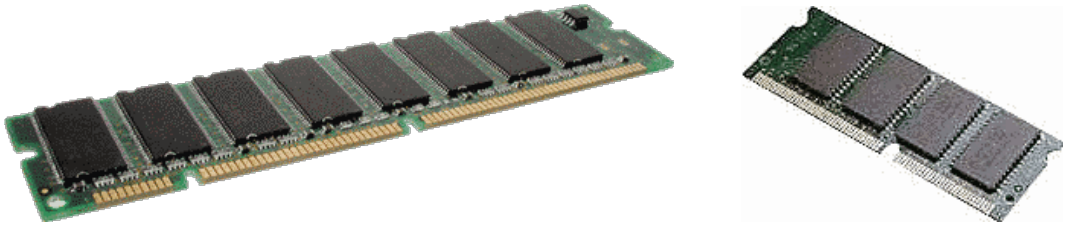
Obrázek 6.5: SIMM moduly (nahoře 30pinové, dole 72pinové)

DIMM (Dual In-line Memory Module) jsou dnes běžně používané. Bývají osazeny paměťmi EDO RAM, SDRAM, DDR SDRAM, DDR2 SDRAM, DDR3 SDRAM, DDR4 SDRAM, atd. Podle toho také mohou být moduly označeny, například DDR II DIMM, DDR III DIMM, DIMM DDR4. Na jednom modulu mohou být buď 4 čipy (nejlevnější), nebo 8 čipů anebo dokonce 16 čipů (v tom případě jsou na obou stranách modulu, na každé straně 8). Pokud je na modulu připevněn chladič (tj. celý modul je zapouzdřen do chladiče), pak jednotlivé čipy samozřejmě neuvidíme.

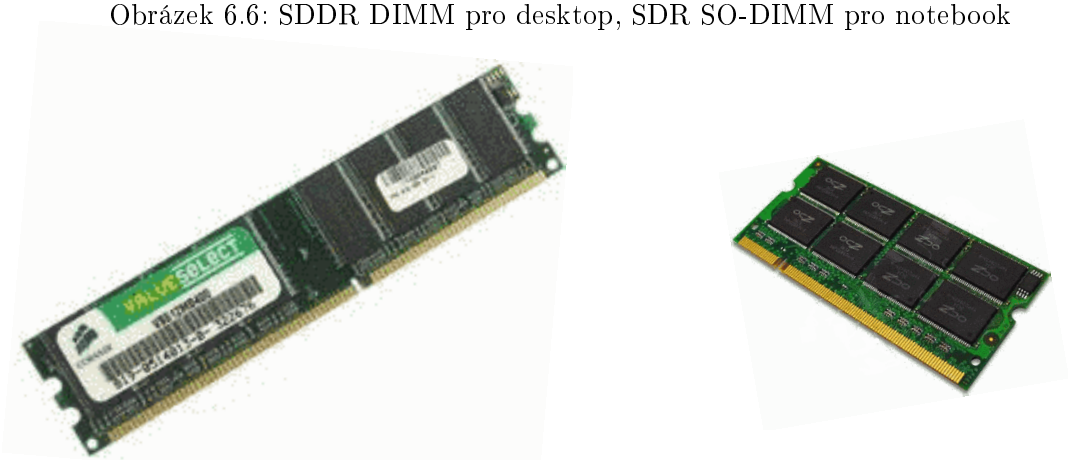
Existují také odlišné varianty pro notebooky (notebookové moduly – SO-DIMM, Small Outline DIMM – jsou výrazně kratší).

DIMM moduly s různými typy pamětí se samozřejmě neliší jen označením, ale také dalšími vlastnostmi. Především mají různý počet pinů (například DDR DIMM má 184 pinů, DDR II DIMM má 240 pinů), jinou šířku a také odlišné klíčování (výřezy na spodní části modulu, která se zasouvá do slotu), a tedy obvykle není možné zasunout modul do slotu, do kterého nepatří. Další odlišnost je v napájecím napětí, ale o tom jsme se bavili už v předchozím textu.

[1] Zdroj: http://www.hd.cz/rady/jak-vybirat-operacni-pamet.php



Obrázek 6.6: SDDR DIMM pro desktop, SDR SO-DIMM pro notebook



Obrázek 6.7: DDR DIMM pro desktop, DDR SO-DIMM pro notebook

Notebookové moduly mohou být zasunuty do slotu podobně jako u desktopů, ale můžeme se setkat také s napevno připájenými moduly z důvodu ušetření místa.

Když kupujeme paměťový modul, musíme si zjistit, který typ podporuje základní deska.

Struktura DIMM modulu je následující:

- deska, piny,
- paměťové čipy (obvykle 8 nebo 16),
- SPD čip.

SPD čip (Serial Presence Detect) obsahuje informace o paměti uložené výrobcem (výrobce, datum výroby, pracovní frekvence, časování, pracovní napětí, latence pro určitou pracovní frekvenci paměti, atd.) o délce 128 B.

SPD čip je obvykle permanentní zapisovatelná paměť (většinou Flash PROM, dříve EEPROM) zasazená v paměťovém modulu. Jedná se o jeden ze standardů JEDEC.

XMP (eXtreme Memory Profile) je rozšíření SPD o (obvykle) čtyři další profily určení parametrů

Obrázek 6.8: DDR III DIMM (desktop a notebook)

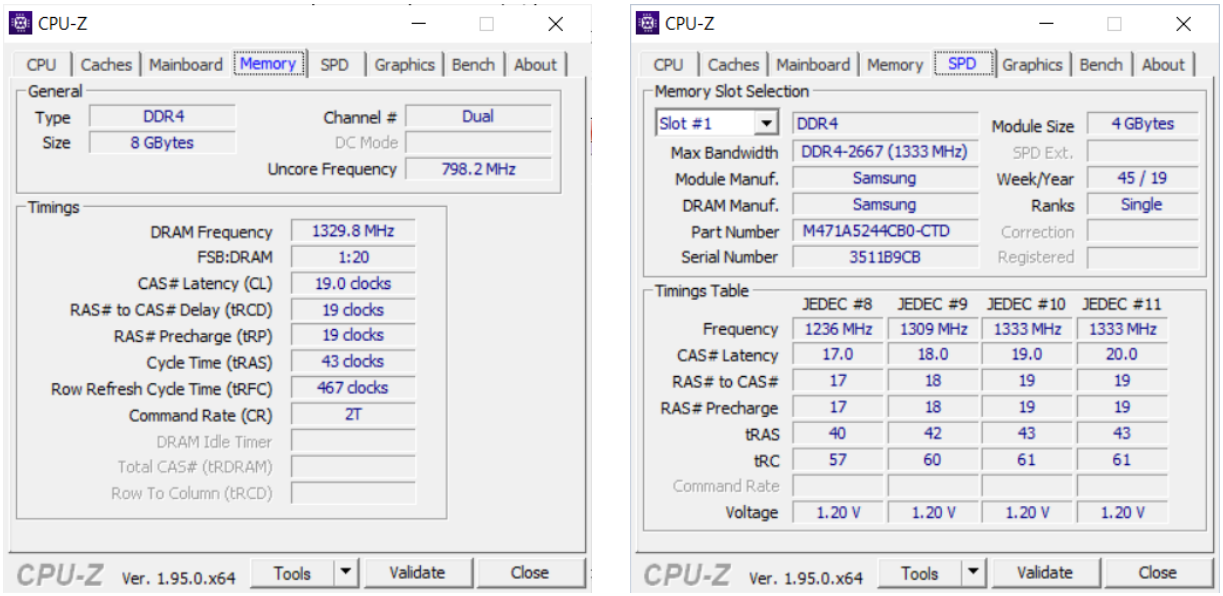
pamětí (nastavení frekvence, časování a napětí). XMP se používají pro snadnější přetaktování pamětí, stačí zvolit jeden z předdefinovaných profilů. Jsou typické především pro základní desky s procesory Intel, ale bývají podporovány také na většině desek s procesorem od AMD.

AMD EXPO (Extended Profiles for Overclocking) je obdoba XMP od AMD. Zatím nejsou tolik rozšířené jako XMP, je to mnohem novější specifikace.

V dokumentaci k základní desce obvykle najdeme informaci o tom, které přetaktovací profily jsou podporovány. Nezapomeňte, že paměťový řadič (který musí být schopen s profily pracovat) se dnes obvykle nachází přímo v procesoru.

Postup

Na obrázku 6.9 vidíme informace, které nám může poskytnout program CPU-Z. Na kartě *Memory* jsou základní informace o operační paměti (typ DDR4, 8 GB celkem) včetně různých parametrů. Na kartě *SPD* pak najdeme obsah SPD čipu včetně předdefinovaných profilů. V tomto případě se jedná o DDR4-2667 (modul pracuje na 1333 MHz, což je zhruba dvojnásobek taktu paměťové sběrnice) a jsou definovány čtyři profily.



Obrázek 6.9: Program CPU-Z, karty *Memory* a *SPD*

Na DIMM modulu je tedy více paměťových čipů. Jejich vzájemné propojení se liší u různých generací použitých pamětí. Zatímco u DDR a DDR2 je víceméně zachována větvená struktura modulů, u DDR3 jsou čipy uspořádány na jediné komunikační sběrnici přes celý modul. Další verze pamětí opět přinášejí změny v komunikační struktuře.

6.4 Jak fungují moduly DIMM

Jak bylo výše uvedeno, paměť v DIMM modulech je organizována v mřížce adresových vodičů, v průsečících řádků a sloupců mřížky jsou paměťové buňky.

Slot pro DIMM moduly bývá opatřen záklopkami (zámky) pro upevnění modulu. Každý slot je připojen na *paměťovou sběrnici* (Memory Bus) vedoucí k paměťovému řadiči (který je dnes obvykle v procesoru).

Frekvence paměťové sběrnice nám zároveň dává limit pro frekvenci, na které budou pracovat paměťové moduly.

Bank je souhrn paměťových slotů, které jsou společně hardwarově adresovány (2D adresa ve smyslu řádek a sloupec), například 4 sloty mohou být ve 2 bancích. Některý z banků obsahujících paměťový modul na stejné sběrnici je *aktivní*, tj. lze s ním právě pracovat.

6.4.1 Komunikační kanály

Můžeme se setkat s pojmem *Dual channel*, *Triple channel*, případně *Quad channel* – dvoukanálový, tříkanálový či čtyřkanálový paměťový řadič. Znamená to, že řadič dokáže komunikovat dvěma paměťovými sběrnicemi najednou (resp. se třemi nebo čtyřmi), tedy šířka paměťové sběrnice na straně řadiče je dvojnásobná (trojnásobná, čtyřnásobná). Jednokanálový řadič paměti se označuje Single channel.

Na každou paměťovou sběrnici (resp. na každou z jejích dvou částí) s podporou Dual channel je napojen jeden bank, takže například pokud máme u Dual Channel celkem čtyři sloty na základní desce, jsou v každém banku dva sloty a s každou dvojicí slotů může paměťový řadič komunikovat zvlášť. Na základní desce bývá většinou barevně vyznačeno, které banky jsou na stejné sběrnici, ale způsob značení musíme vyhledat v dokumentaci – někteří výrobci označují stejnou barvou banky na stejné sběrnici, jiní zase stejnou barvou banky na odlišných sběrnicích.

Poznámka

Proč je to důležité? Pokud chceme využít dva DIMM moduly, měli bychom je zasunout do slotů na různých sběrnicích, aby byly v různých bancích. Kdybychom měli oba moduly na stejné paměťové sběrnici, nebylo by možné číst z obou modulů zároveň (situace by byla podobná jako v případě jediného dvojnásobně velkého modulu v jednom slotu), paměti by měly horší latenci (odezvu, o té se více dozvíme dále).

Také je nutné, aby oba takto použité moduly byly stejného typu (identické včetně frekvence) a opravdu musí být jen pár nebo dva páry (když použijeme lichý počet nebo jinak nedodržíme stanovené podmínky, nebude dual channel funkční). Abychom mohli plně využít vlastnosti Quad channel, museli bychom samozřejmě mít čtyři identické a stejně nakonfigurované moduly.

Nejlepší je pro dual/triple/quad channel nakupovat paměti v kitu (sadě) – například místo jednoho modulu o kapacitě 8 GB lze sehnat kit 2×4 GB.

6.4.2 Operace a časování DIMM

Činnost modulu je dána dobou potřebnou pro jednotlivé dílčí operace, které je nutno provést při čtení/zápisu. Tyto operace jsou především určení banku s příslušným modulem, stanovení řádku a sloupce pro danou paměťovou buňku, obnovení obsahu buňky po čtení (jde o destruktivní čtení), apod.

Příkazy pro moduly DIMM

- RRD (Row Active to Row Active Delay) – vybere se bank, se kterým se pracuje, vybraný bank musí být aktivní (řadič udržuje vždy několik banků aktivních, ostatní jsou vypnuté a je třeba je zaktivnit),
- RAS (Row Address Strobe) – najdi řádek,
- RCD (RAS to CAS Delay) – rozmezí mezi hledáním řádku a sloupce,
- CAS (Column ...) – najdi sloupec na řádku,
- operace read nebo write na dané adrese,

- RP (Row Precharge) – obnovení dat v buňkách.

Pokud se čte ve stejném banku na tomtéž řádku, není nutné znovu provádět RRD, RAS ani RCD.

Doba provedení jednotlivých příkazů

- tRAS – jak dlouho trvá nalezení řádku,
- tRCD (RAS to CAS Delay) – jak dlouho trvá, než je po vyhledání řádku spuštěno hledání sloupce,
- **tCL** (Column Address Strobe Latency, CAS Latency) – jak dlouho trvá, než je nalezen sloupec na řádku,
- tRP – doba obnovení dat v buňkách,
- RFC (Row Refresh Cycle Time) – během obnovování obsahu (Refresh) musí být modul označen jako neaktivní a až po obnovení je označen jako aktivní; *tRFC* je doba od tREF do zaktivnění modulu,
- REF (Refresh) se provádí pravidelně postupně u všech paměťových modulů v intervalech o délce mikrosekund, jde o pravidelné obnovování obsahu (použité kondenzátory se vybíjejí a bez této operace by neudržely uložená data), údaj označujeme *tREF*.

Pokud se čte ve stejném banku na tomtéž řádku (a to je velmi častý případ), není nutné znovu provádět RAS ani RCD. Vzhledem k tomu, že obě tyto operace sice trvají déle než dva cykly, ale při většině přístupu se neprovádějí, jsou pro rychlost modulu určující spíše ostatní operace, především operace CL (výběr sloupce na řádku).

Uvedené hodnoty se používají při značení modulů. Najdeme je někdy už v katalogu (včetně internetových obchodů), ale určitě by měly být uvedeny na balení nebo samotném modulu. Zde hodně záleží na výrobci, které informace uvolní pro prodejce v elektronické formě.

Nejdůležitější údaj je *latence* (časování) – CL (tCL), například CL4. Jestliže jsou uvedeny i další údaje, najdeme je ve formě řetězce tCL-tRCD-tRP-tRAS/CM_RATE.

Příklad

Následují tři možné řetězce údajů:

tCL–tRCD–tRP–tRAS	/	CM_RATE
3 – 3 – 2 – 6	/	1T
8 – 7 – 6 – 19	/	2T
5 – 5 – 5 – 16	/	2T

1. tCL – od okamžiku vyslání signálu CAS k okamžiku, kdy jsou data přístupná, u dat na stejném řádku matice se blíží celkové době zpřístupnění těchto dat,
2. tRCD – od aktivace řádku do vyslání signálu CAS,
3. tRP – od deaktivace předchozího řádku po aktivaci nového,
4. tRAS – od okamžiku vyslání RAS signálu k okamžiku aktivace řádku; nemá velký význam, ale pokud se špatně nastaví, způsobuje nestabilitu paměti,
5. CM_RATE (Command Rate) – od vybrání signálu k vyslání a čipu, kterému je určen (CS – Chip Select) po jeho předání paměťovému čipu (obvykle 1 nebo 2), takto je stanovena prodleva pro komunikaci mezi řadičem paměti a modulem (tento údaj nebývá obvykle u modulů uveden).

S vývojem pamětí pro DIMM moduly se tCL (latence) postupně zvyšuje. Zatímco u DDR byl mezi 2 a 3, u DDR2 se zvyšuje na hodnoty mezi 3 a 6 a DDR3 to je obvykle někde mezi 5–10 (často 9). Jak vidíme, se zvyšujícími se frekvencemi se zhoršuje odezva.

Poznámka

Ovšem už dlouho se vedou spory o skutečném významu co nejnižší latence. Předpokládá se, že potřeba nízké latence je důležitá hlavně ve spolupráci s procesorem s nižší velikostí cache nejvyšší úrovně.

Obecně platí, že při výběru paměťových modulů se vyplácí nejen vybírat podle CL (latence), ale také tak, aby frekvence pamětí a procesoru byly synchronizovány (tj. stejné nebo násobek).

6.4.3 Přetaktování pamětí

Přetaktování se provádí v BIOSU, vpodstatě měníme nastavení stanovená výrobcem v SPD čipu. V BIOSu najdeme potřebné hodnoty buď v samostatné sekci *Memory Timing Setting* nebo v *Advanced Settings* (každý BIOS je trochu jiný).

Každý BIOS také nabízí různé množství údajů, které lze konfigurovat. Většinou určitě najdeme tCL (CAS Latency) – když nastavíme příliš nízkou hodnotu, tak sice zlepšíme odezvu pamětí, ale zvýší se pravděpodobnost chyb (obsah paměťových buněk se „nestihne“ přenést v příliš krátkém časovém okamžiku).

Dále se obvykle setkáme s položkami odpovídajícími zbytku řetězce – tRCD (RAS to CAS Delay), tRP (RAS Precharge), tRAS (RAS, Active to Precharge). U některých BIOSů můžeme také narazit na možnost nastavení *agresivního* nebo *konzervativního* časování.

Položka CM_RATE (také jako CMD) ovlivňuje celkovou rychlost pamětí. Obvykle bývá 1T nebo 2T. Když hodnotu snížíme z 2T na 1T, rychlost pamětí se hodně zvýší, ale mohou se stát nestabilními. Obvykle platí, že pokud máme jen jeden modul, tak hodnota 1T je akceptovatelná, u více modulů na základní desce (zvláště dual/triple/quad channel) je lepší nechat tuto hodnotu na 2T.

Další informace

Zajímavé porovnání DDR, DDR2 a DDR3 (stránka 4 z celkových 9) najdeme na http://www.bit-tech.net/hardware/memory/2007/11/15/the_secrets_of_pc_memory_part_1/4.

Zajímavý článek o časování najdeme na http://pctuning.tyden.cz/navody/upravy-pretaktovani/15013-jak-na-pretaktovani-uvod-a-vysvetleni-zakladnich-pojmu?start=6.

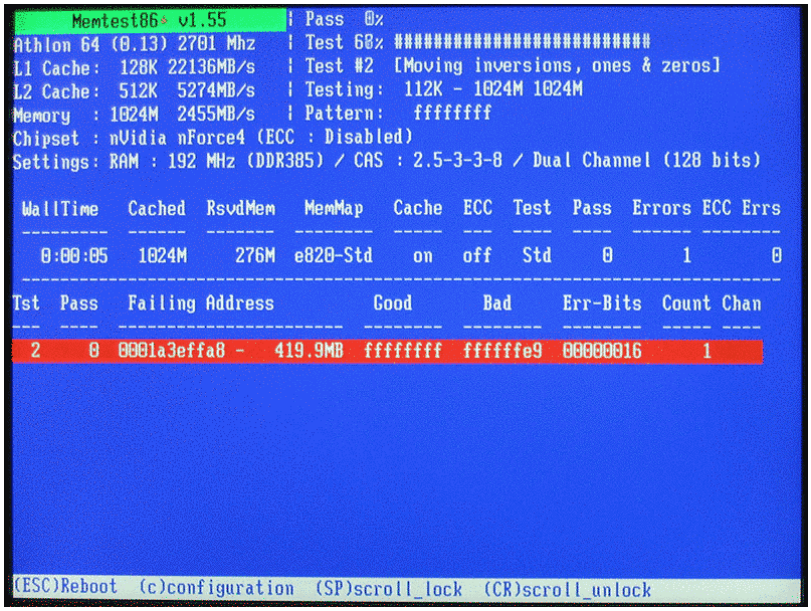
6.5 Testování pamětí

Pokud chceme jen zjistit, jak si stojí naše paměťové moduly z hlediska rychlosti, latence a dalších vlastností, můžeme použít v jednodušším případě nástroje dodávané s operačním systémem (když nám stačí informace o velikosti paměti), ale samozřejmě existují mnohem lepší nástroje. Většinou najdeme jejich volně šiřitelné verze na portálech http://www.slunecnice.cz, http://www.instaluj.cz apod.

K nejznámějším patří

- *SiSoftware Sandra Lite* (http://www.sisoftware.net/, existují varianty volně ke stažení),
- CPU-Z (http://www.cpuid.org je freeware zjišťující informace o procesoru a pamětech,
- HWInfo (http://www.hwinfo.com, varianty HWInfo32 a HWInfo64) je velmi zajímavý freeware s podrobnými výpisy, také existuje jeho portable (přenosná) verze (dá se spustit z USB Flash disku).

Pro bezpečnostní testování pamětí se nejčastěji používá program *MemTest86+* (freeware, dostupný na http://www.memtest86.com). Stáhneme si ISO soubor, který buď vypálíme na CD nebo vytvoříme bootovací USB flash disk (například pomocí nástroje Rufus). Pak z daného média nabootujeme a po chvíli se testy samy spustí (obvykle nemusíme nic předem nastavovat).



Obrázek 6.10: Memtest86+

Způsob testování Chybovost pamětí můžeme testovat buď přímo v operačním systému (spustíme aplikaci například ve Windows), ale tento postup není moc přínosný, protože data jdou přes operační systém a nelze testovat celou paměť. Lepší možností je bootování z výměnného paměťového média (CD nebo USB flash), kde máme nainstalovaný testovací program (stáhneme a vypálíme ISO/uložíme flash disk a nabootujeme. Například program MemTest je dostupný ve dvou formách – odlehčená varianta běží ve Windows, varianta MemTest86+ je bootovací.

Některé operační systémy (typicky Linux) nám v tomto ulehčují práci; v startovací nabídce systému se kromě jednotlivých operačních systémů také obvykle objeví položka MemTest86+ (nebo podobná), po jejím vybrání se provede test paměti bez startu operačního systému.

Pokud některý z výše uvedených programů nahlásí chybu, měli bychom zjistit, co konkrétně chybu způsobuje. Na vině může být vadný hardware (paměťový modul, slot, řadič paměti, případně základní deska či procesor), ale také může jít o špatné nastavení časování pamětí, které způsobuje nestabilitu, nebo nedostatečné napájení (zdroj nestačí). Jestliže tedy je hlášena chyba a máme víc než jeden paměťový modul, přehodíme moduly navzájem, případně můžeme vyzkoušet jiný slot, a spustíme test znovu.

Typy chyb:

- chyba se po přehození modulů objevuje u stejného modulu (při opakování testování ve stejném slotu je adresa chyby stejná, ale mění se při přehození do jiného slotu) – vadný je tento modul,
- chyba se po přehození modulů objevuje na stále stejné adrese – moduly jsou OK, ale slot je vadný nebo zaprášený,
- chyba se objevuje náhodně (v čase i místě) – obvykle je problém v základní desce nebo paměťovém řadiči, málo výkonný zdroj, moduly navzájem nekompatibilní nebo je modul nekompatibilní s deskou.

Kapitola 7

Vnější paměti

Rychlý náhled: Podíváme se na vnější paměti. Budeme se zabývat především pevnými disky, ale také ostatními magnetickými paměťovými médii, magnetooptickými, optickými a flash médii. Součástí kapitoly jsou sekce o zálohování a archivaci, RAID polích a NAS.

Klíčová slova: Sekvenční vs. přímý přístup, lineární a vícerozměrná adresace, geometrie disku, plotna, povrch, osa, čtecí/zápisová hlava, stopa, sektor, cylindr, cluster/blok, advanced format, adresace CHS vs. LBA, parkování hlav, typy zápisu LMR (podélný), PMR (kolmý), SMR (shingle, vrstvený), magnetická reverzace, vystavování hlav, lineární motor, AHCI, NVMe, prokládání, NCQ (Native Command Queuing), prekompenzace zápisu, ZBR (Zone Bit Recording), Stable Track, MTBF, S.M.A.R.T., RPM, RAID, JBOD, NAS, CD, DVD, Blu-Ray, zóny na optickém médiu (vnitřní, lead-in, lead-out, vnější), SSD, SLC, MLC, TLC, QLC, Wear Levelling, hybridní disky (SSHD), SD karta, SDHC, SDXC, zálohování vs. archivace.

Cíle studia: Po prostudování této kapitoly se zorientujete ve světě vnějších pamětí. Rozpoznáte různé druhy paměťových médií, budete znát jejich typické vlastnosti a porozumíte jejich vnitřnímu fungování.

7.1 Základní pojmy

Podle způsobu přístupu k médiu rozlišujeme paměťová média

- *sekvenční* – přístup je prováděn sekvenčně (abychom mohli přečíst konkrétní data, musíme přečíst všechna data předcházející), typické pro zálohovací média (pásky),
- s přímým přístupem – podle adresy, adresa může být
  - *lineární* – jednoduchá lineární adresa, většinou číselná, počítáme obvykle od nuly,
  - *vícerozměrná* – adresa je složena z více částí (podobně jako řádek a sloupec tabulky).

Podle principu čtení/zápisu rozlišujeme

- *papírová média* (děrné štítky, děrné pásky), dnes už nejsou moc používána
- *magnetická a magnetooptická média* – na principu magnetismu (pevný disk, disketa, magnetooptický disk, magnetická páska)
- *optická média* – využívají obvykle vlastností světla s určitou vlnovou délkou (CD, DVD, Blu-ray)
- *elektronická média* – obsahují elektrické obvody, obvykle energeticky nezávislá vnější elektronická paměť (SSD, USB flash disky, karty apod.)

Další informace

Pěkné články o historii paměťových médií najdeme na

- https://www.svethardware.cz/historie-a-soucasnost-datovych-ulozist/23935
- https://www.root.cz/clanky/pametova-media-pouzivana-u-osmibitovych-mikropocitacu/

7.2 Magnetické disky

7.2.1 Diskové paměti

Základem magnetických diskových pamětí je pevný (hard) nebo pružný (floppy) disk, na který je nanesena magnetická vrstva, v ní jsou kromě magnetických částic další složky (ovlivňují také fyzikální vlastnosti disku). Disk umožňuje přímý přístup do paměti (ale má některé prvky sekvenčního přístupu), data jsou organizována na soustředných kružnicích nebo spirále (většinou kružnicích).

Magnetické diskové paměti mají obvykle některý z rozměrů (*Form Factor*), které najdeme v tabulce 7.1. Tyto hodnoty jsou poměrně univerzální a často platí i pro další komponenty, například optické mechaniky do desktopových počítačů obvykle mívají stejný rozměr jako pouzdra $5\frac{1}{4}$" disků.

palce	mm	Použití
8	203	u velmi starých pevných disků a velmi starých disket
$5\frac{1}{4}$	133	dto.
$3\frac{1}{2}$	88	dnes pro pevné disky, dříve i diskety, ZIP apod.
$2\frac{1}{2}$		některé pevné disky, především do notebooků, taky levnější SSD

Tabulka 7.1: Typické průměry (form factor) magnetických diskových pamětí

Každá paměť potřebuje *řadič* (týká se to i vnitřních pamětí), tedy součástku, která fyzicky zajišťuje přístup k paměti a komunikaci mezi pamětí a procesorem nebo čipsetem (podle toho, kam je připojena).

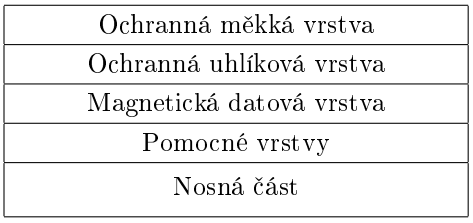
Poznámka

Z magnetických diskových pamětí se dnes setkáváme téměř výhradně s pevnými disky (celá mechanika včetně pouzdra se značí HDD – Hard Disk Drive), proto se jim zde budeme věnovat nejvíce.

7.2.2 Fyzická struktura a geometrie disku

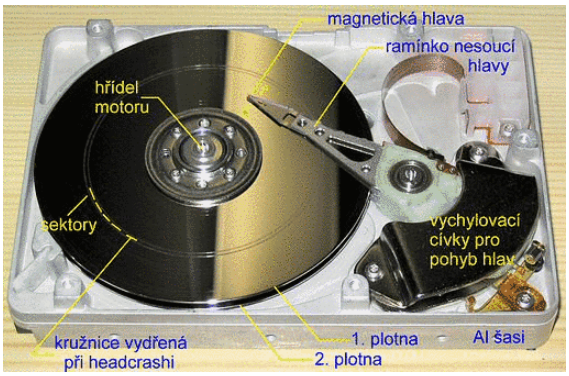
Nosná část disku je obvykle vyrobena z materiálu, který není magnetický (kombinace keramiky, skla, plastu apod.). Starší disky měly tento nemagnetický základ potažený vrstvou oxidu železa, novější mají složitější strukturu.

Na vrstvě pro data je vrstvička uhlíku s některými vlastnostmi diamantu (velmi tvrdá). Na uhlíkové vrstvě je měkčí hladká vrstva chránící hlavu při dopadu. Fyzickou strukturu pevného disku vidíme na obrázku 7.1.



Obrázek 7.1: Povrch pevného disku

Především je důležité, aby disk byl dostatečně tvrdý pro odolnost proti chybám, ale zároveň měkčí, aby při pádu hlavy tuto hlavu nepoškodil. Tepelná roztažnost sousedících vrstev se nesmí příliš lišit.

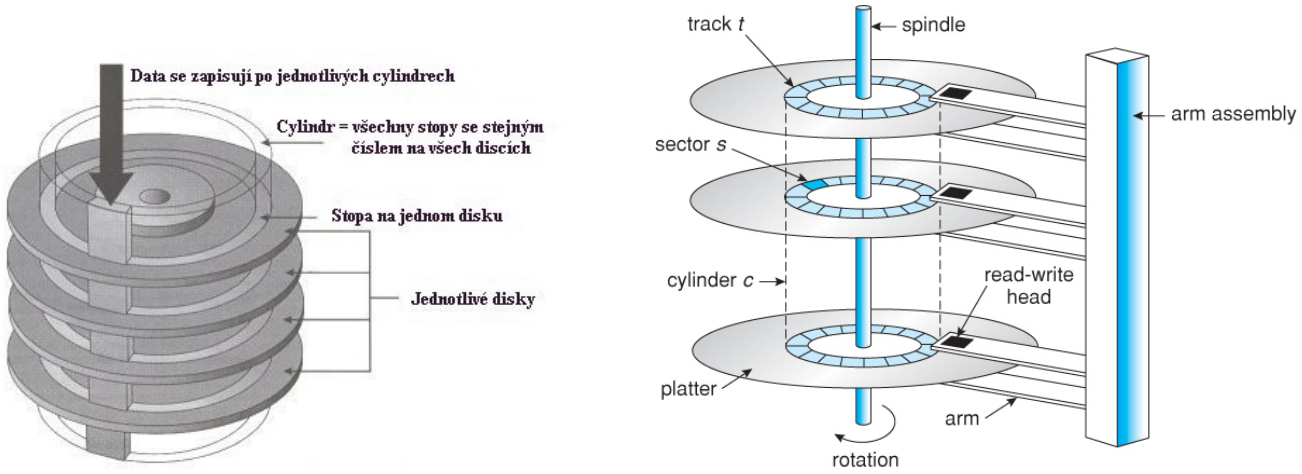


Obrázek 7.2: Vnitřek pevného disku[1]

Složení pevného disku:

- plotny, každá má dva povrchy, jsou na společné ose,
- čtecí/zápisové hlavy na společném rameni,
- mechanika pro pohyb hlav,
- elektromotor pro roztáčení ploten,
- deska s tištěnými spoji, řadič, vyrovnávací paměť (buffer),
- úzký datový kabel propojující řadič s mechanismem řízení hlav,
- konektory, příp. přepínače a propojky.

Důležitou součástí je *elektromotor*, který roztáčí plotny disku. Rychlost elektromotoru se udává v RPM (Revolutions Per Minute nebo Rotations per Minute), počtu otáček za minutu. U větších pevných disků pro desktopy to bývá 7200 rpm (až 10 000 rpm), u notebookových s menším průměrem pak 5400 rpm. Rychlejší (15 000 rpm), obvykle s rozhraním SAS, jsou určeny pro servery.



Obrázek 7.3: Geometrie pevného disku[2]

Geometrie disku. Pevný disk se skládá z několika *kotoučů*, každý má dva *povrchy* (plotny). Data jsou na plotně ukládána ve stopách a sektorech:

- *stopa* = soustředná kružnice na povrchu,
- *sektor* = výsek na této kružnici, 1 sektor obvykle obsahuje 512 B dat (tj. $\frac{1}{2}$ kB), nejmenší jednotka na disku, se kterou dokáže disk pracovat,
- *cylindr* = souhrn všech stop se stejným průměrem přes všechny povrchy, „dutý“ válec zahrnující stopy se stejným průměrem přes všechny povrchy,
- *cluster* (ve Windows), *blok* (v UNIXových systémech) = nejmenší jednotka na pevném disku, se kterou dokáže pracovat operační systém (několik sektorů, typicky mocnina dvou, většinou 8, 16 nebo 32).

[1] Zdroj: http://rayer.ic.cz/hardware/hddtech.htm

[2] Zdroje: http://www.fi.muni.cz/usr/pelikan/ARCHIT/TEXTY/GEOMHD.HTML, https://www.cs.uic.edu/ jbell/CourseNotes/OperatingSystems/10_MassStorage.html

Poznámka

Řadič disku dokáže pracovat vždy s celými sektory, kdežto *operační systém* dokáže pracovat vždy s celými clustery (v Linuxu bloky). Jeden cluster/blok je obvykle 8, 16 nebo 32 sektorů (mocniny čísla 2).

Jak bylo výše řečeno, jeden sektor obsahuje obvykle 512 B dat. Ale to neplatí o všech discích. Disky označené „*Advanced Format*“ (standard z roku 2010) mají ve svých sektorech vždy 4 KB dat (tj. 4096 B). Účelem je především navýšit množství dat, která mohou být na disku adresována. Disk typu „Advanced Format“ lze používat pouze na počítači s operačním systémem, který tento typ podporuje.

Každý používaný povrch má svou vlastní *čtecí/zápisovou hlavu*. Proto pokud jsou všechny povrchy používány, najdeme v disku dvojnásobný počet hlav, než kolik je kotoučů.

Ne každý povrch musí nutně být používán – je možné, aby „krajní“ povrchy nebyly používány, například horní povrch nejvrchnější plotny.

Původně byl na všech stopách stejný počet sektorů (tj. výseků na stopě), ale se zvyšující se kapacitou disku a počtem stop na povrchu se projevil velký rozdíl mezi délkou sektorů na vnějších a vnitřních stopách (stopy blíže středu disku jsou logicky výrazně kratší než stopy blíže okraji disku), přičemž počet bitů uložených v jednom sektoru musel být stejný. V sektorech u okraje disku se plýtvalo místem, v sektorech u středu disku zase při vyšší hustotě signálů hrozilo riziko chybného čtení.

V současné době se tento problém řeší především dvěma vylepšeními (oběma najednou) – *prekompenzací zápisu* (počítá se s případnými chybami zápisu na „hustém“ vnitřním sektoru) a *zónovým záznamem* (na různých stopách je různý počet sektorů), přičemž všechny sektory i nadále nesou stejné množství dat. Obě technologie jsou popsány dále.

Způsoby adresování stanoví, jak se určuje fyzické umístění konkrétních dat na disku.

Cylinder–Head–Sector (CHS) je starší způsob adresace používaný u starých IDE disků. Jeho výhodou je zohlednění geometrie disku. Umístění dat je určeno třemi údaji (tedy jde o nelineární, víceúrovňové adresování), a to cylindrem (válcem), hlavou (ta určuje povrch, na kterém se bude zapisovat, průnik se zadaným válcem je stopa) a sektorem. Nejkrajnější cylindr (a tedy i stopa) má číslo 0.

Hlavní nevýhodou této metody jsou především omezení vyplývající z počtu bitů, které řadiče, BIOSy, kabely atd. používají pro uložení těchto tří údajů, to jest 512 MB na disk. EIDE tyto hodnoty rozšířilo, ale BIOSem nastavený strop zůstal (10 bitů na cylindr, tedy $2^{10} = 1024$ cylindrů), proto citelné omezení kapacity disků se sice zmírnilo, ale zůstalo (těsně pod 8 GB). Proto se u současných disků CHS nepoužívá.

Co se BIOSu týče, u disků s větším počtem cylindrů problém řeší tak, že hlásí nižší počet cylindrů než ve skutečnosti (tolik, jaká je jeho vlastní hranice), ale naopak větší počet ploten než ve skutečnosti.

Logical Block Addressing (LBA) je druhý způsob adresování, který se používá dodnes. LBA bylo původně určeno pro SCSI disky, později je převzal standard ATAPI v novějších verzích. LBA je lineární adresování na logické úrovni, geometrii disku nezohledňuje. Sektory jsou číslovány od 0, začíná se na prvním (vnějším) cylindru, přes všechny povrchy, pak druhý cylindr, atd.

Poznámka

To znamená, že data jsou ukládána po cylindrech, nikoliv po površích.

Pro adresy jsou opět stanoveny limity – původně to bylo 28 bitů na adresu, později byl určen vyšší limit, 48 bitů na adresu, který je zcela dostačující i pro současné terrabytové disky.

Dnes se již prakticky výhradně setkáváme s disky adresovanými podle LBA, nicméně tato logická adresa je řadičem překládána na CHS adresu (podle novějšího standardu s více bity na jednotlivé části adresy), protože řadič potřebuje vědět, kde konkrétně přímo na plotnách má „zasahovat“.

Parkování hlav je vhodné umísťování hlav, které má zabránit poškození disku při náhodném nárazu. Při vypnutí napájení disku nebo při nárazu dojde ke „spadnutí“ hlav na disky (vzduchový polštář přestane existovat), čímž se mohou poškodit data.

Autopark (automatické zaparkování hlav po vypnutí počítače) je založen na tom, že po vypnutí se disk chvíli otáčí setrvačností, tím se vyrobí dostatek energie pro přemístění hlav do parkovací oblasti (využívá se princip rekuperace).

U starších disků byl vyhrazen sektor, nad který se před vypnutím disku přesunuly hlavy (obvykle některý vnitřní sektor disku). U novějších disků s lineárním motorem umístění parkovací oblasti pro hlavy závisí *na rozhodnutí výrobce* (může to být některý sektor, a nebo místo mimo povrchy disku, některé disky dokonce parkují kdekoliv na povrchu). Z bezpečnostních důvodů (pro případ poruchy či otřesu) jsou plotny pokryty tenkou měkkou ochrannou vrstvou, která má zmírnit následky nečekaného dopadu.

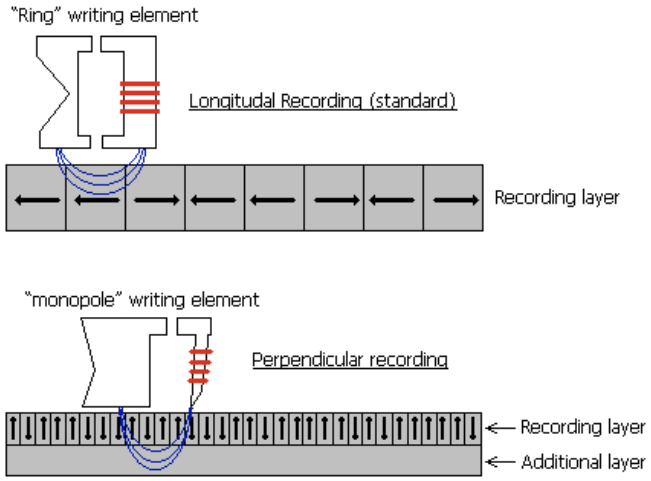
Zvláště u notebooků se parkování provádí velmi často (i několikrát za minutu, jsou vybaveny pohybovým čidlem), což snižuje životnost disku.

7.2.3 Čtení a zápis

Typy zápisu: u starších disků se používal podélný zápis, u nových kolmý zápis. Ještě novější je „šindelový“ zápis, který je však vhodný jen pro určité druhy disků.

Podélný zápis (Longitudinal Magnetic Recording – LMR) znamená, že vektory magnetické indukce (orientované magnetické pole) jsou zapisovány vodorovně s povrchem disku. Možná hustota zápisu jen asi do 150 GB/inch^2, protože při vyšší hustotě dochází k jevu zvanému paramagnetismus, jehož důsledkem je samovolná ztráta uložených dat (magnetická pole pro různé bity se navzájem ovlivňují).

Kolmý zápis (Perpendicular Magnetic Recording – PMR) se komerčně používá od roku 2005 (firma Toshiba). Vektory matematické indukce nejsou orientovány podélně s povrchem, ale kolmo na něj a disková hlava pro zápis má jinou konstrukci, aby dokázala takto orientovaná magnetická pole přečíst. Metoda je založena na přidání nové vrstvy pod vrstvu s uloženými daty, nová vrstva je z magnetického materiálu.



Obrázek 7.4: Podélný vs. kolmý zápis[3]

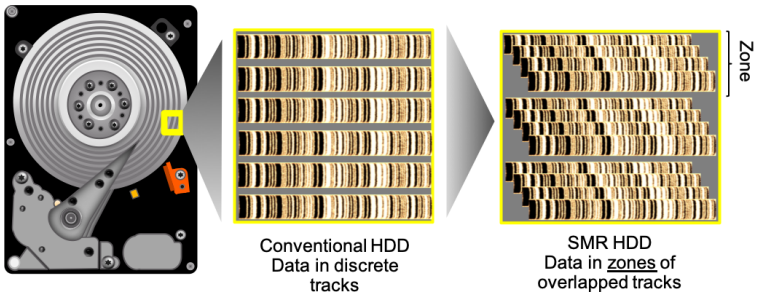
Vrstvený zápis (Shingle Magnetic Recording – SMR) je momentálně nejnovějším počinem firmy Seagate („a shingle“ je v překladu „šindel“), jedná se o vylepšení technologie kolmého zápisu. Jednotlivé stopy na povrchu se posunuly k sobě a dokonce se „šindelovou“ metodou překrývají (nejdřív je zapsána stopa blíže okraje kotouče, pak stopa o něco dál směrem ke středu tak, že tu předchozí částečně překrývá, následuje další, atd.).

[3] Zdroj: http://notebook.cz/clanky/technologie/2006/kolmy-zapis

Poznámka

Uvědomme si, co to znamená, když jedna stopa částečně překrývá sousední stopu – z toho vyplývá, že pokud chceme provést zápis do některé stopy, nutně přepíšeme i obsah sousední stopy, která je blíže středu. To znamená, že *v základu* je tato metoda určena spíše pro sekvenční zápis, kdy zapisujeme jednotlivé stopy ve směru od okraje kotouče směrem doprostřed. V jiných případech se zápis zpomaluje.

Aby bylo možné zapisovat i jinak než sekvenčně, budou stopy sdruženy do tzv. *zón*. Stopy v rámci téže zóny se (částečně) překrývají, stopy na rozhraní mezi zónami se nepřekrývají. Tím byl požadavek používat sekvenční zápis zmírněn a omezen na zápisy uvnitř téže zóny. V každé zóně je definován zápisový ukazatel (kurzor), který určuje, kde byl naposledy ukončen zápis a tedy od kterého místa je povoleno zapisovat. Tedy od řadiče se vyžaduje poněkud více inteligence než u technologie PMR.



Obrázek 7.5: Srovnání PMR a SMR zápisu[4]

Další modifikace: na disku existují zóny se SMR zápisem a zóny s PMR zápisem, SMR zóny jsou určeny pro sekvenční zápis velkých souborů, PMR pro běžné použití. Případně se SMR zónami se dá zacházet podobně jako u SSD – v bufferu se zkompletuje obsah celé zóny, ta se pak (sekvenčně) z tohoto bufferu zkopíruje na disk.

Hlavním účelem této technologie je zvýšit celkovou hustotu záznamu, třeba i na úkor propustnosti datového rozhraní (tj. počítá se s tím, že zápis asi bude pomalejší). Typické využití těchto disků je tam, kde se zapisuje větší množství dat sekvenčně, třeba v NASu pro zálohování, případně ukládání záznamu kamery. Ovšem mezi zákazníky není tato technologie moc oblíbená.

Poznámka

Jsou dva základní druhy SMR disků: *drive-managed*, na kterých určuje organizaci dat diskový řadič, a *host-managed*, kde organizaci dat určuje operační systém. Řízením organizace dat je míněno určování, co kam jakým způsobem bude zapsáno.

Drive-managed SMR disky mají jednu nepříjemnou vlastnost: pokud se dostanou do stavu zahlcení vstupy (což se může stát i při běžném kontinuálním zapisování, které trvá dlouho a tedy fronta požadavků je plná), začnou přeuspořádávat data již na disku uložená, aby zoptimalizovaly jejich umístění. To může trvat minuty, ale také celé hodiny. Během této doby disk nereaguje, není možné ho používat (ani pro čtení). Proto drive-managed SMD disky nejsou vhodné do RAID polí (prakticky rozloží fungování celého pole) a v domácím prostředí je nelze doporučit ani do NAS, případně pro určité konkrétní způsoby použití (například pro ukládání kontinuálního výstupu kamery).

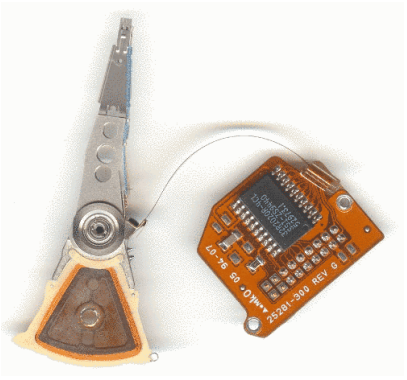
[4] Zdroj: https://zonedstorage.io/introduction/smr/
[5] Zdroj: http://notebook.cz/clanky/technologie/2007/HDD-technologie

Čtecí a zápisová hlava jsou sice dvě různé komponenty, ale najdeme je na jediném rameni, a proto spíše hovoříme o čtecí/zápisové hlavě. Každý povrch, který je používán pro záznam, má vlastní hlavu. Jde o *cívky* navinuté na jádrech.

Pevný disk neustále rotuje a hlava je nadnášena ve výšce menší než 1 μm nad povrchem (vzduchový polštář). Čím menší je vzdálenost hlavy a povrchu, tím hustěji lze zapisovat.

Zápis probíhá následovně:

- pokud hlavou prochází el. proud, vytvoří se magnetický tok, který se uzavírá v mezeře mezi hlavou a povrchem disku, čímž ovlivňuje magnetickou vrstvu na disku,
- podle toho, kterým směrem v cívce teče proud, se vytvářejí různě zmagnetizovaná místa na povrchu,
- vznikají *magnetické reverzace* (místa změny magnetizace).



Obrázek 7.6: Rameno s čtecí a zápisovou hlavou disku[5]

Čtení je víceméně opačný proces – hlavy reagují na magnetické reverzace, které vyvolají v jádru cívky magnetický tok magnetický tok je zpracován jako el. impuls.

Vystavování hlav: vystavovací mechanismus je zařízení, které vystavuje čtecí/zápisové hlavy na správné místo nad povrchem disku. Starší disky používaly krokový motor, novější disky využívají *lineární motor* používající elektromagnet – hlavy se vystavují podle elektrického proudu protékajícího elektromagnetem s ním spojeným, tento elektromagnet je uložen v silném magnetickém poli jiného permanentního magnetu.

Další informace

https://www.root.cz/clanky/soucasnost-a-budoucnost-pevnych-disku/

7.2.4 Technologie

AHCI (Advanced Host Controller Interface) je rozhraní pro komunikaci s SATA řadičem, a to nezávislé na hardwaru (tj. lze takto komunikovat s řadičem disků různých výrobců podporujících AHCI). K čemu slouží AHCI:

- podpora NCQ (viz dále),
- podpora Hot Plug (možnost připojovat a odpojovat SATA zařízení za běhu), což je spíše pro servery,
- eSATA, mSATA, M.2 a dalších podobných technologií,
- vylepšená správa napájení, možnost slučování příkazů do shluků a tím snižování režie vykonávání příkazů, atd.

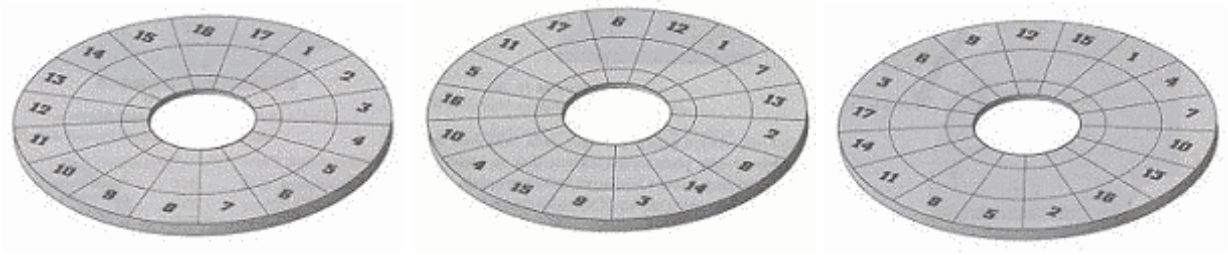
Tato specifikace byla vytvořena společností Intel a v současné době je podporována ve všech novějších operačních systémech.

NVMe (NVM Express) je nástupce AHCI používaný vnějšími paměťmi (typicky SSD) komunikujícími přes sběrnici PCIe nebo rozhraní M.2 (což ve skutečnosti taky znamená komunikaci přes PCIe). U pamětí připojených přes SATA se nepoužívá.

NVMe má oproti AHCI mnohem vyšší propustnost, také dokáže mnohem lépe využít sběrnici PCIe – AHCI by bylo při připojení do PCIe nebo M.2 slotu „úzkým hrdlem“.

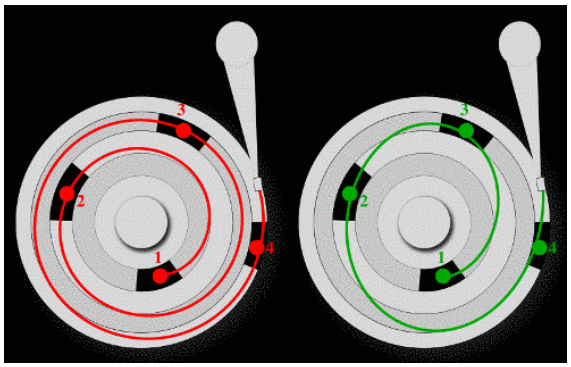
Co se týče podpory v operačních systémech, do Windows byl ovladač dodán aktualizací do verze 7, od 8.1 je podporován nativně. V ostatních operačních systémech (včetně Linuxu a OS X) je NVMe běžně podporováno.

Prokládání (interleave factor): disk se otáčí velmi rychle, proto pokud data následují fyzicky hned za sebou, dostane se k následujícím až po další otočce. To by při zvyšování hotnoty RPM (otáček za minutu) fakticky znamenalo zpomalení místo zrychlení.



Obrázek 7.7: Prokládání 1:1, 1:3, 1:6[6]

Při využití prokládání data nejsou zapisována do sektorů přímo za sebou, ale do každého n-tého sektoru – potom je faktor prokládání 1:n.



Obrázek 7.8: Disk bez NCQ a s NCQ[7]

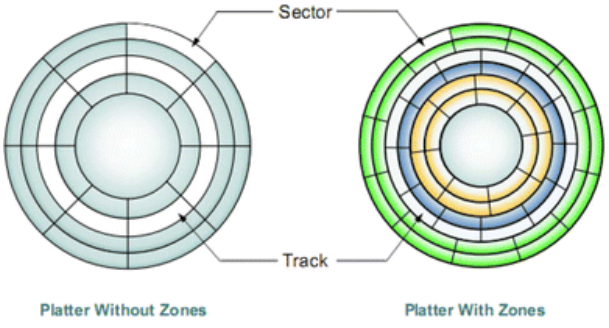
NCQ (Native Command Queuing) je přirozené řazení příkazů ke čtení/zápisu dat. Jedná se o vylepšení technologie prokládání, kdy jsou příkazy k práci s jednotlivými oblastmi na disku seřazeny tak, aby lépe odpovídaly skutečné dráze hlav nad diskem a je možné provést i více příkazů během jediné otočky disku.

Prekompenzace zápisu (Write Precompensation) počítá s tím, že při velmi hustém zápisu se jednotlivé dipóly (zmagnetizovaná místa, miniaturní magnetické částice) mohou vzájemně ovlivňovat, protože jsou moc blízko sebe. Při prekompenzaci se tedy dipóly záměrně umístí posunutě (šikmo) a počítá se se vzájemným ovlivněním dipólů, které se tak samy posunou do správné pozice.

Prekompenzace se provádí jen na stopách (cylindrech) blíže ose disku, stopa, od které se má začít, je nastavena v BIOSu. Najdeme ji obvykle pod zkratkou CPZ.

Zone Bit Recording (ZBR) řeší problém různé délky sektorů na stopách různě vzdálených od osy disku. Při ZBR je disk rozdělen do zón podle vzdálenosti cylindrů od středu disku. V zónách blíže středu je méně sektorů, v zónách blíže okraji disku je více sektorů.

U specifikací disků se setkáme i s dalšími technologiemi, namátkou SilentSeek pro co největší ztišení pohybu hlaviček nad povrchy, teplotní kalibrace, IntelliSeek, AFC a dalších.



Obrázek 7.9: Srovnání geometrie disku bez ZBR a s využitím ZBR pro vnitřní stopy[8]

[6] Zdroj: http://www.fi.muni.cz/usr/pelikan/ARCHIT/TEXTY/GEOMHD.HTML
[7] Zdroj: http://expertester.wordpress.com/2008/07/19/how-to-enable-ahci-without-reformating/
[8] Zdroj: https://link.springer.com/article/10.1007/s41870-018-0144-x

7.2.5 Největší a nejmenší

RAMAC 305 byl v r. 1956 vyroben firmou IBM, vážil asi 1 tunu a měl kapacitu 5 MB. Šlo o skříň s 50 disky průměru automobilového kola. Magnetická záznamová vrstva byla tvořena oxidem železa rozprášeným v plastu, výsledná barva byla stejná jako u Golden Gate. Jeho rychlost byla poměrně vysoká – 1200 RPM.

První Winchester byl vyroben firmou IBM jako model IBM 3340 v roce 1970 pro počítač System 370. Skládal se ze dvou částí – pevný disk bez možnosti výměny (30 MB) a na stejné ose výměnný pevný disk (také 30 MB). Výsledná kapacita se proto zapisovala dvěma údaji, 30-30 (tento zápis je podobný Winchestrovce kalibru 30 označované 30-30, od té doby se pevné disky označují názvem Winchester).

IBM Microdrive je pro změnu miniaturní produkt firmy IBM z roku 1998. Jeho rozměry jsou pouze $1,4 \times 1,7 \times 0,2$ palce, kapacita až 340 MB při použití obou povrchů disku.

Obrázek 7.10: IBM Microdrive[9]

7.2.6 Sledování pevných disků

U disků se hovoří o vlastnosti *Střední doba mezi chybami* (MTBF – Mean Time Between Failures). Jedná se o statistický údaj, který naznačuje, jak je to se spolehlivostí daného disku.

Poznámka

K poškození pevného disku může dojít různými způsoby, například opotřebení, náraz hlavy na povrch disku, časté výkyvy teplot, zadřená ložiska, nefunkční motorky, poškozená elektronika, atd., narušená integrita. Některé typy poškození se dají opravit (narušená integrita, drobné poškození povrchu, jehož důsledkem je poškození dat na tomto místě zapsaných), jiné ne, ale obvykle bývá možné alespoň zachránit data (ne vždy).

Uživatelé se nejčastěji setkávají s *vadnými sektory*. Ty se většinou projeví spíše při čtení, bohužel se stává, že zápis se bez chybových hlášení provede do vadného sektoru.

Občas je třeba zkontrolovat disk vhodným programem (scandisk, chkdsk a další specializované programy, nejlépe nezávislé na operačním systému). Nalezený vadný sektor je „přemostěn“ – program se pokusí z něj zachránit data a poté jej označí jako vadný (aby již nemohl být používán). Přemosťování vadných sektorů je sice užitečné, ale když je hlášeno mnoho chyb, pak je lepší uvažovat o novém disku, i když starší ještě funguje.

S.M.A.R.T. (Self Monitoring Analysis and Reporting Technology) je mechanismus pro monitorování, analyzování a hlášení chyb pevného disku nezávisle na operačním systému. Tento mechanismus je integrován v novějších discích (asi tak od druhé poloviny 90. let).

Různé disky poskytují různý počet informací (čím novější disk, tím více informací). Řadič disku neustále sleduje stanovené hodnoty (počet chybných čtení, počet chyb při zápisu, změny rychlosti roztočení disku, počet přemapovaných sektorů, počet „odpracovaných“ hodin, teplota disku, atd.) a zapisuje je na stanovené místo, odkud si tyto hodnoty dokáže vytáhnout specializovaný program a případně „vyvolat poplach“. Některé obvyklé parametry (je jich mnohem více):

- `Raw_Read_Error_Rate` – počet chyb při čtení,

[9] Zdroj: http://www.pcworld.com/article/127105/timeline_50_years_of_hard_drives.html

- `Spin_Up_Time` – doba, po kterou trvá roztočení ploten do potřebné rychlosti, pokud není OK, znamená to obvykle poškozený motorek,
- `Reallocated_Sector_Count` – při nalezení poškozeného sektoru se provede jeho přemapování, nahrazení rezervním sektorem; tato hodnota ukazuje množství takovýchto přemapování, když roste, disk brzy dosloužil (tj. sledujeme četnost změn),
- `Seek_Error_Rate` – počet chybných vystavení hlaviček, narůstající chyby znamenají, že mechanismus posunu hlaviček není úplně vpořádku,
- `Spin_Retry_Count` – počet nepovedených startů motorku pro roztočení ploten, při chybách bychom měli zálohovat a pořídit nový disk,
- `Temperature_Celsius` – je to spíše informační údaj (obvykle by neměla překročit cca 60 °C).

Kromě konkrétní hodnoty (*raw value*) získáme odpovídající „index rizika“ – *value*, což je hodnota v rozmezí 0–100 (číslo 0 obvykle znamená maximální riziko, 100 je OK), a k tomu také hodnotu *thresh* (taktéž 0–100) představující hranici mezi rizikem a dobrým stavem. Tedy čím nižší hodnota value, tím hůře!

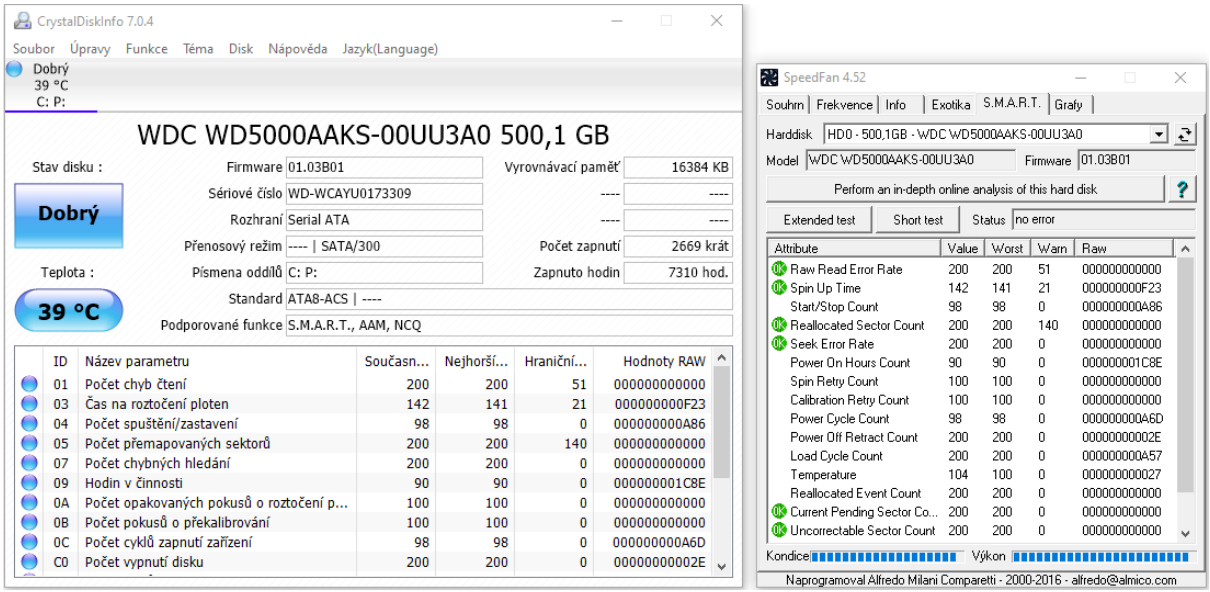
V příslušných nástrojích lze obvykle získat všechny tyto hodnoty v tabulce, jejíž řádek může vypadat například takto:

ID#	Attribute	Flag	Value	Worst	Thresh	...
3	Spin_Up_Time	0x0007	079	079	011	...
7	Seek_Error_Rate	0x000f	061	058	051	...
194	Temperature_Celsius	0x0022	074	067	000	...

...	Type	Updated	When_failed	Raw_Value
...	Pre-fail	Always	-	6260
...	Pre-fail	Always	-	7863642
...	Old_age	Always	-	28 (Lifetime)

Tabulka 7.2: Ukázka tabulky s parametry S.M.A.R.T., vybrané tři řádky

V tabulce 7.2 vidíme, že testovaný disk má zatím dobrou dobu roztočení ploten, ale počet chybných vystavení hlaviček nebezpečně narůstá a blíží se (shora) k hodnotě thresh, teplota je ucházející.



Obrázek 7.11: Hodnoty S.M.A.R.T. v Crystal Disk Info a SpeedFan (tentýž disk i systém)

K nastavení S.M.A.R.T. se částečně dostaneme v BIOSu, ale jde pouze o základní nastavení (především povolení samotného mechanismu). Abychom mohli s tímto mechanismem lépe pracovat, potřebujeme k tomu specializovanou aplikaci:

- ve Windows: HDDScan, CrystalDiskInfo, SpeedFan, HDD Health, SiS Sandra, atd., výrobci disků také často poskytují vlastní nástroje
- v Linuxu: balíček smartmontools, k nalezení ve všech repozitářích, obvykle je již nainstalován

Nástroje pro údržbu a sledování disků často najdeme i u jejich výrobců (stačí hledat na stránkách výrobců, například pro Seagate, Samsung, Western Digital, Hitachi). V těchto případech samozřejmě používáme dotyčný nástroj pouze u disku od daného výrobce.

Další informace

Podrobněji o S.M.A.R.T. na https://www.svethardware.cz/kdyz-disk-neco-boli/11178-2.

Úkol

Vyberte si některý z programů pro čtení S.M.A.R.T. údajů (například pro Windows existují volně šiřitelné nástroje HDDHealth, HDDScan, CrystalDiskInfo) – některé z nich jsme viděli na přednášce. Vyzkoušejte, zjistěte s jejich pomocí tyto údaje:

- rozhraní disku, podporované technologie, teplotu disku,
- počet zapnutí (pokud jej dokáže vybraný nástroj zjistit),
- S.M.A.R.T. hodnoty pro počet chyb čtení, počet přemapovaných sektorů, počet opakovaných pokusů o roztočení disku, počet chyb při zápisu sektorů.

7.2.7 Vybíráme pevný disk

Důležité vlastnosti pevných disků:

- rozměry, podpora novějších technologií,
- *doba vystavení* (seek time, track-to-track seek) – čas nutný k vystavení hlavy, tj. k jejímu přesunutí na žádané místo s daty, určena jako jedna třetina času potřebného pro pohyb přes celý poloměr disku, obvykle několik ms.

  Výrobci se dobu vystavení snaží co nejvíc minimalizovat, proto zápis dat obvykle probíhá po cylindrech (s využitím vhodného řetězení instrukcí),
- *doba čekání* (rotary latency period) – když se hlava dostane nad správnou stopu, musí počkat, až bude nad správným sektorem, určuje se většinou jako doba poloviny otáčky disku,
- *přístupová doba* (access time) – rychlost vyhledávání dat na disku (doba vystavení + doba čekání), obvykle kolem 10 ms,
- *rychlost otáčení disku* (RPM) – dnes běžně 4200, 5400, 7200 otáček/min, od 10 000 otáček/min je nutné disk chladit přídavným chladičem a stroj s tímto diskem by měl být za chodu stabilně usazen (tyto disky jsou většinou určeny pro servery nebo pro velmi výkonné pracovní stanice).

Disk mívá obvykle jednu až čtyři plotny, a odpovídající počet hlaviček (pro každý zapisovatelný povrch jednu). Platí, že čím méně je ploten, tím je disk tišší a tím méně tepla produkuje. To bychom měli zvážit zvláště tehdy, když potřebujeme co nejtišší počítač (například HTPC).

Údaj o počtu ploten se ne vždy při koupi dovíme, zvláště u značky Western Digital je problém se jich dopátrat. Často se k podobným údajům dostaneme na stránkách výrobce v souborech s technickými specifikacemi disků. U Western Digital se setkáme s pojmem *IntelliPower*, který bývá mylně interpretován jako možnost řídit otáčky disku podle vyžadované spotřeby, ale ve skutečnosti jsou tyto otáčky nastaveny již z výroby a nelze je modifikovat.

Magnetické disky už nejsou tak výhodný artikl jako dřív a obecně klesá i rozmanitost nabídky. Nejširší výběr mají momentálně Seagate a Western Digital, kde najdeme několik řad disků pro různé účely:

- Seagate nabízí disky Barracuda pro běžné použití, IronWolf pro datová centra, NAS apod., SkyHawk pro ukládání dat z kamer.
- WD nabízí řadu Blue pro běžné použití, Black pro maximální výkon, Red pro datová centra a NAS, Purple pro ukládání dat z kamer a Green pro úsporný provoz.

V současné době se můžeme běžně setkat s *hybridními disky* (obvykle se označují SSHD), které v jednom pouzdře kombinují klasický pevný disk s plotnami a dále flash čip. Jedná se o dobrou koupi, protože výkonností, spotřebou i dalšími vlastnostmi (i cenou) jsou tyto komponenty někde mezi klasickými pevnými disky a SSD (budeme probírat v části této kapitoly o flash pamětech), tj. vyšší propustnost včetně rychlostí čtení a zápisu, dobrá životnost, vysoká kapacita, apod.

Další informace

- http://www.tomshardware.com/reviews/msata-ssd-flash,2948-3.html
- http://www.svethardware.cz/art_doc-D35E78C6C3B894FFC125727F005BE243.html
- http://assembler.cz/carch/rok2004-2005/lekce14.pdf
- http://pctuning.tyden.cz/hardware/disky-cd-dvd-br/18863-definitivni-pruvodce-optimalizaci-diskoveho-prostoru
- http://www.fccps.cz/download/adv/frr/hdd/hdd.html

7.3 RAID

RAID (Redundant Array of Independent Disks) znamená využití pole disků (tj. více než jeden disk). Jde tedy o rozdělení dat na více disků. Účelem je zvýšení zabezpečení dat (proti ztrátě) a případně rychlosti. Rozlišujeme několik typů RAID podle způsobu využívání jednotlivých disků. Mezi těmito typy jsou rozdíly také v minimálním počtu vyžadovaných disků.

7.3.1 Typy RAID

RAID 0 je rozdělení dat mezi dva nebo více disků, každá informace je uložena jen na jednom místě. Rozlišujeme dva typy:

- *zřetězení* (nepoužívá se) – po zaplnění prvního disku se použije druhý, po jeho zaplnění třetí, atd., zřetězujeme disky a stíráme fyzické hranice mezi nimi,
- *prokládání* (stripping) – data se při zápisu rozdělí a jednotlivé části jsou uloženy na různé disky (může se rozdělit i jeden soubor na více disků), dovoluje paralelní práci více procesů (vláken) s tímtéž souborem (na částech uložených na různých fyz. discích).

Výhodou je, že máme sice dva menší disky, ale logicky jeden velký, dále rychlost paralelní práce se souborem bývá vyšší (u druhého typu).

Nevýhodou RAID 0 je absence odolnosti proti chybám, data nejsou redundantní,[10] při poruše vždy část (ale často vše) ztratíme. Také může nastat problém s případnou obnovou poškozených dat.

JBOD (Just a Bunch of Disks) se podobá RAID 0 v podobě zřetězení. Pokud máme více disků zapojených do JBOD, nejdřív se zaplňuje první, když je zaplněn, přijde na řadu druhý, atd.

Rozdíl oproti RAID 0 typu zřetězení je v tom, že JBOD nevyžaduje, aby všechny zapojené disky měly stejnou kapacitu – pokud takto zřetězíme několik disků, dokáže plně využít kapacitu každého z nich.

RAID 1 je implementace *zrcadlení* (mirroring). Zapisovaná data se zaznamenávají na (minimálně) dva disky zároveň, každá informace je uložena na obou (resp. všech) discích zároveň.

Oproti předchozímu typu je hlavní výhodou bezpečnost – v případě výpadku jednoho disku máme data zálohovaná na druhém, další výhodou je zvýšení rychlosti čtení (čte se paralelně z obou disků).

Nevýhodou je nutnost dvojnásobné diskové kapacity oproti skutečné potřebě (předpokládáme, že oba disky mají stejnou kapacitu). Zápis je pomalejší, je třeba data zapsat na oba disky v RAIDu.

RAID 0 (stripping)

disk A	disk B
1	2
3	4
5	6
7	8
⋮	⋮

RAID 1 (mirroring)

disk A	disk B
1	1
2	2
3	3
4	4
⋮	⋮

Obrázek 7.12: RAID 0 a 1

Kombinace RAID 0 a 1 je dvoustupňový RAID. Rozlišujeme dva typy podle způsobu kombinace typu 0 a 1:

- RAID 0+1 (také RAID 01) – data ukládáme prokládaně na dva disky – A, B, oba pak zrcadlíme na další dva disky – C, D (tj. 4 disky A, B, C, D, při zápisu nejdřív data dělíme a pak zrcadlíme),
- RAID 1+0 (také RAID 10) – první část uložíme zároveň na disky A, B, druhou na disky C, D, třetí na A, B, atd. (tj. taky 4 disky, ale zrcadlení provádíme před dělením, resp. v rámci dělení).

RAID 01

disk A	disk B	disk C	disk D
1	2	1	2
3	4	3	4
5	6	5	6
7	8	7	8
⋮	⋮	⋮	⋮

RAID 10

disk A	disk B	disk C	disk D
1	1	2	2
3	3	4	4
5	5	6	6
7	7	8	8
⋮	⋮	⋮	⋮

Obrázek 7.13: RAID 01 a 10

RAID 3 využívá paritní disk. Použijeme $N+1$ stejných disků, z nichž N je určeno pro ukládání dat, na poslední (paritní) disk je uložen kontrolní součet XOR těchto dat.

[10] Redundantní data jsou data uložená na více místech, tj. máme více kopií dat.

Pokud vypadne paritní disk, data zůstanou netknutá. Jestliže vypadne některý z datových disků, je možné jeho obsah zrekonstruovat (podle stejných pozic všech ostatních disků včetně paritního).

Bezpečnost dat je poměrně velká. „Úzkým hrdlem“ je však paritní disk – na něj se zapisuje při každém zápisu na kterýkoliv jiný disk, čímž je zápis zpomalován, a navíc se tím snižuje jeho životnost. Selhání paritního disku je statisticky výrazně pravděpodobnější než selhání ostatních disků.

RAID 2 a RAID 4 jsou varianty RAID 3:

- RAID 2 – oproti RAID 3 se navíc na N disků data stripují (dělí) mezi jednotlivé disky a dále místo prosté parity jsou data chráněna Hammingovým kódem,
- RAID 4 – oproti RAID 3 se parita na paritní disk vytváří po blocích a provádí se stripování mezi disky.

RAID 5 je funkčně podobný RAIDu 4, ale jde o pokus eliminace zvýšeného zatížení jednoho paritního disku. Bloky dat se stripují na různé disky a některý z $N + 1$ disků je použit jako paritní, ale u každé $(N + 1)$tice bloků jde o jiný disk.

RAID 3

disk A	disk B	disk C	disk D
1	2	3	P(1–3)
4	5	6	P(4–6)
7	8	9	P(7–9)
⋮	⋮	⋮	⋮

RAID 5

disk A	disk B	disk C	disk D
1	2	3	P(1–3)
4	5	P(4–6)	6
7	P(7–9)	8	9
⋮	⋮	⋮	⋮

Obrázek 7.14: RAID 3 a 5

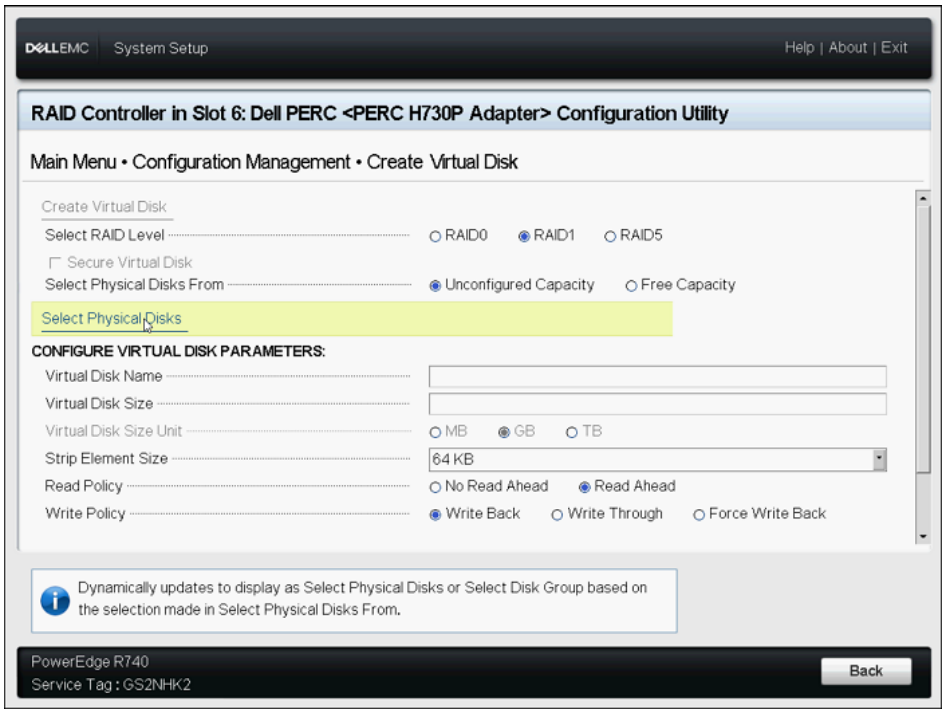
7.3.2 Nastavení RAID pole

Řadič RAID (fyzicky to je čip na základní desce nebo je integrován do čipsetu) se musí aktivovat v BIOSu. Dále je pole obsluhováno buď na straně operačního systému (softwarově) anebo řadičem disku.

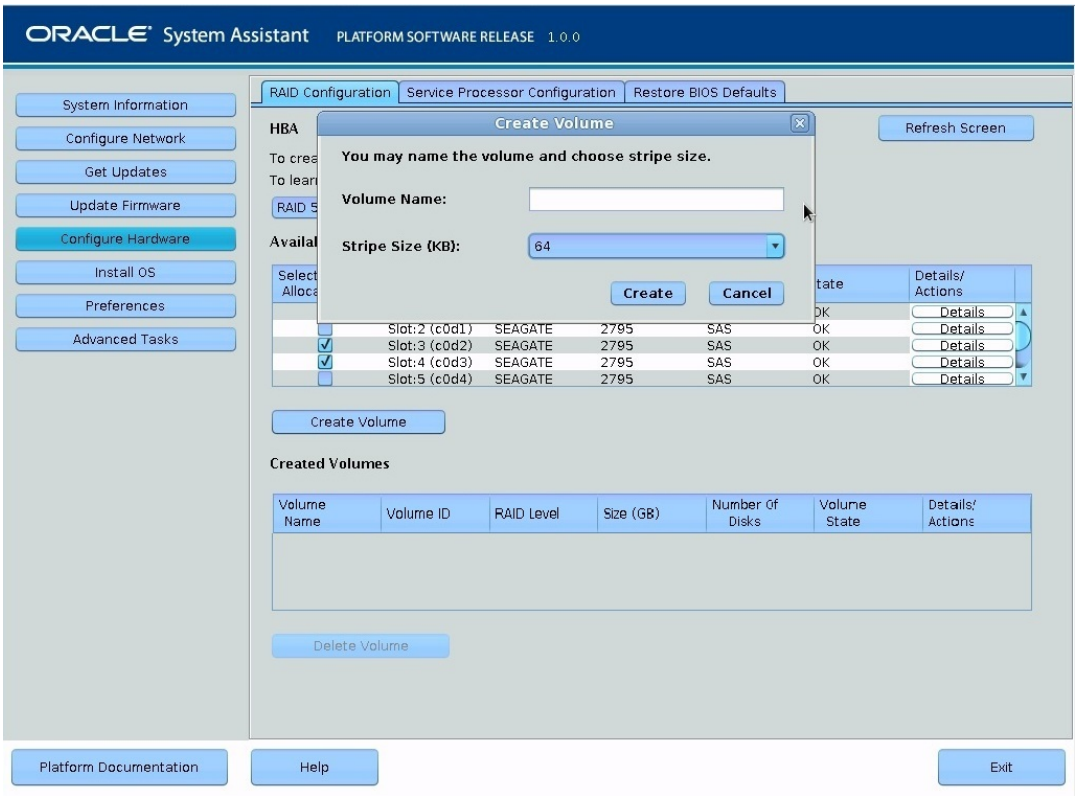
Pokud máme základní desku s podporou RAID a samozřejmě více fyzických disků k ní připojených (pokud možno se stejnými parametry), musíme nejdřív *aktivovat diskové pole*. Typ RAID závisí na základní desce a případně operačním systému, rozhodně není běžné, abychom si mohli vybírat kterýkoliv z výše uvedených. Obvykle je k dispozici několik z RAID 0, 1, 10, JBOD, 3 a 5.

Začneme v *BIOS/UEFI Setup*. Aktivace řadiče RAID může být například v části *Advanced*, položka *OnBoard Device Configuration* (ovšem v každém BIOSu/UEFI může být jinde, záleží na jeho výrobci, také to může být třeba *Integrated Peripherals*, položka *OnBoard SATA/IDE Ctrl Mode*). Najdeme položku pro RAID (opět se může jmenovat různě, podle výrobce). Vždy je třeba nastavit mód podle typu disků a případně také nastavit položku *Enabled*.

Dále je nutné nakonfigurovat samotný *řadič RAID* (zatím jsme ho jenom aktivovali). Jeho konfigurace se provádí specializovaným programem nebo v určité části UEFI, pokud je RAID řadič integrovaný na desce či v čipsetu. Do ovládacího programu (pokud je zvlášť) se dostaneme podobně jako do BIOS Setup během startu počítače po provedení testu POST stisknutím určité klávesové zkratky. Ta se liší podle výrobce, například Ctrl+I (Intel), Tab (VIA), F10 (nVidia), Ctrl+A (Adaptec), Ctrl+F4 (Silicon), Ctrl+F (Promise), atd. V tomto programu (s podobným rozhraním jako má BIOS Setup) vytvoříme RAID pole (určíme, které disky do něj mají patřit), určíme jeho typ a potřebné parametry.



Obrázek 7.15: Konfigurace RAID pole na Dell serveru[11]



Obrázek 7.16: Konfigurace RAID na Oraclu[12]

Je lepší, když jsou disky, které chceme do pole zařadit, prázdné (teprve budou formátovány na určitý

[11] Zdroj: https://www.dell.com/support/kbdoc/en-uk/000178269/dell-poweredge-how-to-create-a-virtual-disk-through-the-system-setup-on-14g-servers?lwp=rt

[12] Zdroj: https://docs.oracle.com/cd/E41059_01/html/E48314/z4000e311419867.html

souborový systém). Ovládací programy pro ten případ nabízejí automatizované vytvoření pole. Za určitých okolností je možné vytvořit RAID 1 (zrcadlení) na jednom disku s daty a druhém prázdném, pak jsou data během vytvoření pole zkopírována na druhý disk. I přesto je vytvoření RAIDu z disku s daty poněkud riskantní a u jiných typů polí jsou data ztracena.

Další informace

http://forums.pcper.com/showthread.php?t=444831 (o konfiguraci RAID včetně nastavení ve Windows)

7.4 NAS

NAS (Network Access Storage) jsou síťová úložiště dat. Na rozdíl od „obyčejných“ externích disků mají operační systém a přistupuje se k nim pomocí SFTP (většinou) nebo HTTPS (podporuje jen málo NAS disků), konfigurace se provádí přes webové rozhraní. Obvyklým rozhraním je Ethernet (RJ-45, většinou alespoň dva konektory), USB, případně eSATA nebo jiné. NAS se obvykle zapojují přes síťové rozhraní k některému síťovému prvku, jako je například router či switch.

V operačním systému na zařízení NAS (často jde o Linux a souborový systém ext4fs) jsou nainstalovány různé (softwarové) servery. Nutností je *souborový server* s podporou protokolu SMB, ale mohou být i další. Dokonce existují NAS s tiskovým rozhraním s nainstalovaným *print serverem*, tedy můžeme k nim připojit tiskárnu, která pak může být přístupná v síti.

NAS většinou při koupi neobsahuje pevné disky (ale může), tedy dokupujeme jeden nebo více pevných disků podle vlastních potřeb. Více disků v NAS může spolupracovat v RAID, často je používáno zrcadlení.

NAS bývají (například v internetových obchodech) řazeny buď do kategorie pevných disků nebo do kategorie síťových zařízení. Podle čeho vybíráme:

- *síťové rozhraní* – ethernet RJ-45 (zde také záleží na rychlosti, i pro domácnost se hodí 1Gb porty, označuje se také jako LAN nebo GLAN),
- případná síťová *bezdrátová rozhraní* – Wi-fi podle některého vhodného standardu, ale i další, například pro spolupráci se spotřební elektronikou (jako je DLNA),
- *vnější lokální rozhraní* pro připojení externích disků či jiných pamětí, a také tiskáren, pokud NAS má fungovat jako tiskový server (USB konkrétní verze, eSATA, apod.),
- *vnitřní rozhraní* pro připojení interních disků – obvykle SATA konkrétní verze,
- další komponenty – vnitřní paměť, procesor, atd.,
- *formát interních disků* – 3.5 palce nebo 2.5 palce, větší disky mají větší kapacitu a rychlost, menší disky jsou tišší a mají menší spotřebu,
- *jsou/nejsou interní disky v ceně* – často je třeba disky koupit zvlášť,
- podporované *funkce* – kromě funkce souborového (file) serveru může být podporována funkce FTP server, databázový (DB) server, tiskový (print) server, web server, obvyklá je podpora RAID, také lze někdy použít pro napojení bezpečnostního systému (IP kamer),
- podporované *přenosové techniky* – obvykle FTP a HTTP, dále je velmi vhodná podpora šifrování a zabezpečeného přenosu,
- operační systém, možnosti konfigurace, vybavenost a přehlednost *konfiguračního rozhraní*, případné možnosti rozšíření (i zde bývají appstory s aplikacemi).

Ceny NAS začínají někde mezi 1 a 2 tisíci, nejdražší stojí několik desítek tisíc korun.

Co se disků týče, existují disky přímo optimalizované pro použití v NAS. Poznáme je podle toho, že někde ve specifikaci bývá napsáno o NAS, RAID, cloudu apod., například Western Digital Red (RAID Edition). Každý výrobce NAS má určitá doporučení na parametry disků, měli bychom je alespoň částečně zohlednit.

Poznámka

Domácí routery dnes mívají běžně funkci NAS serveru. To znamená, že přes USB připojíme k routeru externí disk, a ten by měl být pak přístupný v síti. Takže takový router vlastně máme místo samostatného zařízení (které bychom jinak k routeru připojili síťovým kabelem). Toto řešení za určitých okolností může vyhovovat, ale je třeba si uvědomit, že router má procesor, který předně plní funkce spojené s provozem routeru, a my ho tímto zatížíme navíc prací NAS serveru. Rozhraní USB vyžaduje asistenci procesoru při každém přenosu, a pokud má procesor dost *jiné práce* (a ta jiná práce bývá upřednostňována), pak je přenos velice pomalý bez ohledu na to, jakou verzi USB používáme.

Nejznámější výrobci NAS serverů jsou QNAP, Synology, Western Digital.

Další informace

Zajímavý článek o NAS najdeme například na
http://www.svethardware.cz/art_doc-38BF655343D6BDB5C125734B002D5401.html.

Úkol

V některém internetovém obchodě najděte několik odlišných NAS od různých výrobců ve zhruba stejné cenové hladině. Porovnejte je podle výše jmenovaných parametrů (pokud seženete potřebné informace).

7.5 Optické paměti

7.5.1 CD

CD (Compact Disc) je optické médium (tj. pro čtení a záznam se používá laser), jednostranné (záznam na spodní straně disku). Průměr je většinou 12 cm, ale existuje i menší varianta (8 cm). Otvor pro osu je vždy stejný, 15 mm. CD (jako jednovrstvá DVD) vkládáme do mechaniky potištěnou stranou nahoru.

Stejně jako pevný disk, i na CD (DVD) mechaniku je nutné připojit napájecí kabel a datový kabel (buď IDE 40žilový nebo stíněný 80žilový, anebo SATA, podle typu mechaniky), ale navíc je nutné propojit CD/DVD mechaniku se zvukovou kartou.

Na rozdíl od magnetických médií jsou zde data uložena na spirále, nikoliv na soustředných stopách (začátek je u středu disku, což je také rozdíl oproti magnetickým médiím). Záznam je proveden pomocí různě dlouhých prohlubní (*pits*) oddělených mezerami (*lands*). Přechod prohlubeň/mezera nebo mezera/prohlubeň je logická 1, dlouhá mezera nebo prohlubeň je logická 0, díky použitému kódování nikdy nenásledují dvě jedničky za sebou.

Přenosové rychlosti se udávají jako násobek rychlosti 150 kB/s (ta je typická pro Audio CD), například 16× znamená šestnáctinásobek této rychlosti.

CD-ROM je pouze pro čtení. Jeho fyzická struktura je následující:

- základem je disk ze směsi polykarbonátu a polymetylmetakrylátu, ve kterém je vytlačena spirála s prohlubněmi,
- následující vrstva je reflexní, dobře odráží světlo laseru (vlnová délka 780 nm),
- třetí vrstva je ochranná, může být vyrobena z laku opatřeného potiskem.

Čtení probíhá pomocí prvku citlivého na světlo (fotodioda, fototranzistor apod.), který detekuje odražený laserový paprsek. Reflexní vrstva na datovém médiu odráží laserový paprsek v plné intenzitě, ale v místě prohlubně v reflexní vrstvě je odraženo méně energie. Změna intenzity laseru je interpretována jako logická 1.

Rychlost otáčení disku při čtení může být buď konstantní pro všechny pozice čtecí hlavy a nebo se může měnit se vzdáleností čtecí hlavy od osy disku, případně lze oba postupy kombinovat. Z tohoto hlediska rozlišujeme několik metod pro konstrukci CD mechanik:

- CLV (Constant Linear Velocity) – při čtení jsou otáčky průběžně (plynule) měněny podle vzdálenosti od středu (u středu pomaleji, u okraje rychleji), aby byl datový tok konstantní – tato metoda je ideální pro záznam zvuku, tedy CD DA a starší CD-ROM,
- CAV (Constant Angular Velocity) – tato metoda již zohledňuje čtení dat. U hudebních souborů je využita metoda CLV, ale *při čtení dat* se disk otáčí konstantní rychlostí, konstantnost datového toku je zajišťována logikou podle vzdálenosti od středu, tyto mechaniky jsou tišší a déle vydrží,
- P-CAV (Partial Constant Angular Velocity) – kombinace obou předchozích, blíže středu se použije CAV a dále od středu CLV,
- Multibeam – čtecí hlava čte sedm stop najednou; kolem hlavního laserového paprsku jsou po obou stranách ještě další tři, které snímají okolní stopy, účelem je zvýšit přenosovou rychlost bez nutnosti zvyšování otáček.

CD-ROM *se vyrábějí* lisováním podle předem vytvořené matrice (ta stanovuje pozice pitů), vylisovaný disk se zalije do ochranného plastu. Proto nelze na CD-ROM zapisovat v běžné mechanice, museli bychom mít matrici a specializované zařízení. Výhodou tohoto postupu je poměrně nízká cena, zvláště při hromadné výrobě, a také vysoká odolnost média.

CD-R (CD Recordable) umožňuje jednorázový zápis a po něm pouze čtení. Má stejné rozměry jako CD-ROM, ale mírně odlišnou vnitřní strukturu včetně použitých materiálů.

Na disku je vrstva *organického barviva*, do které se provádí záznam, tato vrstva je překryta ochrannou (odrazovou) vrstvou ze stříbra, slitiny stříbra s jiným kovem nebo zlata a krycí vrstvou z polykarbonátu.

Ve výrobě se do disku vylisuje spirála, která slouží jako vodítko zapisovací hlavy při zápisu. Na spirále jsou značky *ATIP* (Absolute Time in Pregroove) sloužící k synchronizaci zápisu dat konstantní rychlostí. Dále jsou z výroby na disku obsaženy informace o kapacitě disku, výrobci, požadavcích na zápis (výkon laseru), atd. ATIP značky zůstávají i po zápisu, slouží také k detekci typu disku.

Při *zápisu* hlava s laserovou diodou sleduje spirálu s orientací podle ATIP značek. V místě, kde má být prohlubeň, se zvýší intenzita laseru a tím i jeho teplota, změní se chemické (optické) vlastnosti ve vrstvě organického barviva. Ve skutečnosti se samozřejmě nic nepálí, jde o chemický proces.

Čtení probíhá stejně jako u CD-ROM, jen místo vylisovaných prohlubní máme světlá a tmavá místa.

Mechaniky pro CD-R (obecně všechny vypalovací mechaniky) používají *vyrovnávací paměť* (buffer), do které ukládají data k zápisu. Tato vyrovnávací paměť je důležitá především proto, že vypalování by správně mělo probíhat bez přerušení se zajištěním neustálého přísunu dat.

Nebezpečí „opožděného“ přísunu dat do vyrovnávací paměti řeší většina výrobců svými vlastními algoritmy (BurnProof u firmy Sanyo, JustLink u firmy Ricoh apod.). Všechny tyto algoritmy zajišťují hlídání momentálního stavu vyrovnávací paměti (kolik z jejího obsahu již bylo použito) a při poklesu pod stanovenou hranici zajistí korektní přerušení vypalování, opětovné naplnění vyrovnávací paměti a správné navázání následného vypalování.

CD-RW (CD Re-Writeable) dovolují zapisovat i opakovaně po smazání. Datová vrstva, do které se provádí záznam, je ze slitiny, která při určité teplotě krystalizuje a při jiné teplotě získává amorfní (nekrystalickou) strukturu. Místa s krystalickou strukturou odrážejí více světla a z toho vychází princip čtení dat z média (krystalická struktura je obdobou landu a amorfní struktura obdobou pitu).

Mazání disku se provádí převodem z amorfní na krystalickou strukturu – mazané místo je zahřáto na teplotu nižší než je bod tání materiálu, dojde ke krystalizaci. Následný zápis probíhá tak, že místo, které má méně odrážet světlo (analogie prohlubně u CD-R), je zahřáto na teplotu převyšující bod tání materiálu a mění se na amorfní.

Změna mezi amorfní a krystalickou strukturou je částečně destruktivní, proto mazání s následným zápisem je možné opakovat jen asi 1000×.

Poznámka

Odrazivost různých typů médií se velmi liší. u CD-ROM je to 80 %, u CD-RW pouze 25 %, CD-R jsou někde mezi. Kvalita vypálení závisí na kvalitě média, vypalovací mechaniky a také rychlosti při vypalování. Obecně platí, že vyšší rychlosti znamenají vyšší pravděpodobnost výskytu chyb (laser pracuje s vyšším výkonem, aby při vyšší rychlosti dokázal zahřát místa na disku na stanovenou teplotu).

7.5.2 DVD

DVD (Digital Versatile Disc nebo Digital Video Disc) má mnohem větší kapacitu než CD. Kromě toho se používají menší pity a odstup stop je mnohem menší (to vše je hlavním důvodem navýšení kapacity), přičemž samotná spirála je stejně dlouhá a stejně široká jako u CD. Další navýšení kapacity je možné ukládáním dat do dvou vrstev a také využitím obou povrchů média.

Rozměry DVD jsou z důvodu kompatibility stejné jako u CD (včetně středového otvoru), ale odlišná je vnitřní struktura. Nosná vrstva je poloviční, obvykle jsou ve skutečnosti dvě (aby se zachovala tloušťka média), mezi nimi jedna nebo dvě datové vrstvy a jednostranná nebo oboustranná reflexní vrstva.

Při *čtení* se používá laserový paprsek s kratší vlnovou délkou než u CD (660 nm – červené světlo), jiná je i vzdálenost, na kterou se tento paprsek ostří. Srovnání vidíme na obrázku 7.17.

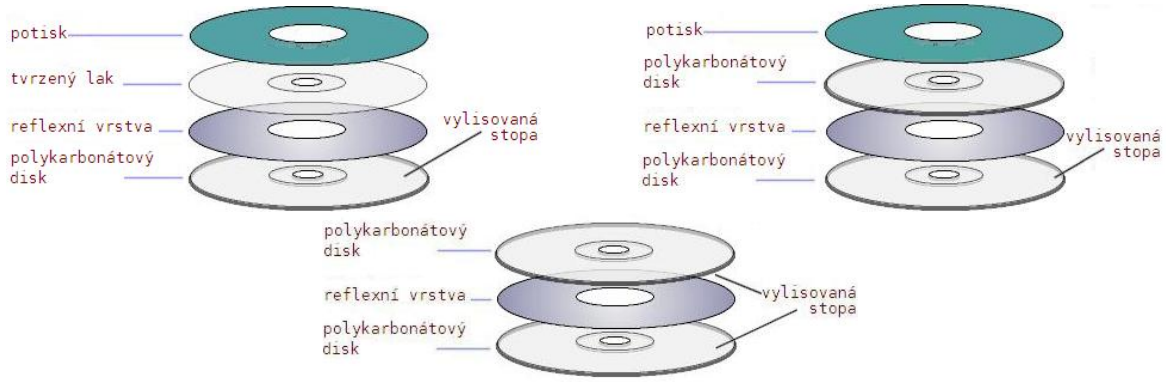
Standardy jsou rozděleny do dvou skupin se vzájemnou nekompatibilitou. První skupina standardů byla vydána uskupením *DVD Forum* (minusové standardy). Tyto standardy se setkaly s kritikou některých firem zejména z důvodu nevyhovujících licenčních podmínek a ceny, proto byly další (plusové) standardy vydány uskupením *DVD+RW Alliance*.

- DVD-ROM
- DVD-R, DVD-RW
- DVD-RAM – místo spirály má spoustředné stopy, princip (ne technologie) podobný MO diskům, lepší detekce a korekce chyb
- DVD+R, DVD+RW (+RW je starší než +R)

Liší se použitými materiály, chybovostí (+ jsou mírně lepší, ale mají mírně menší kapacitu), původní cenou licence. Současné DVD mechaniky obvykle podporují standardy z obou skupin (většinou s výjimkou DVD-RAM). Dvouvrstvá média se také označují přidáním písmen „DL“, například DVD+R DL je dvouvrstvé DVD+R. DVD+/-RW lze použít i pro paketový zápis – pokud máme příslušný software, dá se s nimi zacházet jako s USB flash disky.

Poznámka

Jaký je tedy rozdíl mezi DVD-RAM a DVD+/-RW s paketovým zápisem? Obojí je přepisovatelné, obojí umožňuje s optickým médiem pracovat podobně jako třeba s pevným diskem či výměnným médiem. Pro DVD-RAM musíme mít speciální optickou mechaniku, která tento typ zápisu podporuje, a zapisuje se do soustředných kružnic (jako na magnetickém disku). Pro DVD+/-RW s paketovým zápisem stačí běžná optická mechanika, ale musíme mít speciální software, data se zapisují na spirálu.



Obrázek 7.17: Srovnání – CD, jednovrstvé DVD, dvouvrstvé DVD[13]

Varianty DVD najdeme v tabulce 7.3 včetně rozměrů, počtu stran a vrstev, a kapacity. Nejčastěji se setkáváme s DVD-5 a DVD-9.

Označení	Průměr	Stran	Vrstev/str	Kapacita
DVD-1	8 cm	1	1	1,46 GB
DVD-2	8 cm	1	2	2,66 GB
DVD-3	8 cm	2	1	2,92 GB (2×1,46)
DVD-4	8 cm	2	2	5,32 GB (2×2,66)
DVD-5	12 cm	1	1	4,70 GB
DVD-9	12 cm	1	2	8,54 GB
DVD-10	12 cm	2	1	9,40 GB (2×4,70)
DVD-14	12 cm	2	2/1	13,24 GB
DVD-18	12 cm	2	2	17,08 GB (2×8,54)

Tabulka 7.3: Varianty DVD

Vlastnosti DVD se odvozují především od požadavků filmového průmyslu. Je možné použít *DRM* (Digital Rights Management) pro zamezení kopírování média: na disk lze uložit copyright, který se při

[13] Zdroj: http://www.root.cz/clanky/nasledovnici-kompaktnich-disku-dvd/

kopírování nepřenáší, a podle toho se rozpoznávají nelegální kopie.

Použitelnost DVD lze určit pro jeden nebo několik *regionů*. Těchto regionů je stanoveno celkem 8, z toho 6 pro oblasti na Zemi (například Evropa je v regionu číslo 2, USA a Kanada mají region číslo 1), region 7 není stanoven a 8 je určen pro pohybující se oblasti (letadla, lodě apod.). DVD určené tímto způsobem pro některý region lze přehrávat pouze na přehrávači s tímto regionem nastaveným.

Přenosové rychlosti se stejně jako u CD udávají jako násobek základní rychlosti, tato základní rychlost je však odlišná. Označení 1× odpovídá u DVD rychlosti 1350 kB/s, což je rychlost 9× pro CD.

7.5.3 Struktura přepisovatelného disku

Struktura přepisovatelného disku (včetně CD/DVD-R) je dána použitou technologií vypalování. Obvykle tam najdeme několik *typů zón*:

- *vnitřní zóna* většinou obsahuje tabulku obsahu disku (TOC), administrační informace (například počet stop a umístění jejich začátku a konce), testovací a kalibrační informace,
- *Lead-In* – bezpečnostní a administrační zóna pro kontrolní data, identifikátor apod., je to zóna s informacemi o uložených souborech,
- *Lead-Out* – upozornění na konec disku,
- *vnější zóna* – administrační, testovací a kalibrační informace.

Kalibrační informace slouží ke kalibraci laseru pro zápis.

Metody vypalování CD a DVD:

1. *DAO* (Disc-at-Once) – celý disk se vypálí najednou bez průběžného vypínání laseru, používá se například u hudebních CD. Disk je rozdělen na pět zón:

vnitřní zóna	Lead-in	data	Lead-out	vnější zóna

2. *TAO* (Track-at-Once) – najednou se vypálí jedna stopa, pak je laser vypnut, je možné kdykoliv přidat další stopy. Disk má strukturu podobnou předchozí, ale za Lead-out následuje prázdný prostor pro zápis dalších stop (každá stopa určená pro data obsahuje zóny Lead-in, data a Lead-out).

	vnitřní zóna		
1. stopa:	Lead-in	data první stopy	Lead-out
2. stopa:	Lead-in	data druhé stopy	Lead-out
atd.		⋮	
	vnější zóna		

3. *MultiSession* – Není nutné zapisovat najednou celý disk, najednou se zapisuje jen jediná session (relace) na jednu nebo více stop. Zóna Lead-in je jen jedna (vytvoří se při prvním vypalování), zóna Lead-out také jen jedna (vytvoří se při posledním vypalování). TOC je vytvořen až při posledním vypalování zároveň s Lead-out.

   Při ukončení vypálení takové session, která ještě nemá být na disku poslední, se místo Lead-out vloží za data *blok uzavření*, na začátku vypalování následující session se začíná vložením *bloku otevření* a až pak následují data. Za daty poslední vypálené session se již místo bloku uzavření vloží zóna Lead-out. Níže vidíme strukturu disku vypáleného v režimu MultiSession (již celý zaplněný, včetně závěrečných sekcí). Zapisované úseky dat mohou mít samozřejmě různou délku.

vnitřní zóna		
Lead-in	$data_1$	blok uzavření
blok otevření	$data_2$	blok uzavření
blok otevření	$data_3$	blok uzavření
	⋮	
blok otevření	$data_n$	**Lead-out**
vnější zóna		

4. *Incremental Recording* (u DVD) – podobné jako MultiSession, ale bez uzavíracích a otevíracích zón. Na začátku prvního zápisu se vytvoří zóna Lead-in, pak se disk postupně zaplňuje a na konci posledního zápisu se vytvoří zóna Lead-out. Dokud není tato poslední oblast vytvořena, není disk čitelný v mechanikách nepodporujících tuto metodu, po jejím vytvoření je struktura disku podobná disku s metodou DAO.
5. *Sequential Recording* – podobné jako Incremental Recording, ale z důvodu lepší kompatibility s jinými mechanikami se za každým zápisem dat vytvoří dočasná zóna Lead-out, při dalším zápisu na disk se tato zóna přepíše daty (a vytvoří se nová dočasná zóna Lead-out).
6. *Mount Rainier* (paketový zápis) – není kompatibilní s ostatními způsoby zápisu, tyto disky lze zapisovat pouze ve vypalovačce označené „Easy Write“. Číst je lze sice i v nekompatibilních mechanikách, ale jen pomocí přídavného softwaru (například EasyWrite Reader).

   Data jsou vypalovačce posílána v paketech o velikosti 2 kB. Další důležitou vlastností je zavedení řízení chyb. Součástí disku je kromě jiného také oblast DMA (Deffect Managed Area). Pokud je nalezen vadný sektor, je přemapován sektorem z DMA.
7. *Random Recording* – jedná se o metodu velmi podobnou metodám pro pevné disky, data mohou být zapisována kdykoliv a kamkoliv.

✎ **Pojmenování a organizace souborů:** Při vypalování bychom také měli nastavit standard pro pojmenování souborů a adresářů. Nejstarší používaný standard pro CD je *ISO 9660*, který stanoví osm znaků pro název a tři znaky pro příponu souboru. Protože se dnes běžně používají dlouhé názvy souborů a také diakritika, tento standard obvykle nevolíme.

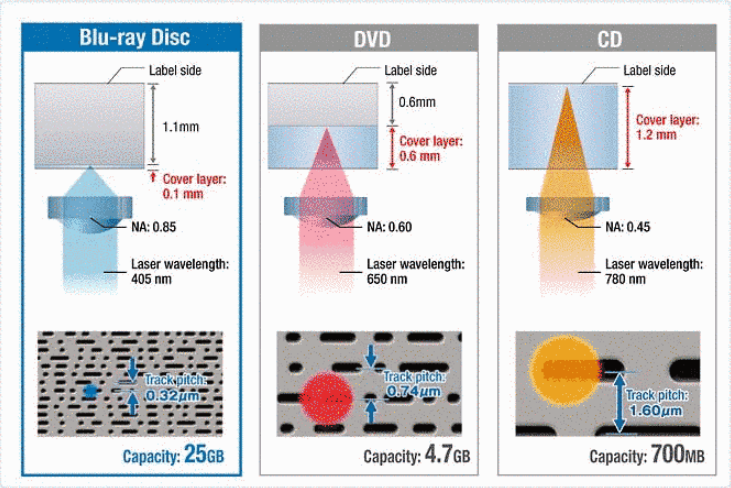
Hodně používaný je *ISO 9660 Level 2* (taktéž pro CD), který rozšiřuje prostor pro názvy souborů na 30. Dále se setkáme se standardy *Joliet* a *Romeo* (v překladu Julie a Romeo). Joliet ukládá názvy souborů ve dvou tvarech najednou – dlouhý název převzatý z operačního systému a pak ještě zkrácený název čitelný v MS DOSu. Romeo ukládá pouze dlouhý název (128 znaků).

Při vypalování DVD se používají jiné standardy (souborové systémy). Nejpopulárnější je *UDF* (používá se pro DVD, ale lze jej využít i pro CD). Podporuje dlouhé názvy souborů. V UNIXových systémech včetně Linuxu je běžně čitelný, ve Windows od verze 2000.

Na DVD je také možné používat souborový systém *MRW*. Je novější než UDF a má několik nových vlastností, z nichž asi nejzajímavější je vylepšená správa chyb.

7.5.4 Blu-Ray

✎ Blu-Ray disky mají opět z důvodu zpětné kompatibility stejné rozměry jako CD/DVD včetně tloušťky 1,2 mm. Data se ukládají na spirále, laserový paprsek má vlnovou délku 405 nm (modré světlo, i když fyzikálně se jedná spíše o fialové). Díky této vlnové délce a menší vzdálenosti hlavy od povrchu bylo možné zvýšit kapacitu disku – jednovrstvé Blu-Ray o průměru 12 cm má kapacitu 25 GB, dvouvrstvé až 50 GB.



Obrázek 7.18: Srovnání CD, DVD, Blu-Ray[14]

Varianty Blu-Ray:

- BD-ROM – pouze pro čtení, lisovaný ve výrobě,
- BD-R – jednorázový zápis,
- BD-RE – přepisovatelný,
- BD-AV – pro domácí amatérská videa,
- BDMV (Blu-Ray Disk Movie) – pro profesionální video jako varianta BD-ROM,
- BD-J (Java) – interaktivní Blu-Ray využívající Javu,
- Mini-BD – menší rozměr (8 cm).

Přidáním „DL“ se označuje dvouvrstvé médium, přidání BD+ (pro BD-ROM) znamená využití níže popsané hardwarové ochrany.

Video může být kódováno několika různými algoritmy: MPEG-2 (ten zvládlo i DVD), H.264/ MPEG AVC, VC-1.

Pro ochranu obsahu (DRM) se u Blu-Ray (a také u HD DVD) původně využíval systém *AACS* (Advanced Access Content System), který však byl prolomen. Proto byla přidána nová hardwarová ochrana *BD+ klíč* od firmy Cryptography Research, jedná se o součástku, která musí být výrobcem přidána do mechaniky (příp. stolního přehrávače).

7.5.5 Životnost a údržba optických médií

Co škodí?

- světlo, UV záření,
- vyšší teploty (ideální je kolem 25 °C nebo nižší),
- velká vlhkost (oxidace reflexní vrstvy), může se dovnitř dostat z boční části,
- ohýbání, lámání, atd.,
- kouř (také z cigaret).

Další vlivy: použitý materiál reflexní vrstvy (nejlepší je zlato), použité organické barvivo u -R varianty, způsob výroby nosné vrstvy, přesnost tvaru disku (středová část), kvalita vypalovačky a jejího firmwaru, vhodná rychlost zápisu.

[14] Zdroj: http://www.root.cz/clanky/vyvoj-optickych-pameti-od-dvd-k-blu-ray/

Po určité době (zvláště v rizikovějším – například prašném či zakouřeném – prostředí) se optická mechanika zanese. Projevuje se to chybami při čtení a zápisu, nemožností přečíst některá CD či DVD apod.

Pokud se setkáme s těmito příznaky, pak bychom měli mechaniku co nejdřív vyčistit. Sice se prodávají čisticí sady, ale jejich účinnost není zrovna nejlepší. Proto je vhodnější použít tento postup:

- mechaniku vymontujeme z počítače, odpojíme všechny kabely,
- rozšroubujeme, najdeme čočku (je malá, má průměr menší než centimetr),
- *opatrně* vyčistíme čočku vatou namotanou na špejli (nebo lépe využijeme vatové tyčinky) namočenou v *izopropylalkoholu* (dá se koupit v drogeriích),
- necháme odpařit, zašroubujeme, připojíme, vyzkoušíme.

Poznámka

Rozhodně nepoužíváme přípravky na čištění oken! Tak bychom čočku mohli poškodit, přinejlepším by na ní ulpěla vrstva, která by zkreslovala čtení i zápis. Pokud nemáme izopropylalkohol, použijeme jen suchou vatu, ale to funguje jen proti lehkému naprášení. Při čištění je třeba dávat velký pozor, nesmíme přitlačit, čočka je obklopena jemnými součástkami, které by se mohly poškodit.

Další informace

Postupy čištění optické mechaniky a seřízení optiky najdeme například na http://www.bastleni.eu/vypocetni-technika/51-opticka-mechanika.

7.6 Flash paměti

Flash paměti se v počítačích ve skutečnosti využívají již velmi dlouho, a to jak u vnitřních, tak i u vnějších pamětí. U vnitřních jsme se setkali například s využitím pro uložení kódu BIOSu.

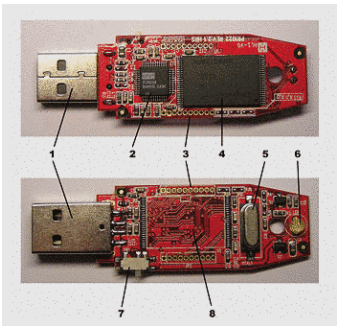
Jde o čistě elektronické řešení (nepoužíváme laser ani magnetismus). Zatímco u vnitřních flash pamětí jde většinou o čipy s obvody typu NOR, u vnějších pamětí se používají obvody typu NAND.

7.6.1 USB flash disk

USB flash disk je vpodstatě klíčenka vybavená pamětí typu flash s rozhraním USB. Toto řešení je komerčně dostupné od roku 2000.

USB flash disk se skládá z

1. paměťového čipu NAND
2. Mass Storage Controller – řadič
3. konektor USB (většinou USB-A male)
4. krystalový oscilátor pro hodinový signál
5. další – LED dioda, obal, krytka, příp. zámek pro blokaci zápisu



1. USB konektor
2. Mass storage controller
3. Testovací kontakty
4. Flash paměť
5. Krystalový oscilátor
6. LED
7. Zámek
8. Místo pro druhou flash

Obrázek 7.19: Struktura USB flash disku[15]

Pro USB flash disky je typické to, co známe i od jiných typů pamětí – zápis probíhá *v blocích* a je částečně destruktivní (počet přepisů paměťové buňky je omezen).

[15] Zdroj: http://cs.wikipedia.org/wiki/USB_flash_disk

Poznámka

Jak bylo výše uvedeno, flash paměti mají omezený počet zápisů do paměťových buněk. Z toho důvodu není vhodné na tomto typu média používat souborový systém NTFS, který je optimalizovaný spíše pro klasické pevné disky (má vysokou režii, ukládá velmi mnoho informací a navíc hodně často). Proto se na USB flash discích setkáváme spíše se souborovými systémy, které jsou v tomto směru šetrnější – FAT32, exFAT a na Linuxu ext2fs.

7.6.2 SSD

SSD (Solid State Drive) je alternativa k běžnému pevnému disku. Třebaže se o SSD často mluví jako o „disku“, ve skutečnosti uvnitř neobsahuje nic, co by mělo diskovitý tvar.

Oproti klasickým pevným diskům se vyznačuje (relativně) nižší spotřebou energie (záleží na způsobu využívání), nízkou hmotností, malými rozměry, tichým chodem, vysokou spolehlivostí (žádné pohyblivé části, nic se mechanicky neopotřebovává), a obecně vyššími přenosovými rychlostmi.

Vzhledem k vyšší ceně flash čipů se v noteboocích setkáváme s SSD s obecně nižší kapacitou než u klasických disků. Také je horší poměr cena/kapacita.

Z mnoha testů vyplývá, že SSD disky mají nižší spotřebu než HD. Ovšem tuto spotřebu si udržují po celou dobu, i když zrovna nejsou používány, kdežto HD spotřebovávají energii jen v době, kdy rotují (mnoho uživatelů na desktopu má nastaveno automatické zastavování disků při delší době nečinnosti). Proto se spotřeba SSD a HD špatně srovnává, záleží na skutečném využívání disku.

SSD je možné získat v těchto formách:

- v pouzdře o rozměrech 2,5” pevného disku, pak se také zapojuje jako běžný pevný disk (kabel SATA II nebo SATA 3.0, a také samozřejmě napájecí kabel),
- v pouzdře do SATA Express (SATAe),
- v pouzdře 2,5” s rozhraním U.2 (SFF-8639) – podobný SATAe, ale jinak využívá jednotlivé piny,
- úzká rozšiřující karta do slotu mSATA nebo M.2 na základní desce,
- rozšiřující karta do PCIe.

Rozhraní M.2, SATAe, U.2 a PCIe znamená možnost používat místo ATAPI technologii NVMe, o které již v těchto skriptech několikrát byla řeč. NVMe je optimalizováno pro komunikaci přes PCIe, propustnost je několikanásobně vyšší než při použití ATAPI (samozřejmě záleží na tom, jakou verzi PCIe máme, uváděné výsledky předpokládají PCIe 3.0).

U nejnovějších operačních systémů se předpokládá schopnost detekce typu disku (Windows 7 už dokážou detekovat SSD a pro novější jádra Linuxu to také není problém).

Paměťová buňka v SSD může být jednoho ze těchto typů:

- *SLC* (Single Level Cell) – rozpoznává 2 úrovně napětí, které jsou interpretovány jako 0 nebo 1, do SLC buňky lze tedy uložit 1 bit,
- *MLC* (Multi Level Cell) – rozpoznává 4 úrovně napětí, tedy je možné do ní uložit dva bity (hodnoty 00, 01, 10, 11),
- *TLC* (Triple Level Cell) – rozpozná 8 úrovní napětí, do paměťové buňky ukládáme tři bity,
- *QLC* (Quad Level Cell) – rozpoznává 16 úrovní napětí, do paměťové buňky ukládáme čtyři bity.

Poznámka

SSD se SLC buňkami jsou dražší (horší poměr cena/kapacita), ale pomaleji se opotřebovávají, déle vydrží.

Zatímco SLC buňky fyzicky přežijí asi 100 000 zápisů, MLC buňky pro dva bity cca 2000 zápisů, TLC cca 1000 zápisů. Ovšem pozor, nejedná se o tak výrazné omezení, jak by se z těchto čísel mohlo zdát. K tomuto tématu se vrátíme o něco dále.

Paměťové buňky (flash) jsou organizovány *v blocích*, každý blok obsahuje stanovené množství stránek (o velikosti několika kB, obvykle 4 kB); operační systém obvykle pracuje s jednotlivými stránkami, ale flash paměť dokáže pracovat jen s celým blokem najednou (to platí obecně pro všechny flash paměti).

Mechanismus zápisu: Na rozdíl od HDD má řadič disku SSD mnohem více práce. Jedním z jeho úkolů je co nejvíce eliminovat škody způsobené mnohočetným zapisováním do paměťových buněk. Zápis probíhá takto:

- jak bylo výše napsáno, pracuje se s celými bloky, proto si řadič vypomáhá vyrovnávací pamětí (bufferem), jehož velikost odpovídá velikosti bloku,
- řadič do bufferu přenese nejdřív celý blok, do kterého má provést změny, v bufferu pak provede žádné změny, tedy v bufferu se utvoří blok v tom tvaru, v jakém má být po ukončení zápisu,
- blok, do kterého se má obsah bufferu uložit, se celý smaže (flash paměť dokáže zapisovat jen do takového bloku, který byl smazán),
- pak teprve dojde k zápisu bloku (vcelku) na místo.

Mazání není nutné, pokud daný blok ještě nebyl používán. Proto SSD disk ze začátku vykazuje celkem slušné rychlosti zápisu, ale postupem času (jak se zaplňuje) se zápis stává pomalejší (operace mazání bloku je časově poměrně náročná).

Proti tomuto zpomalení se dá bojovat dvěma způsoby:

1. SSD disk je ve skutečnosti vybaven větším množstvím paměti než dává na vědomí při detekci, místo navíc je používáno kromě jiného i k „nastavení“ počtu smazaných bloků.
2. Firmware SSD disku může podporovat (a dnes také obvykle podporuje) příkaz *ATA TRIM*, kterým operační systém informuje řadič disku, že určité bloky v paměti mají být smazány, operace mazání se tedy může provést nezávisle na zápisu nových dat. Ovšem tento ATA příkaz musí být podporován i na straně operačního systému.

U HD disků není nutné používat příkaz TRIM, protože současné souborové systémy provádějí jen „lehké“ mazání, které při smazání souboru pouze smaže odkaz na tento soubor a skutečné místo na disku, které soubor zabíral, zůstává nedotčeno, ovšem před opětovným zápisem se nemusí mazat. U SSD se však jedná o důležitou vlastnost, která může zabránit zbytečnému zpomalování zápisu.

Poznámka

Příkaz TRIM je podporován ve Windows až od verze 7, v Linuxu se podpora TRIM objevila už v první polovině roku 2009. Dnes už většina SSD ve svém firmwaru tento příkaz podporuje, většina výrobců také chystá update starších firmwarů s dodáním tohoto příkazu.

Životnost SSD. Paměťové buňky typu SLC mají životnost cca 100 000 zápisových operací, MLC jen 3000 zápisů a TLC pouze 1000 zápisů. Což by znamenalo, že například SSD s flash čipem obsahujícím MLC buňky lze 3000× přepsat (kompletně celý, vynásobte kapacitou SSD), než budou všechny buňky nepoužitelné. Řešení problému postupného opotřebení paměťových buněk (resp. celých bloků) při zápisu:

1. Je třeba dát na vědomí operačnímu systému, že jde o SSD a nikoliv o klasický pevný disk, operační systém pak provede určitá nastavení, kterými se sníží počet zápisů.

2. Oproti udané kapacitě SSD je skutečná kapacita flash čipu větší. Paměťové oblasti, které jsou „navíc“, slouží kromě jiného k postupnému nahrazování těch používaných bloků, které se již přiblížily hranici své životnosti.
3. Aby byly paměťové buňky využívány co nejrovnoměrněji, firmware SSD používá souhrn algoritmů nazývaných *Wear Levelling*.

Nejdřív se podíváme na metodu *Wear Levelling* (v překladu něco jako „vyrovnání opotřebení“). Tato metoda spočívá v tom, že u každho paměťového bloku se zaznamenává počet zápisů:

- bloky s menším počtem zápisů jsou preferovány při dalších operacích zápisu (*dynamický Wear Levelling*, porovnávání probíhá při každém zápisu),
- bloky obsahující často aktualizovaná data (například log soubory) se čas od času přesunou na jiné místo na disku do dosud méně frekventovaných oblastí (*statický Wear Levelling*).

Opotřebení může redukovat i operační systém, pokud v tomto smyslu dokáže zacházet s SSD diskem. Z hlediska operačního systému (a také procesů) jsou pro SSD nebezpečené některé jinak běžné a užitečné operace:

- defragmentace nejenže nemá u SSD smysl (logické řazení buněk je zcela jiné než fyzické), ale naopak nadbytečné stěhování dat SSD škodí,
- Ready Boost (nastavení vyrovnávací paměti disku flash pamětí) nemá smysl, protože SSD je flash paměť, zrychlení je nulové a dokonce zbytečné kopírování na přídavnou flash paměť může systém mírně zpomalit,
- Prefetch a Superfetch jsou opět zbytečné,
- Volume Shadow Copy slouží k urychlení zálohování systémových dat (ukládá na disk do zvláštní oblasti systémové soubory, aby mohly být zálohovány bez nebezpečí ztráty integrity na straně originálních souborů), ale pokud tato data soustavně nezálohujeme, je tato služba bez významu (navíc zbytečně zabírá místo na disku),
- aplikace pro úplné mazání dat (likvidaci všech stop) fungují tak, že mazané místo mnohokrát přepisují náhodnými daty; jenže na SSD se ve skutečnosti nezapisuje opakovaně na totéž místo, takže jen zbytečně snižujeme životnost disku,
- ukládá se mnoho dat do log souborů, používají se dočasné soubory, cookies, atd., také u mnoha aplikací včetně internetových prohlížečů.

Poznámka

Starší operační systémy nedokážou detekovat SSD (poznají, že jde o paměťové médium, ale ne, že jde o SSD), a proto s tímto typem disku zacházejí stejně jako s běžným magnetickým diskem. Proto bychom měli potřebná nastavení provést sami, například vypnout službu defragmentace, Prefetch, Superfetch, Ready Boost, Volume Shadow Copy, a také omezit používání dočasných souborů a dalších podobných dat.

Další informace

Rady jak zacházet s SSD diskem najdeme na internetu, zajímavé jsou například tipy na
http://tombuntu.com/index.php/2008/09/04/four-tweaks-for-using-linux-with-solid-state-drives/.

Další informace o SSD:
http://www.svethardware.cz/art_doc-90BC248F94714B88C12575B4003F5730.html

Také u SSD se používá technologie NCQ, ale funguje trochu jinak. Také se používá fronta (odložených) požadavků (přenášená data, ale také potvrzení po vyplnění požadavku na data), a to oficiálně pro vyrovnání latencí mezi SSD a systémem (vyrovnání rozdílů v rychlosti komunikace). Ve výkonnostních testech se však přínos NCQ u SSD moc neprojevuje.

7.6.3 Hybridní disky

Hybridní disk (SSHD) kombinuje v jednom pouzdře plotny běžného pevného disku a flash paměť používanou v SSD. Účelem je zachovat vysokou kapacitu běžných disků a rychlost SSD. Ve flash čipu jsou uložena nejčastěji používaná data (ale pozor, záleží na firmwaru, jak konkrétně bude flash čip využíván), na plotnách pevného disku jsou uložena všechna data (tj. flash paměť zde funguje jako obdoba rychlé vyrovnávací paměti, i když vyrovnávací paměť je zde také).

Na trhu existuje také varianta, kdy flash čip je v systému viditelný jako samostatný oddíl (tj. celý SSHD je vidět jako disk se dvěma oddíly). Pak má smysl nainstalovat na „flash oddíl“ systém a oddíl z pevného disku používat na data.

Flash čip mívá jen malou kapacitu (obvykle několik GB), proto nárůst ceny proti běžným pevným diskům není velký. Obvyklé datové rozhraní u hybridních disků je SATA II nebo SATA 3.0.

7.6.4 Paměťové karty

Paměťové karty jsou také flash paměti. První karty (PCMCIA karty) používaly obvody NOR, současné flash paměti včetně karet již využívají NAND obvody, stejně jako USB flash disky a SSD.

Flash paměť je uzavřena v tenkém plastovém pouzdře (na destičce je především NAND flash čip a řadič). Kontakty nejsou zakryté, což negativně ovlivňuje životnost (kontakty se postupně opotřebovávají a navíc jsou vystaveny vnějším vlivům).

Paměťových karet existuje více druhů. Jsou navzájem nekompatibilní a liší se i svými rozměry. Nekompabitility se dají vyřešit koupením redukce.

Paměťové karty se většinou používají v přenosných zařízeních (smartphony, PDA, notebooky, digitální fotoaparáty, videokamery, apod.), ale lze koupit i čtečku pro desktop.

SD (Secure Digital) měla kapacitu původně omezenou na 2 GB. Novější forma, *SDHC* (také SD 2.0), má již vyšší kapacitu (několik desítek GB, do 32 GB), jen ji nelze používat ve starších čtečkách pro SD. V současné době se můžeme setkat i s další generací – SDXC (Secure Digital eXtended Capacity) má horní hranici 2 TB. Opět je třeba mít čtečku podporující tento formát.

Existují také rozměrově menší varianty microSD (často slouží k rozšíření paměti smartphonů) a miniSD.

SD a SDHC karty komunikují přes sběrnici SPI nebo QSPI (SPI přenáší data po jednom bitu, QSPI po čtyřech bitech). Co se rychlosti týče, pro SDHC karty (operace zápisu) existuje několik *rychlostních tříd* (Class Rating):

- Class 2 představuje minimální rychlost zápisu 2 MB/s,
- Class 4 znamená minimálně 4 MB/s,
- Class 6 znamená minimálně 6 MB/s,
- Class 10 znamená minimálně 10 MB/s.

Rychlostní třída by měla být uvedena na kartě pod logem. Obvykle s dodržením těchto rychlostí nebývá problém, SDHC karty mívají v reálu dokonce vyšší než udávanou rychlost.

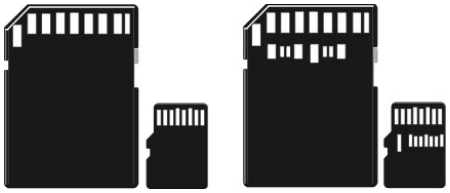
Tyto rychlosti postupně přestávají stačit (zejména tehdy, když chceme na kartu nahrávat video ve vyšším rozlišení), proto pro SDHC a SDXC karty vznikla nová sběrnice *UHS* (Ultra High Speed) s dalšími rychlostními třídami:

- specifikace sběrnice UHS-I nabízí rychlostní třídy U1 a U3,
- specifikace sběrnice UHS-II nabízí rychlostní třídu U3.

Třída U1 znamená rychlost minimálně 10 MB/s (stejně jako Class 10), třída U3 znamená rychlost minimálně 30 MB/s.

Zatímco specifikace UHS-I mění především komunikační sběrnici (piny na kartě defacto zůstávají tak jak jsou), specifikace UHS-II přidává nové piny, které jsou na kartě umístěny v druhé řadě pod původními – viz obrázek 7.20.

Jak je to s kompatibilitou: platí zpětná kompatibilita, ale s omezeními. V starší čtečce karet bude fungovat novější karta, ale na „starší“ rychlosti. Podobně v novější čtečce bude fungovat starší karta, ale opět na „starší“ rychlosti. U karty UHS-II se v čtečce podle staršího standardu nebude používat druhá řada pinů.

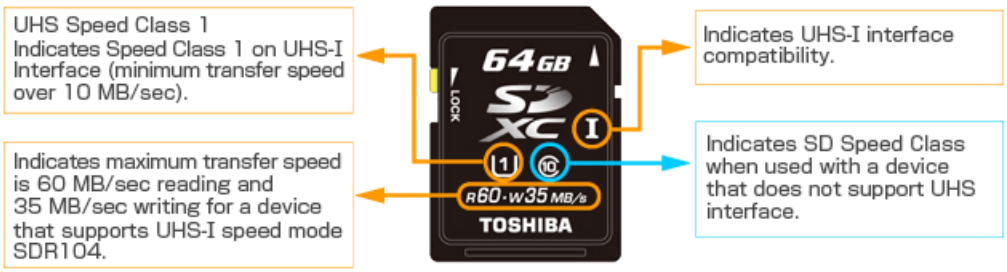


Obrázek 7.20: Vlevo karty pro sběrnici UHS-I, vpravo pro UHS-II[16]

Poznámka

Pokud potřebujete na kartu ukládat natáčené video v rozlišení FullHD, pak jakž takž stačí rychlost 10 MB/s (Class 10 nebo lépe U1). Pro vyšší rozlišení už je nutná třída U3.

Důležité je vědět, jak na kartě poznat rychlostní třídu. Původní rychlostní třídy se značí číslem uzavřeným v písmenu „C“, UHS třídy jsou označeny číslem v písmenu „U“ a taky tam najdeme označení specifikace („I“ nebo „II“), přičemž na kartě mohou být oba typy tříd (tj. pokud se karta vloží do čtečky zvládající UHS, použije se třída pro UHS, ale když ji dáme do starší čtečky, která „neumí“ sběrnici UHS, použije se klasická třída).



Obrázek 7.21: Označení rychlostní třídy na paměťové kartě[17]

Na obrázku 7.21 je paměťová karta SDXC o kapacitě 64 GB s oběma označeními tříd (U1 a C10), s označením specifikace UHS-I. Obě třídy určují minimální rychlost zápisu 10 MB/s, což ale neznamená, že by karta nedokázala komunikovat rychleji (sběrnice to umožňuje, záleží taky na řadiči). Pod rychlostními třídami je dokonce napsáno, že karta dokáže číst (R) rychlostí až 60 MB/s, zapisovat (W) rychlostí až 35 MB/s v případě, že je v čtečce podporující UHS-I.

[16] Zdroj: http://electronicdesign.com/memory/whats-difference-between-sd-and-uhs-ii-memory-cards
[17] Zdroj: http://www.sdxc2.com/

Další informace

Detailní porovnání různých paměťových karet: http://en.wikipedia.org/wiki/Comparison_of_memory_cards

7.7 Zálohování a archivace

Zálohování (backup) je uchovávání dat, informací a vědomostí (obvykle pravidelné – denní, týdenní, měsíční) za účelem jejich obnovy v případě poškození dat na původním umístění. Provádí se na přepisovatelné médium (páska, magnetooptický disk, optický disk). Cílem je co nejrychlejší záchrana dat po havárii, ochrana provozuschopnosti informační soustavy, apod.

Typy záloh:

1. *výchozí záloha systému* – slouží k rychlé obnově stavu systému v případě jeho poškození, zálohuje se *hned po instalaci* systému a programového vybavení,
2. *úplná záloha* – podobně jako výchozí, ale provádí se pravidelně a zálohuje se vše stanovené,
3. *inkrementální* – jednou se provede úplná záloha, pak se pravidelně provádějí inkrementální zálohy obsahující pouze ty soubory, které byly změněny od posledního zálohování (úplného nebo inkrementálního); když obnovujeme z inkrementální zálohy, musíme projít všechny zálohy od začátku (prvotní úplnou a pak aplikujeme jednotlivé inkrementální),
4. *rozdílová* – také je jednou provedena úplná záloha, a dále je pravidelně vytvářena rozdílová záloha vzhledem k té úplné, tj. v rozdílové jsou všechny soubory, které byly vytvořeny nebo změněny od doby, kdy byla vytvořena úplná záloha (jedna rozdílová obsahuje vše, co by jinak bylo v celém souhrnu inkrementálních záloh); obnova dat z rozdílové zálohy je rychlejší, bereme nejdřív úplnou zálohu a pak poslední provedenou rozdílovou.

Archivace je uchování dat na obvykle velmi dlouhou dobu (příp. stanovená zákonná lhůta) za účelem jejich případného dalšího využití (například mobilní operátoři musí uchovávat základní informace o hovorech svých klientů). Provádí se na přepisovatelné nebo nepřepisovatelné médium, obvykle se nepočítá s jejich aktualizací. Cílem je dlouhodobá úschova dat, uvolnění prostředků pro jiné účely, atd.

Pravidlo 3-2-1 určuje, jak by mělo zálohování vypadat: uděláme nejméně 3 kopie, nejméně na 2 rozdílných zařízeních, alespoň 1 kopie mimo domov/práci (pole toho, kde zálohujeme).

7.7.1 Magnetická páska

Základem magnetické pásky je pásek z fólie (polyesterová nebo z podobného materiálu s chemicky výhodnějšími vlastnostmi) dlouhá desítky až stovky metrů, na ni (z jedné nebo obou stran) je nanesena tenká magnetická vrstva (o tloušťce v mikrometrech). Páska je navinuta na cívku, může být zároveň s druhou cívkou uzavřena v kazetě.

Na rozdíl od většiny ostatních paměťových médií se při čtení/zápisu pohybuje páska, nikoliv hlavy. Výhodou je velmi příznivý poměr cena/výkon a dlouhá životnost, nevýhodou sekvenční přístup.

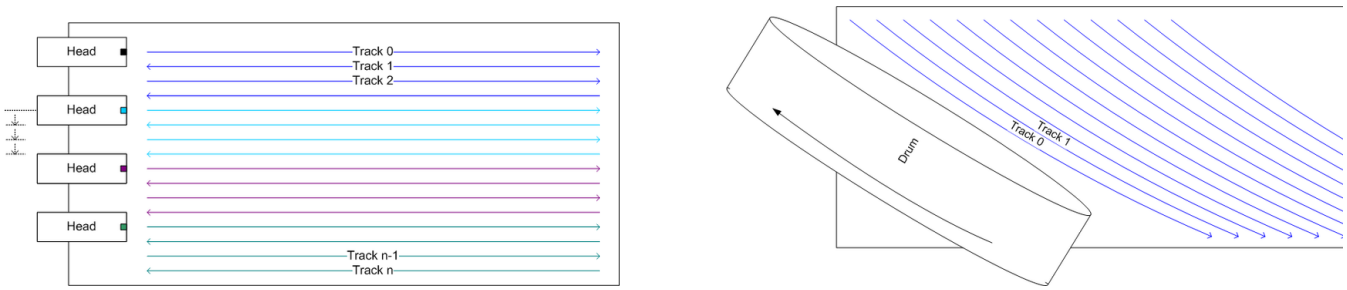
Dříve byly magnetické pásky běžné u sálových počítačů, dnes se používají jako zálohovací médium (vysoká kapacita, při simultánním záznamu vysoká rychlost). U zálohovacích pásek se používají pásky *vícestopé* (8 stop nebo násobek 8, případně plus paritní stopy).

7.7.2 Pásky momentálně používané pro zálohování

Nosnou vrstvou pásky je pružný plast, na kterém je nanesena magnetická vrstva. Obvyklé datové rozhraní je SAS, někdy RJ-45, SATA nebo USB. V současné době se setkáváme s páskami využívajícími jeden z těchto způsobů záznamu:

1. lineární serpentinový zápis,
2. spirálový záznam (Helical).

Srovnání principu těchto typů záznamu včetně role čtecích/zápisových hlav vidíme na obrázku 7.22.



Obrázek 7.22: Srovnání lineárního serpentinového (vlevo) a spirálového (vpravo) záznamu[18]

Na rozdíl od optických médií se magnetické pásky vyznačují velmi dobrou životností – podle konkrétního materiálu, provedení a technologie záznamu i v desítkách let. Kapacita se u jednotlivých technologií značně liší, i levnější typy mívají běžně kapacitu v desítkách GB. U kvalitnějších pásek bývá běžná hardwarová komprese.

DAT pásky jsou levné (v přepočtu na jednotku uložené informace). Materiál je podobný tomu, co dřív bylo používáno u video- a zvukových kazet, princip zápisu také (používá spirálový záznam). Formát odpovídá starým zvukovým kazetám.

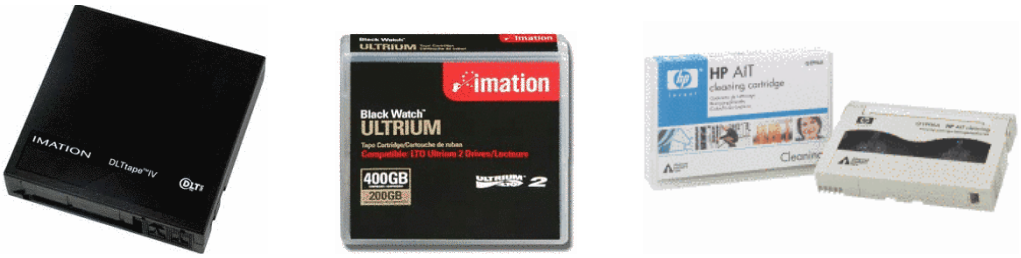
V současné době se DAT pásky dostanou v několika variantách – DDS2, ..., DDS6. Můžeme se také setkat s označením DAT 72, DAT 160, apod. podle kapacity (například DAT 72 má kapacitu 72 GB). Datová propustnost novějších verzí je kolem 10 MB/s.

Obrázek 7.23: DAT mechanika a kazeta

DLT (Digital Linear Tape) jsou o něco dražší, ale obvykle mívají vyšší kapacitu a rychlost zápisu. Používá lineární serpentinový záznam. Novější verze – SDLT – využívá k synchronizaci čtecích/zápisových hlav laser.

AIT (Advance Intelligent Tape) využívá spirálový záznam dat na 8mm pásce (existují verze s půlpalcovou páskou).

[18] Zdroj: http://www.wikiwand.com/en/Magnetic_tape_data_storage



Obrázek 7.24: DLT kazeta, kazeta LTO Ultrium a kazata AIT

LTO, Ultrium (Linear Tape Open) využívá lineární vícekanálový serpentinový zápis na půlpalcové pásce a hardwarovou kompresi dat.

Dnes se u LTO pásek setkáváme především s variantou Ultrium s 384 podélnými stopami při délce pásky 600 m. Pro urychlení čtení dat lze v jednom okamžiku lze číst 8 sousedních stop (u novějších dokonce ještě více). Typická kapacita je ve stovkách GB až jednotkách TB.

Obrázek 7.25: Mechanika Ultrium s rozhraním SAS, sada s rozšiřující kartou

Kapitola 8

Rozšiřující karty

Rychlý náhled: V této kapitole se budeme zabývat různými druhy rozšiřujících karet, které lze v počítači najít. Podrobněji grafickými, zvukovými a síťovými. Rozšiřující karty (taky dedikované) se zasouvají do slotu vedoucího k určité sběrnici, dnes nejčastěji PCIe.

Klíčová slova: Rozšiřující karta, low-profile, grafická karta, grafický čip, integrovaná karta, textový režim, kódová tabulka, Unicode, UTF-8, grafický režim, pseudografika, videopaměť, GPU, shader, stream processor, zvuková karta, vzorkovací frekvence, síťová karta, Wake-on-LAN.

Cíle studia: Po prostudování této kapitoly získáte obecný přehled o běžných typech rozšiřujících karet – grafických, zvukových a síťových kartách.

8.1 Co je to rozšiřující karta

Rozšiřující karta (adaptér, přídavná karta) je modul, kterým se rozšiřuje funkčnost základní desky. Je vyrobena z podobného materiálu jako základní deska a má hodně podobnou strukturu. Připojuje se k základní desce (a tím i k procesoru) zasunutím do *slotů na sběrnici*. Pokud má karta vnější plugy na konektory, jsou umístěny na jedné straně (hraně), která je dostupná vně skříně.

Také existují *integrovaná řešení* (to, co bychom jinak měli zvlášť na kartě určené do slotu, je integrováno v čipu na základní desce, v čipsetu, procesoru apod.). Pak se ovšem nejedná o kartu v pravém smyslu slova, přesto se i v tomto případě můžeme setkat s názvem „karta“, třebaže lepším názvem by byl například „modul“.

Naopak *dedikovaná* rozšiřující karta je právě karta určená do některého slotu (tedy nikoliv integrovaná a vlastně ani externí).

Karty rozlišujeme především podle určení (funkce) – grafická, síťová, zvuková, RAID, SSD (do sběrnice PCIe), různé řadiče, atd.

Rozšiřující karty mají také svá rozhraní:

- slot (do urč. sběrnice) na základní desce – obvykle PCI, PCIe×1, PCIe×16, M.2,
- připojení periferií na kartu – různé vnější a vnitřní konektory.

Karty označené jako low-profile se vejdou i do nízkých skříní (to je důležité nejen kvůli výšce karty, ale i kvůli výšce externí hrany s plugy). V každém případě je však dobré si zkontrolovat celkové rozměry karty (také délku), pokud ji chceme použít do jiné než klasické skříně formátu ATX.

Protože především grafické karty (ale i některé zvukové) jsou opatřeny velkým chladičem, může být problém i šířka – kolize s kartami v sousedních slotech, s chladičem procesoru nebo čímkoliv dalším v okolí.

8.2 Grafická karta

Grafická karta (videokarta) slouží k ovládání grafiky zobrazené většinou na monitoru. Jejím úkolem je zpracovat a transformovat grafická data do podoby, které rozumí monitor či jiná zařízení ke kartě připojená. Dnes bývá často integrovaná obvykle v procesoru, u notebooků je to prakticky pravidlem.

✎ Také se setkáváme s pojmem *grafický čip* – to je čip, který obsahuje grafický procesor. Většina výrobců grafických karet používá grafické čipy od dvou hlavních výrobců a osazuje je na grafické karty pod svým vlastním logem, případně přidává „něco svého“.

Téměř výhradně se dnes setkáváme s grafickými čipy společností AMD (obvykle čipy Radeon) a Nvidia (obvykle čipy GeForce), v menší míře se specifickými čipy Matrox (má například speciální čipy pro CAD). Čipy od společností AMD a nVidia najdeme na grafických kartách od společností MSI, Asus, Gigabyte, Sapphire, Zotac a dalších.

✎ Integrované grafiky mají tyto vlastnosti:

- pro kancelářské použití dostačující,
- levnější (ušetříme několik tisíc korun),
- videopaměť si většinou „půjčuje“ z operační paměti, tedy spolupráce grafiky s pamětí je pomalejší.

Dnes jsou grafická jádra integrovaná ve většině procesorů. Obecně se za lepší považují grafiky v procesorech typu APU od AMD (ale záleží na konkrétním typu procesoru), v procesorech Intel je kvalita grafických jader taky odstupňovaná. Obecně platí, že čím výkonnější procesor jako celek, tím lepší integrované grafické jádro.

8.2.1 Základní typy zobrazení a obsah videopaměti

✎ **Textový režim** umožňuje zobrazovat jen text (v barvách, případně blikající). Setkáme se s ním obvykle při startu systému, v BIOSu (ne v jeho pokračovatelích) a v textových konzolách (také Příkazový řádek Windows pracuje v textovém režimu, třebaže emulovaném), dřív byl běžný na terminálech.

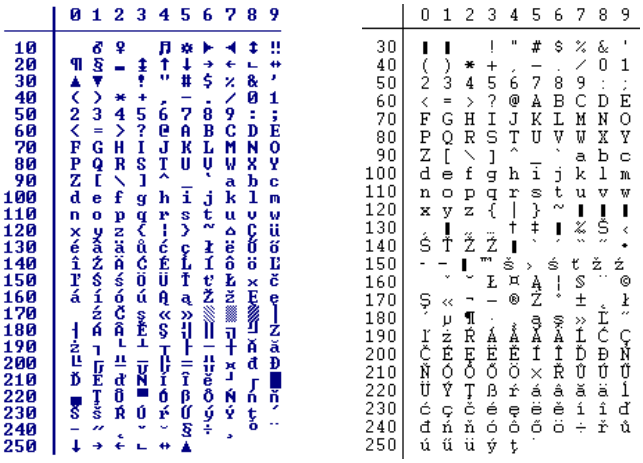
Textový režim lze provozovat v různých *rozlišeních* (počet řádků a sloupců na zobrazitelné znaky), například 80×25 znamená 80 sloupců a 25 řádků, na obrazovku se vejde $80 \times 25 = 2000$ znaků.

✎ V *kódové tabulce* (tabulce znaků) jsou specifikace pro jednotlivé znaky. Tyto tabulky můžeme brát jako pole znaků, každý znak má v tomto poli svůj index (pořadí), tedy číslo, které (zároveň s informací o tom, se kterou tabulkou právě pracujeme) jednoznačně tento znak určuje.

Takových kódových tabulek existuje více. Hlavní odlišnost je v počtu bitů (nebo Bytů), do kterých je index (kód) uložen, a samozřejmě v tom, jaké znaky v tabulce jsou. Od počtu bitů na znak se odvíjí také rozsah tabulky a tím i množství znaků, které v se v tabulce nacházejí.

U nás se dříve v MS-DOSu používala tabulka *Latin2*, ve Windows už mnoho let přetrvávala tabulka *Windows 1250*, na obrázku 8.1 je jejich porovnání. Obě tyto tabulky kódují znaky do 8 bitů, a tedy lze podle nich reprezentovat maximálně 256 znaků (2^8).

V současné době to rozhodně nestačí (zvláště u východních jazyků, kde jeden znak nepředstavuje hlásku, ale obvykle slabiku nebo celé slovo), proto se často používají některé tabulky s kódem *Unicode*, nejčastěji ve formě označované jako UTF-8. V Linuxu i v jiných UNIXových systémech se už velmi dlouho používá kódování UTF-8 (pro všechny jazyky), ve Windows od verze 2000 konečně také.



Obrázek 8.1: ASCII tabulky Latin2 a Windows 1250

Existuje několik variant Unicode – UTF-32 (všechny znaky kóduje do 32 bitů), UTF-16 (některé znaky na 16 bitech, jiné na páru dvou 16bitových čísel) a *UTF-8* (jeden znak je kódován 8, 16, 24 nebo 32 bity). Nevýhodou proměnlivé délky kódů znaků je složitější zpracování, výhodou je (oproti „pevné šířce“) poměrně úsporná velikost souborů s tímto kódem při zachování velkého množství znaků, které lze reprezentovat.

Další informace

UTF-8 si můžeme prohlédnout například na http://www.fileformat.info/info/charset/UTF-8/list.htm.

V textovém režimu je obrazovka rozdělena na *rastr* (mřížku) z buněk pro znaky uspořádaných do určitého počtu řádků a sloupců. Obsah těchto buněk je uložen ve videopaměti, což je paměťový prostor pro udržování aktuálního stavu obrazovky, můžeme si ho představit jako 2D pole. Pro každou buňku jsou zde uloženy dvě informace – ASCII kód znaku, který má být zobrazen, a dále jeho atribut. Každá z těchto informací zabírá jeden Byte, pro každou buňku tedy máme ve videopaměti 2 B. Fyzicky jsou ve videopaměti buňky uloženy „po řádcích“ za sebou, tj. nejdřív první řádek, pak druhý, atd.

Atribut znaku určuje především barvu. V nejběžnějším textovém režimu máme tedy pro atribut k dispozici 8 bitů, z nichž poslední 4 se používají pro barvu samotného znaku ($2^4 = 16$ různých barev), 3 bity před nimi pro barvu pozadí znaku ($2^3 = 8$ různých barev) a zbývající bit (první) určuje, zda má znak blikat (blink). Pokud by mělo být zobrazováno více různých barev, potřebovali bychom pro uložení atributů víc bitů, takže na každý znak mřížky by musely být víc než 2 B.

Příklad

Například na obrazovce je zobrazen tento barevný text: a b c

pokud jsou ve videopaměti tyto údaje:

1 B	1 B	1 B	1 B	1 B	1 B	
Znak	*Atribut*	*Znak*	*Atribut*	*Znak*	*Atribut*	
'a'	0 000 1111	'b'	0 001 1110	'c'	0 000 0100	...

Místo písmen by tu správně měly být ASCII hodnoty těchto znaků. Atribut se skládá z osmi bitů, které určují barvu pozadí, textu a fakt, zda má znak blikat. Význam jednotlivých bitů je následující:

1 bit	3 bity	4 bity
blink	barva pozadí	barva textu

Stránky v textovém režimu: stránka je paměťový blok ve videopaměti obsahující zobrazitelná data, na jednu stránku se vejde obsah obrazovky (ve skutečnosti se na ni vejde ještě o něco více, velikosti stránek jsou zarovnány obvykle na násobky kB). V grafickém režimu velikost stránky přibližně odpovídá rozlišení a barevné hloubce, v textovém režimu je velikost dvojnásobek počtu zobrazitelných znaků (ASCII kód a atribut) zaokrouhlený nahoru na mocninu dvou.

Většina karet v textovém (často i grafickém) režimu umožňuje používat stránky, se kterými lze pracovat i „offline“ – pro jedinou obrazovku existuje více stránek, z nichž jen jedna je právě vykreslena, ale lze ukládat data na kteroukoliv ze stránek a pak jen „přepnout“ zobrazení na tuto stránku. Výhodou je rychlejší vykreslování, protože při provádění změn na zobrazené stránce se po každé změně vykreslí celá obrazovka (třeba změna jediného znaku nebo jeho barvy), ale změny na stránce, která není zrovna zobrazená, se vykreslují, až se stránka zaktivní. Tento princip je i v grafickém režimu, včetně grafických API.

Grafický režim umožňuje zobrazovat matici pixelů (zobrazitelných bodů) místo matice znaků. Každý pixel je určen pouze svou barvou z dané palety barev.

Barevná hloubka je počet bitů, které jsou potřeba k uložení informace o barvě. Například barevná hloubka 8 bitů znamená, že máme k dispozici $2^8 = 256$ barev, barevná hloubka 24 bitů, tedy 3 B, znamená, že je k dispozici $2^{24} = 16\,777\,216$ barev.

Každá barva má tedy své číslo (obvykle 0 bývá černá, nejvyšší možná hodnota bývá bílá, ostatní jsou něco mezi tím). Jaká konkrétní barva se skrývá pod kterým číslem, je určeno v *paletě barev*.

Ve videopaměti je uložena informace o každém pixelu zvlášť a tedy pracujeme s jednotlivými pixely nebo stanovenými skupinami pixelů (například na úsečce, křivce). Množství potřebné paměti se odvíjí od rozlišení obrazovky a barevné hloubky, opět jsou ve videopaměti údaje „po řádcích“, jen místo buněk se znakem a atributem tam máme uloženu barvu, kterou má daný pixel zobrazovat. Množství zabrané paměti je tedy ovlivněno těmito parametry:

- čím větší rozlišení je použito, tím více pixelů je třeba uložit (tj. počet řádků a sloupců),
- čím větší je barevná hloubka (čím více barev), tím víc bitů zabírá jeden pixel (tj. velikost buňky na daném řádku a sloupci),
- další parametry mohou souviset s funkcemi podporovanými kartou.

Pseudografika nabízí možnosti práce s (hrubými) objekty v textovém režimu. Obrázky nebo různé prvky rozhraní (menu, tlačítka, tabulky apod.) vytváříme pomocí speciálních znaků (různě velké obdélníky, intenzita je dána maskou rastru obdélníka), a to monochromatické nebo barevné (co nabízí textový režim).

Například v Latin2 lze použít znaky s ASCII kódem 176, 177, 178, 219, 220, 224, dále jednoduché a dvojité „tabulkové“ čáry, ale i běžná písmena. S pseudografickým rozhraním se setkáváme v BIOS Setup, ale i v mnoha dalších aplikacích. Vpravo je ukázka jednoduchého vystínovaného tlačítka vytvořeného z pseudografických znaků.

8.2.2 Vývoj grafických karet

Vývoj grafických karet šel od úložných prostorů pro videopaměť přes 3D akcelerátory až k dnešním GPU.

S tím, jak se objevovaly monitory s čím dál vyšším rozlišením a množstvím zobrazovaných barev, se vyvíjely i příslušné standardy, které se zjednodušeně označují různými zkratkami:

- MDA (Monchrome Display Adapter) – 2 barvy (monochromatický textový),
- Hercules –oproti MDA zvládá také monochromatický grafický režim,

- CGA (Color Graphics Adapter) – kompatibilita s televizemi, 4 barvy, více barev pouze při velmi nízkém rozlišení,
- EGA (Enhanced Graphics Adapter) – 16 barev z palety 64 možných,
- VGA (Video Graphics Array) – 16 barev, ale mnohem širší základní paleta pro výběr (2^{18} barev), používala rozlišení 640×480 pixelů,
- SVGA (SuperVGA) – začaly se používat grafické akcelerátory, ze začátku jen možnost vykreslení nejčastěji používaných základních tvarů (úsečky, hardwarový kurzor, vyplňování oblasti apod.) – grafický čip,
- XGA (eXtended Graphics Array), SXGA, UXGA, QXGA, HXGA postupně zvyšující rozlišení a barevnou hloubku.

Další informace

Na webu jsou tyto standardy popisovány na mnoha stránkách, například na:
https://www.cknow.com/cms/ref/video-display-standards.html.

3D akcelerátory jsou grafický čipset, který má vlastní grafické API (aplikační programové rozhraní, tedy sadu funkcí, kterými ho bylo možné ovládat) a provádí zpracování objektové 3D reprezentace obrazu do 2D pole pixelů ve videopaměti. Akcelerátor se mu říká, protože urychluje zobrazování (výpočty nemusí provádět procesor, navíc 3D akcelerátor je na ně specializován).

Původně byl 3D akceletátor speciální rozšiřující kartou (běžná grafická karta uměla 2D, tato karta byla pro 3D a napojovala se na „2D“ kartu, které posílala svůj výstup, ta ho posílala monitoru). Velice brzy se však objevila řešení obsahující obě funkcionality na jedné kartě, dnes už ani nic jiného běžně nekoupíme.

GPU (grafický procesor, Graphic Processing Unit) je specializovaný SIMD procesor určený především ke zpracování obrazových dat (což neznamená, že by nemohl zpracovávat něco jiného s podobnou strukturou). A právě GPU máme v grafických čipech, potažmo na grafických kartách. Součástí jeho pracovních povinností je i 3D akcelerace.

GPU implementují určitá *API* (v současné době často některou verzi DirectX, OpenGL, OpenCL, Direct3D, CUDA a další API, resp. jejich rozšíření). S GPU tedy proces může přímo komunikovat pomocí příkazů příslušného API.

8.2.3 Struktura grafické karty

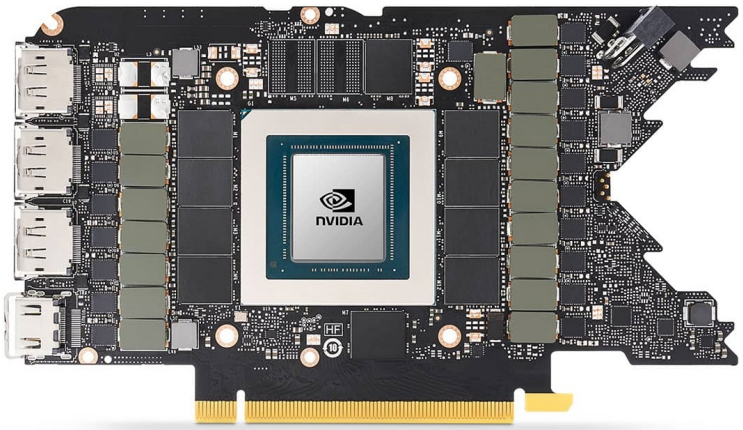
Grafická karta kromě samotné desky plošných spojů může obsahovat tyto komponenty:

- grafický procesor (GPU),
- grafická paměť,
- RAMDAC je čip, který připravuje grafická data z videopaměti na transport do analogového výstupu (VGA) – obsahuje SRAM paměť pro data a barevnou paletu a D/A převodník (pro převod digitálních dat na analogová), pro digitální výstup se nepoužívá,
- chlazení (vzduchové, ale i vodní), slabším kartám stačí pasivní chlazení,
- *VIVO* (Video-in, Video-out) je obvod, který umožňuje také přijímat signál z televize nebo některého videozařízení, zároveň s ním najdeme i rozhraní TV-IN; tato funkce není u grafických karet zcela běžná,
- přídavné napájení (čím výkonnější karta, tím spíše se s ním setkáme), atd.

Grafická paměť bývá reprezentována několika čipy v okolí grafického procesoru na kartě a slouží k uložení těchto informací:

- firmware karty,
- úložiště objektů a textur pro výpočet 3D scén, obecně všeho, co má GPU zpracovat a uložit do videopaměti pro výstup,
- vlastní *videopaměť* s obsahem obrazovky/obrazovek a stránek (*Frame Buffer* – zobrazované „rámce“),
- cache paměti.

Velikost a rychlost videopaměti má velký vliv na kvalitu grafické karty, na její propustnost. Integrované grafiky nemají vlastní čipy pro videopaměť, proto si pro tento účel rezervují část operační paměti.



Obrázek 8.2: Deska tištěných spojů grafické karty Nvidia GeForce RTX 3080[1]

Na obrázku 8.2 je PCB (deska tištěných spojů) grafické karty Nvidia GeForce RTX 3080 z roku 2020. Uprostřed vidíme grafický procesor (GPU) obklopený čipy s grafickou pamětí. V prodeji je samozřejmě ještě opláštěná chladičem.

Provedení pamětí pro grafické karty může být různé:

- starší varianty – VRAM (VideoRAM, dvouportová), SGRAM (Synchronous Graphic RAM) dovolující blokové operace, WRAM (Window RAM) dvouportová s blokovými operacemi,
- DDR2/3/4 paměti – stejné jako u operačních pamětí, nejsou optimalizovány pro roli videopaměti, ale jsou levnější,
- GDDR paměti (Graphics Double Data Rate) – paměti určené přímo pro grafické karty, které se používají na kartách s grafickými čipy od Nvidie, dnes generace GDDR6, GDDR6X, GGDR7
- HBM (High Bandwidth Memory) – od AMD a Hynixu (používá je i Samsung), hodně používaná generace je HBM2 z roku 2017, další jsou HBM2E, HBM3, HBM3E.

Grafický procesor (GPU) vypočítává postupně všechny potřebné parametry obrazu a určí konkrétní hodnoty pixelů na různých pozicích ve videopaměti. GPU běžně používají některé technologie, o kterých jsme se dozvěděli v kapitole o procesorech – například skalární architekturu, tedy *pipeline* se nazývá cesta pro zpracování instrukcí, dnešní GPU mají více pipelines pracujících paralelně.

GPU tedy běžně zpracovává data paralelně. To sice umí i současné procesory (CPU), ale jsou zde jisté odlišnosti. V grafickém procesoru běží na celém poli výpočetních jednotek speciální drobné paralelní programy, kterým říkáme *shadery*, a také zde máme více paměti.

[1] Zdroj: https://www.cnews.cz/nvidia-geforce-rtx-3080-20gb-potvrzena-behem-2-3-mesicu/

Jak je to tedy s *výpočetními jádry*? Běžný procesor CPU má několik (výkonnější až několik desítek) jader. Grafický procesor má stovky až tisíce výpočetních jednotek (pojem jádro se zde nepoužívá, to je až vyšší úroveň slučování). Zatímco procesorová jádra jsou univerzální (složitá) schopná provádět nejrůznější typy úloh, výpočetní jednotky grafického procesoru jsou malé, jednoduché, pro úzkou skupinu úloh.

GPU mohou být pro určité typy úloh rychlejší než hlavní procesor, ale pouze tehdy, pokud jde o potřebu paralelně a opakovaně provádět podobný typ výpočtů. Využívá se toho samozřejmě při zpracování grafiky a multimédií, ale existují aplikace pro využití GPU pro vědecké výpočty, anebo pro šifrování či naopak prolamování bezpečnostních mechanismů (typicky metodou hrubé síly, kdy se zkoušejí různé kombinace znaků či čísel), těžbu kryptoměn a další typy úloh, kdy tutéž relativně jednoduchou operaci provádíme paralelně na dlouhém vektoru dat.

V grafickém procesoru jsou jádra sdružena vždy po několika do modulu, který se označuje SM (Streaming Multiprocessor, u Nvidie) nebo CU (Compute Unit, u AMD). Princip je podobný jako u procesorů, kde třeba u AMD jsme si říkali o CCX modulech a čipletech. Každý takový modul obsahuje určitý počet jader, registry, několik druhů cache, řadič (plánovač) a některé přídavné speciální výpočetní jednotky, také podle výrobce a konkrétní architektury. Některé další speciální výpočetní jednotky jsou mimo tyto moduly.

3D objekty jsou ve scéně reprezentovány jako shluk ploch, nejčastěji trojúhelníků. Vrcholy těchto trojúhelníků se nazývají *vertexy* (tj. každý trojúhelník má tři vertexy) a jsou určeny předně svou polohou v prostoru, barvou (případně světlostí), normálovým vektorem a dalšími parametry, které se vztahují k tomu, jak má být daný vertex zpracován při převodu 3D objektu do 2D obrazu, který vidíme na obrazovce. Převod objektu do reprezentace pomocí trojúhelníků se nazývá *triangulace*.

Textura je obrázek určující vzhled povrchu, kterým se daný objekt „obalí“. Určuje například barvy jednotlivých částí objektu, odrazovost materiálu objektu, atd.

Shader je program vytvořený z určitých vektorových instrukcí (vzpomeňte si na SIMD instrukce). GPU má na vstupu strukturovanou informaci o objektech (kostra daná polynomem, textura, atd.), plánovač výpočetních vláken (Thread Sheduler) podle tohoto vstupu rozděluje úkoly běžícím shaderům:

- *vertex shader* podle instrukcí vytváří a zpracovává 3D scénu, pracuje tedy s reprezentací objektů pomocí vertexů, určuje viditelnost objektů (vzájemné překrývání),
- *geometry shader* dokáže na rozdíl od vertex shaderu také generovat nové jednoduché objekty nebo dotvářet existující, může například přidat či ubrat body, upravit kostru apod., toho se využívá při generování změn obrazu v reálném čase (například zarůstání trávou),
- *pixel shader* pracuje s barvou a průhledností objektů, provádí povrchové úpravy, pracuje s texturami, určuje vzhled povrchu objektů,
- další shadery pro různé účely.

Výstupem grafického procesoru je 2D pole pixelů ve videopaměti, které je při určité „obnovovací“ frekvenci posíláno zobrazovacímu zařízení prostřednictvím výstupního bufferu (frame buffer).

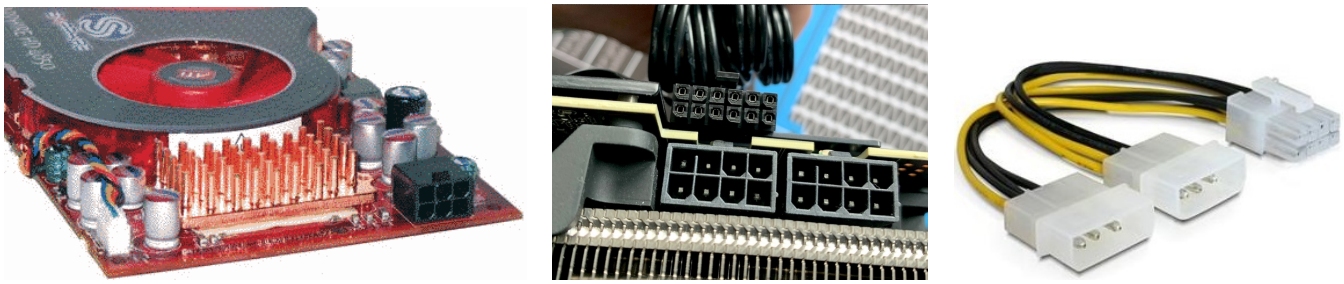
Shadery jsou programovány v jazyce podle zvoleného grafického API. Například jazyk GLSL slouží k programování shaderů pro OpenGL, v jazyce HLSL se programuje pro Direct3D. Tyto jazyky jsou podporovány ve speciálních vývojových prostředích (většinou pro vývoj her nebo virtuální realitu) jako je například Unity nebo Unreal Engine.

8.2.4 Napájení grafické karty

Slabší karty jsou napájeny přes sběrnici PCI Express, ale výkonnější karty vyžadují *přídavné napájení*. Pokud není zajištěno, pracují v „úsporném režimu“, který výkonem odpovídá dokonce jen integrovaným

kartám. Používá se 4 až 8pinový PCIe napájecí konektor („kostičky“), případně může jít o dva 6pinové nebo 8pinové konektory nebo něco jiného s redukcí.

Pokud ze zdroje nevede příslušný konektor, použije se redukce ze dvou 4pinových Molex konektorů na 8pinový PCIe napájecí konektor. Zdroj by měl být výkonnější (400–500 W, při použití více karet ještě víc, také záleží na výkonu karty a účinnosti zdroje – viz kapitolu 10).



Obrázek 8.3: Přídavné napájení na grafice – 6pinové, 2× 8pinové, redukce molexů na 8pinové[2]

8.3 Zvuková karta

8.3.1 Princip

Zvuková karta (nebo její integrovaná varianta) je zařízení, které provádí převod mezi analogovým (mikrofon) a digitálním (CD, DVD, soubor apod.) signálem, vyrábí zvuky z digitálních dat, odesílá do reproduktorů. Obsahuje rozhraní k multimediálním přístrojům/nástrojům včetně případných hudebních (rozhraní MIDI – Musical Instrument Digital Interface).

Může být dedikovaná, externí nebo integrovaná (v čipsetu). U dedikovaných se setkáme s rozhraním PCI, PCIe×1, PCIe×16, externí se obvykle připojují přes USB, Thunderbolt nebo starší přes Firewire.

Z nejznámějších výrobců zvukových karet a čipů: Creative Labs, Axagon, Asus, Focusrite, Genius, Gamebird, Sweex, Orico, Realtek (obvykle integrovaná).

✎ **3D zvuk** je důležitý u některých her, filmů a někdy při fyzikálním modelování. Dociluje se ho dvěma možnými způsoby:

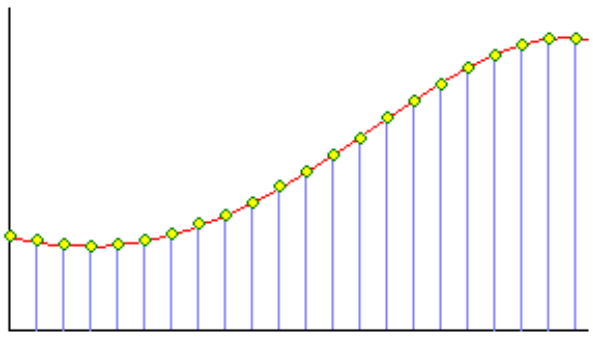
- pravý – Dolby Sorround nebo jiný podobný standard, používá se obvykle pět nebo sedm reproduktorů vhodně rozmístěných tak, aby obklopovaly posluchače, buď horizontální nebo 3D zvuk,
- emulovaný – čip DSP na zvukové kartě nebo v reproduktoru upravuje zvuk tak, aby působil 3D dojmem (máme jen dva reproduktory).

8.3.2 Vzorkování a syntéza zvuku

✎ **Vzorkování.** Při vzorkování (samplování), tj. převodu analogového zvuku (obecně signálu) na digitální data, se vždy určitá část informace ztrácí. Princip vidíme na obrázku 8.4. Vodorovná osa je čas, v pravidelných intervalech se snímá hodnota signálu.

[2]Zdroj: http://www.legitreviews.com/article/730/1/, https://www.hwcooling.net/nvidia-ampere-geforce-rtx-3000-opravdu-novy-12pinovy-napajeci-konektor-udajne-ceny/, https://www.exasoft.cz/

Vzorkovací frekvence je frekvence, s jakou se zjišťují vzorky (momentální hodnoty) analogového zvuku, tyto hodnoty jsou pak digitální data. Čím vyšší frekvence (častější snímání), tím věrnější zvuk po digitalizaci (například pro zvuková CD to bývá přes 44 kHz, tedy 44 000 vzorků za sekundu). Vychází se z toho, že lidské ucho dokáže vnímat v průměru rozmezí 18 Hz až 18 kHz.



Obrázek 8.4: Vzorkování analogového zvuku

Se vzorkovací frekvencí souvisí také pojem *záznamová hloubka*. Je to počet bitů, který je použit pro uložení jednoho vzorku. Vyšší záznamová hloubka (více bitů) znamená jemnější dělení intervalu (na obrázku 8.4 by tento interval byl zakreslen vertikálně). Obvyklá záznamová hloubka je 8 nebo 16 bitů.

Syntéza zvuku je vlastně proces opačný k vzorkování. Digitální data převádíme na zvuk. Existuje více možností, jak z dat generovat zvuk, liší se především kvalitou výsledného zvuku a náročností procesoru převodu.

Nejjednodušší a nejlevnější je FM syntéza (a taky nejméně kvalitní). Zvuky nástrojů jsou generovány speciálním čipem skládáním z předpřipravených vzorků. Další metody používají kvalitnější reprezentace vzorků, například MIDI syntéza používá moduly pro různé nástroje. Syntéza fyzikálním modelováním provádí softwarovou emulaci zvuku založenou na fyzikálních principech, ovšem tato metoda je velmi náročná na výpočetní výkon.

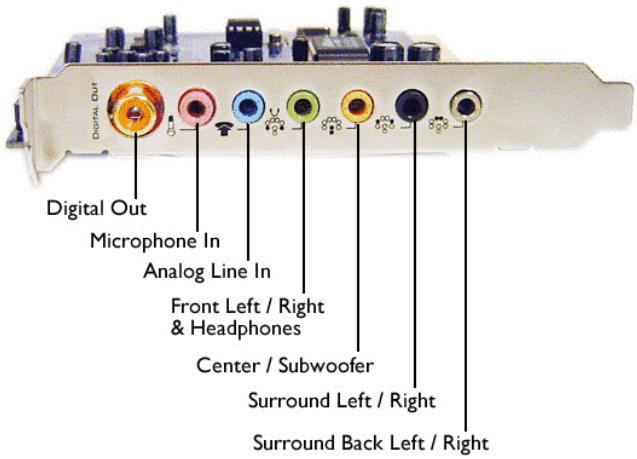
8.3.3 Rozhraní

Zvukovou kartu je třeba jako všechny ostatní karty propojit se základní deskou, a dále obvykle s CD/DVD mechanikou (datový-audio kabel, stejně jako u disků), a pak pomocí externích konektorů s dalšími zařízeními, například reproduktory, rádiem, hudebními nástroji.

Sběrnice pro připojení zvukové karty byla dříve ISA nebo PCI, dnes se interní zvukové karty připojují přes PCI nebo PCIe×1. Externí karty připojujeme většinou přes sběrnici USB nebo ExpressCard (u notebooků vybavených tímto externím slotem).

Kvalitnější karty, které podporují vícekanálový zvuk, mají více různě barevně označených vývodů. 6kanálové mají modrý Line-In, růžový Mic-In, dále zelený pro přední reproduktor (Front Speaker Out), šedý pro zadní reproduktor (Rear Speaker Out), žlutooranžový (yellow orange) pro střední nebo basový (Center/Subwoofer). 8kanálový má navíc černé konektory pro boční reproduktory (Side Speaker Out).

Můžeme se setkat také s digitálním výstupem, konektor je označen S/P-DIF (zkratka z Sony/Philips Digital Interconnect Format), tento konektor vidíme na obrázku 8.5 zcela vlevo.



Obrázek 8.5: Konektory na zvukové kartě[3]

[3]Zdroj: http://koalazereeukalipt.wordpress.com/

Obrázek 8.6: Externí zvukové karty Axagon ADA-17 USB HQ mini autio a Motu 896mk Hybrid[4]

8.4 Síťová karta

8.4.1 Princip

Síťové karty spadají spíše do oblasti počítačových sítí a fyzicky pro ně platí mnohé z toho, co jsme se učili o jiných typech karet, proto se zde na ně podíváme jen krátce.

Síťová karta je zařízení umožňující připojit počítač k síti (obvykle myšleno LAN – lokální síť). Nejčastěji se setkáme s některou formou ethernetové karty s konektorem RJ-45, zvláště u notebooků najdeme také síťovou kartu pro bezdrátové připojení (Wi-Fi kartu), případně v kombinaci s Bluetooth, existují také karty pro připojení do mobilní sítě.

Dedikovaná síťová karta se připojuje na sběrnici PCIe×1 nebo PCI. Pokud jde o integrovanou variantu, o sběrnici se samozřejmě nemusíme starat. Existují také externí síťové karty (většinou W-fi), které se k počítači připojují obvykle přes USB. Některé Wi-Fi karty mají také externí anténu.

U síťových karet nás zajímá především jejich propustnost. Ethernetové karty pro osobní počítače již běžně bývají gigabitové (Gigabit Ethernet), to pro lokální síť bohatě postačí. Ale pozor, pokud ve specifikaci notebooku vidíme jen LAN, Ethernet nebo RJ-45 bez informace že jde o gigabitovou LAN, půjde pravděpodobně o pomalejší 100Mb Ethernet (protože tyto čipy jsou levnější a ceny střední a nižší třídy notebooků jsou tlačeny dolů).

Poznámka

Externí síťové karty připojované přes USB mají jednu nectnost – čím jsou rychlejší, tím víc vytěžují procesor (protože komunikaci USB rozhraní plně řídí procesor). Proto se můžeme setkat s tím, že „bohatší“ komunikace se sítí se projeví sekáním aplikací, a naopak když je procesor velmi zaměstnán výpočetními úlohami, budeme se sítí komunikovat pomaleji.

Výrobců síťových karet je velmi mnoho, z nejznámějších: Intel, D-Link, Zyxel, Netgear, Asus, TP-Link, Belkin, Edimax, Linksys, Axagon, Gamebird, Evolveo.

8.4.2 Související technologie

Wake-On (také Wake-On-LAN) je technologie pro vzdálené buzení počítače (tj. zapnutí na dálku). Počítač s touto síťovou kartou je možné zapnout vzdáleně po síti (například z důvodu automatické údržby systému), kromě podpory u karty je nutné mít zdroj ATX (to obvykle bývá, případně kompatibilní) a podporu u základní desky.

[4] Zdroje: https://www.tsbohemia.cz, https://www.muzikant.cz

Síťová karta s podporou Wake-On, která je takto připojená, naslouchá na síti a čeká na signál Wake-On (Magic Packet) se svou MAC adresou. Tento signál lze generovat na jiném počítači (většinou serveru) speciálním programem nebo skriptem. Počítač, který je zdrojem Magic Packetu, obvykle bývá v lokální síti, ale existují možnosti (například přes virtuální síť, tunel apod.), jak tento signál dostat do LAN zvenčí. Formu Magic Packetu a také skripty pro řízení Wake-On lze najít na internetu.

Poznámka

Karta, která takto naslouchá na síti, musí být samozřejmě pod proudem. Další nevýhodou je, že signál Wake-On se ke kartě může dostat „omylem“ nebo sice záměrně, ale ten záměr nepatří majiteli zařízení (pak ráno přijdeme k počítači a divíme se, že je zapnutý). Pokud tedy nemíníme funci Wake-On využívat, je lepší ji vypnout v BIOSu.

Vzdálené bootování je možnost bootovat systém ze serveru na síti. Síťová karta musí být vybavena čipem Boot ROM (nebo variantou ROM paměti, viz kapitolu o vnitřních pamětech), ve kterém je uložen program inicializující bootování přes síť. Pokud počítač (pracovní stanice) takto nabootuje ze serveru, operační systém a aplikace běží na serveru a tedy na serveru probíhá veškeré vyhodnocování (výpočty). Samotná pracovní stanice pak funguje jako terminál (textový nebo s grafickým rozhraním) pro přístup k prostředkům serveru, vůbec na ní nemusí být nainstalován operační systém (nebo může, pak by bootování přes síť bylo jednou z více možností při spuštění).

Takto řešená síť se používá především pro tenké (thin) klienty (tj. bezdiskové stanice). Není vhodná pro Windows (aplikace pro Windows jsou velmi objemné, síť by se zahltila), ale pro unixové systémy je to dostačující. Také je možné toto řešení použít pro spouštění diagnostických a administračních programů, které ke svému provozu nepotřebují lokální operační systém, zajímavé může být řešení ve spojením s UEFI.

Kapitola 9

Vstupní a výstupní zařízení

Rychlý náhled: Tématem kapitoly jsou nejrůznější vstupní a výstupní zařízení, včetně zařízení zvládajících „oba směry“, tedy vstupně-výstupních. Budeme se zabývat jednoduchými zařízeními typu klávesnice, myš, touchpad, a dále zařízeními pracujícími s obrazem jako je scaner, monitor, tiskárna.

Klíčová slova: Znakové, blokové a speciální zařízení, ukazovací zařízení, trackball, trackpoint, touchpad, tablet, dotyková obrazovka, rezistivní a kapacitní technologie, aditivní a subtaktivní model, RGB, CMYK, normovací systém CIE, sRGB, Adobe RGB, pixel, barevná hloubka, rozlišení, doba odezvy monitoru, CRT monitor, LCD panel, CCFL a LED podsvícení, Edge LED, TFT, TN, IPS, VA, OLED, PMOLED, AMOLED, tiskárna, InkJet, BubbleJet, cartridge, toner.

Cíle studia: Po prostudování této kapitoly budete vědět, jak fungují různé druhy vstupních a výstupních zařízení, naučíte se principu fungování zobrazovacích zařízení a budete se orientovat v jejich vlastnostech. Budete umět posuzovat různé druhy zobrazovacích zařízení podle jejich parametrů.

9.1 Vstupy a výstupy

Nejdřív si všechna zařízení, která lze zahrnout do této kapitoly, rozdělíme podle směru toku dat a podle velikosti dávky zpracovávaných dat.

Podle směru toku dat

- vstupní – slouží ke vstupu dat do počítače (klávesnice, myš a další ukazovací zařízení, scanner a další digitalizační zařízení, mikrofon, kamera, atd.),
- výstupní – slouží k výstupu dat z počítače (monitor, tiskárna, plotter, reproduktory, apod.),
- vstupně/výstupní – umí obojí (dotyková obrazovka, multifunkční zařízení, atd.).

Podle velikosti dávky zpracovávaných dat

- znaková – malá šířka přenosu, jeden nebo několik znaků (myš, klávesnice, tiskárna, mikrofon),
- bloková – větší šířka přenosu, přenášený blok dat má určitou strukturu či je nutné přenášet vedle dat i metadata[1] (paměťová zařízení),

[1] Metadata jsou data o datech. Například pro soubor ukládaný na disk jsou metadata název souboru, pozice v souborovém systému (kam má být uložen), přístupová oprávnění, atd.

- speciální – jeden typ dat s velmi malou šířkou přenosu (jeden nebo několik bitů) bez jakékoliv struktury, například časovač.

Poznámka

Správně bychom měli v této kapitole probírat také klávesnice, myši apod., ale vzhledem k tomu, že máme k dispozici opravdu jen jeden semestr a i ten je poněkud „přecpán“, od toho upustíme a témata osekáme jen na to nejnutnější.

9.2 Ukazovací zařízení

Myši jsou prvním a nejobvyklejším ze zástupců tzv. „ukazovacích“ zařízení. Ukazovací zařízení jsou taková zařízení, která umožní svým posouváním, posouváním, otáčením nebo jiným typem pohybu některé své části nebo jiným způsobem zadávat souřadnice na zobrazovacím zařízení (třeba obrazovce), a nebo jejich změny.

Nejznámější výrobci současných myší se celkem kryjí s výrobci klávesnic (ostatně často se setkáme se setem myš plus klávesnice), tedy Logintech, Genius, Mirosoft apod.

9.2.1 Trackball, Trackpoint, Touchpad, Tablet

Trackball je kulový ovladač, kulička vestavěná v ploše, ale může být samostatné zařízení. Jde o „mechanickou myš naruby“, také bývají poblíž obdoby tlačítek. S trackballem se dnes už moc často nesetkáme. Dříve býval u notebooků jako náhrada myši, ale postupně byl nahrazen touchpadem. Velkou nevýhodou bylo především zanášení kuličky špínou z uživatelových prstů, což způsobovalo nepřesnost (poskakování kurzoru), kuličku a snímací válečky bylo třeba často čistit.

Trackpoint (Pointing stick, Flex point, SensePoint,...) je kolík na klávesnici (nebo jiném povrchu), na kloubu, funguje podobně jako joystick. Jsou jím vybaveny některé notebookové klávesnice. Pohyb trackpointu se přenáší do číselné podoby podobně jako u myši.

Původně to byla specialita IBM (divize později prodána Lenovu), pod jinými názvy se vzácně vyskytuje i u jiných výrobců.

Touchpad je dotyková destička, obvyklá u notebooků. Používá se už od roku 1988, masového nasazení se dočkal roku 1994 na noteboocích Apple PowerBook 500. Lehký pohyb prstu na destičce je dekódován a přenášen do počítače, ovladače umožňují i operaci scrool (rolování) – scrooling je tedy záležitostí ovladačů, samotný touchpad nic takového neumí. Současné touchpady používají stejné technologie jako dotykové obrazovky.

Další informace

V současné době existují dvě základní technologie, na kterých je touchpad založen. Obě jsou popsány například na https://notebook.cz/clanky/technologie/2009/ovladani-notebooku.

Tablet (tím není míněno samostatné zařízení) se skládá z pevné podložky a speciálního pera. Pero může být *aktivní* (snímá svůj pohyb) nebo *pasivní* (pohyb pera snímá podložka), aktivní pero vyžaduje napájení a je obvykle napájeno elektromagnetickou indukcí.

Obrázek 9.1: Trackball, trackpoint, touchpad, tablet

U tabletu se kromě změn souřadnic (podobně jako u myši) také rozlišují různé přítlaky. Na kvalitě tabletu pak závisí celkové množství přítlaků, které lze rozlišit. Reakcí na změnu přítlaku může být například změna tloušťky kreslené čáry.

Poznámka

Z tabletu ve výše naznačeném smyslu se postupně stalo samostatné zařízení s vlastním operačním systémem. Na druhou stranu se objevily tzv. grafické tablety s velkými obrazovkami, které naopak nemají vlastní operační systém a připojují se k počítači přes HDMI+USB.

9.2.2 Dotykové obrazovky

Obrazovkám samotným se budeme věnovat později. Na začátku 70. let 20. století se nezávisle na sobě objevilo několik různých řešení dotykových obrazovek, všechny byly tzv. „single-touch“, tedy v jeden okamžik dokázaly zpracovat pouze jeden dotyk. Vícedotykové ovládání (multi-touch) se objevilo až v 21. století v produktech Apple iPhone a MS Surface.

Existuje více různých technologií dotykových obrazovek. Nejpoužívanější jsou

- *odporová (rezistivní) dotyková technologie* – dvě vrstvy z elektricky vodivého materiálu se při dotyku dostanou k sobě, místo dotyku je pak možné zjistit; možnost ovládání jakýmkoliv předmětem, vysoká přesnost, nízká spotřeba energie, je levnější, ale naopak průzračnost je horší, je méně odolná (při poškození svrchní vrstvy zcela přestane fungovat), dnes už je na ústupu,
- *kapacitní dotyková technologie* – máme pouze jednu vodivou vrstvu nanesenou na izolantu (třeba skle), dotyk je registrován jen tehdy, když se dotkneme něčím vodivým (i lidský prst je vodivý); přesná a odolná technologie, průzračnost je lepší než u rezistivní technologie, je odolnější, ale lze ovládat pouze prstem nebo speciálním perem pro kapacitní displeje.

Existují i další technologie – například akustická vodivá technologie, IR technologie.

Další informace

- https://smartmania.cz/evoluce-dotykovych-displeju-od-svetelnych-per-po-soucasne-technologie/
- https://mobilenet.cz/clanky/techbox-dotykove-displeje—cim-se-lisi-rezistivni-od-kapacitniho-11566

9.3 Barvy v grafice

9.3.1 Počítač zpracovává barvy

Počítač umí zpracovávat pouze čísla, proto i barvy musíme převést na čísla. Každé číslo má svůj horní limit. Barev je z lidského hlediska nekonečný počet, ale počítače nekonečno nezvládají, proto musel být vy-

myšlen způsob, kterým reprezentujeme dostatečně široké spektrum barev pomocí konečných čísel. Metoda reprezentace barev v počítači vychází z těchto zásad:

- každá barva vzniká kombinací *základních barev*
- jsou určeny tyto základní barvy a metoda jejich kombinace

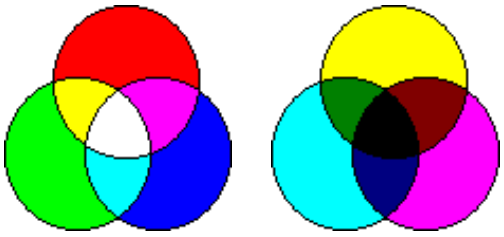
Způsob vnímání barev lidským okem je důležitý pro pochopení principu zobrazovacích zařízení, je na něm založen samotný mechanismus zobrazování. Ovšem je třeba rozlišit tato zařízení do dvou typů:

1. zařízení, která zobrazují generováním světla o určitých vlnových délkách (monitory, generují vlastní světlo – obraz je podsvícen)
2. zařízení, která vytvářejí „neelektronické“ výstupy tisknuté na papír, fólii nebo jiný podobný materiál (tiskárny apod.), a tedy barvy jsou vnímány odrazem světla od povrchu a vlastní světlo negenerují.

První typ zařízení používá aditivní (přičítací) model skládání barev, druhý typ zařízení používá subtraktivní (odečítací) model.

Aditivní model je určen pro zařízení, která barvy aktivně zobrazují vyzařováním (monitor, projektor). Základní barvy jsou červená, zelená a modrá (**RGB**, konkrétně model sRGB). Smícháním všech základních barev v plné intenzitě vznikne bílá.

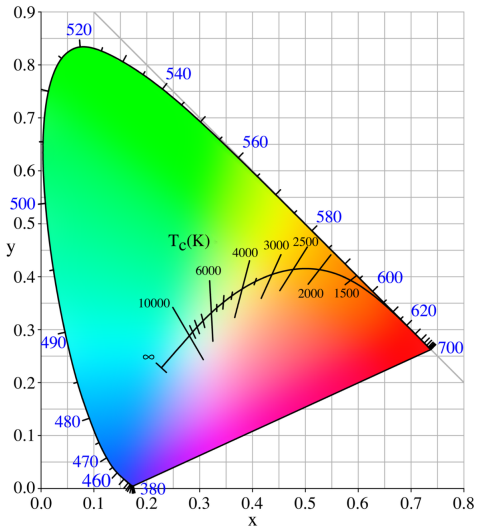
Subtraktivní model je určen pro zařízení, která zobrazují pasivně (ne vyzařováním), například tiskárny. Základní barvy jsou cyan (modrozelená), magenta (červenofialová) a žlutá (**CMY**). Smícháním všech barev v plné intenzitě vznikne černá (šedočerná).



Obrázek 9.2: Aditivní a subtraktivní model skládání barev

CMYK (Cyan-Magenta-Yellow-Key – black) je typickým příkladem subtraktivního modelu, který se běžně používá v tiskárnách. Jedná se obohacení výše popsaného modelu CMY o černou barvu. Důvodů je více – jeden z nich je finanční (samostatná barva vyjde levněji než její míchání, když černá je při tisku nejpoužívanější) a také smícháním CMY se z technických důvodů nedá získat naprosto čistá černá (získali bychom „špinavou“ tmavě šedou barvu).

RGB (Red-Green-Blue) je zase typickým příkladem aditivního modelu. V několika podobách se používá především u monitorů a projektorů. Ve skutečnosti existují i jiné aditivní modely než RGB, ostatní jsou na RGB převeditelné a obvykle s nimi pracujeme v rozsáhlejších a kvalitnějších grafických editorech. Jejich účelem je zjednodušení nastavení některých vlastností souvisejících s barvami (například světlosti obrazu). My se jimi zabývat nebudeme, jsou záležitostí předmětů zabývajících se zpracováním grafiky.



Obrázek 9.3: Chromatický diagram CIE[2]

9.3.2 CIE

Normovací systém CIE (CIE colorometric system) vznikl za účelem sjednocení standardů pro RGB s tím, aby co nejvíce korespondovaly se systémem CMYK. Důvodem byly zejména četné

[2]Zdroj: http://en.wikipedia.org

problémy s rozdíly mezi zobrazením obrázku (zejm. fotografie) na monitoru a následným vytisknutím tohoto obrázku. Má tyto vlastnosti:

- „podkova“ zachycující světlo různých vlnových délek v závislosti na červené a zelené, vidíme ji na obrázku 9.3,
- uvnitř této podkovy se stanoví body pro základní barvy (tolik bodů, kolik základních barev).
- konvexní n-úhelník vzniklý spojením těchto bodů (gamut) určuje barevný prostor daného modelu.

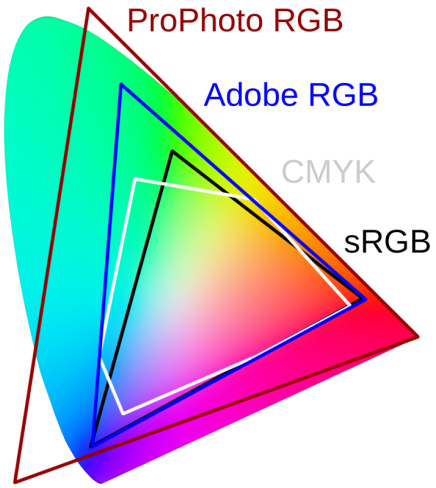
Model sRGB je starší implementací normovacího systému CIE (roku 1996) původně pro CRT monitory. Tento standard najdeme u mnoha dnešních monitorů, spíše levnějších. Souřadnice červené, zelené, modré a bílé obdržely v systému CIE hodnoty blízké vyjádření CMY. Účelem bylo přiblížení vzhledu obrázku na monotoru jeho vytisknutému ekvivalentu.

Model sRGB používá většina českých fotosběren, proto když posíláme fotografie do fotosběrny a nejsou v tomto barevném modelu, je vhodné předem provést převod (i když to obvykle není potřeba, digitální fotoaparáty a většina programů pro zpracování grafiky pracuje v sRGB).

Adobe RGB vznikl z potřeby zvětšit barevný prostor, a také proto, že v barevném prostoru sRGB některé barvy nevypadaly moc přirozeně (především v oblasti zelené barvy, například tráva).

V normovacím systému CIE má trojúhelník pro Adobe RGB větší obsah než sRGB především směrem k zelené. Fotografie z přírody mají při zobrazení na Adobe RGB monitoru přirozenější barvy.

Adobe RGB podporují některé monitory (většinou dražší), ale existují mezistupně (například o něco levnější monitory, které dokážou „téměř“ Adobe RGB, třeba 95 %).



Obrázek 9.4: Různé druhy barevných modelů[3]

Poznámka

Adobe RGB používáme obvykle tehdy, když máme kvalitní fototiskárnu (nebo profesionální tiskárnu), protože při tisku přes fotosběrnu je stejně (většinou) nutné provést převod do sRGB. Pozor, nemá smysl používat monitor podporující Adobe RGB, pokud vytváříme elektronické výstupy (například webovou grafiku, kterou nemíníme tisknout) a nebo když navrhujeme rozhraní aplikace – v obou případech budou našimi zákazníky lidé s „obyčejnými“ monitory, a tedy co se nám na kvalitním monitoru zdá naprosto úchvatné, může na běžném monitoru vypadat mnohem méně esteticky.

ProPhoto RGB od Kodaku se vyznačuje velmi rozsáhlým gamutem. Není určen pro výstupy, ale spíše pro úpravy obrazu, protože při různých operacích s obrazem redukuje ztráty dat.

Další informace

- https://toolstud.io/color/rgb.php (converter)
- https://fotocesta.cz/blog/fototipy/barevne-prostory-2-dil-je-lepsi-zvolit-srgb-nebo-adobe-rbg
- http://en.wikipedia.org/wiki/Adobe_RGB_color_space, http://en.wikipedia.org/wiki/SRGB_color_space
- http://en.wikipedia.org/wiki/RGB_color_space, http://en.wikipedia.org/wiki/CMYK
- http://en.wikipedia.org/wiki/Color_theory

[3] Zdroj: https://www.milujemefotografii.cz/srgb-prophoto-rgb-a-dalsi-vyznate-se-v-barevnych-prostorech

9.3.3 Pojmy z grafiky

Pixel je zkratka z picture element, je to jeden zobrazitelný bod.

Rastrový obrázek je obvykle chápán jako mřížka pixelů. U tohoto typu obrázku (s příponou bmp, png, jpg, gif, ...) je tedy důležité, jaké barevné hodnoty jsou přiřazeny jednotlivým pixelům, i když nemusejí být vždy uloženy ve formě seznamu barevných hodnot (často se používá některá forma komprese, aby obrázek nezabíral moc místa).

Vektorové obrázky naproti tomu nejsou definovány jako mřížka pixelů, ale jako posloupnost operací (například operace pro vykreslení úsečky, křivky, apod.), převod na mřížku pixelů se provádí jen při výstupu obrázku na zobrazovací zařízení.

Barevná hloubka je množství paměti potřebné k reprezentaci jednoho barevného bodu – pixelu (v bitech), obvykle zvlášť pro každou základní barvu (8+8+8, 16+16+16 apod.). U zobrazovacích zařízení může být část barevné hloubky vyhrazena pro alfa-kanál (stupeň průhlednosti).

Rozlišení je počet bodů na jednotku délky, většinou palec, případně rozlišení obrazu může být bráno jako počet řádků a sloupců matice pixelů. Obvykle se udává v *dpi* (Dots per Inch) nebo jako dvojice hodnot řádky×sloupce. Při porovnávání kvality výstupu na monitor a tiskárnu se setkáváme s rozlišením dpi pro tiskárnu a ppi (pixels per inch) pro obrázek na monitoru.

Denzita je schopnost rozlišovat (příp. odrážet nebo zpracovávat) různé vlnové délky světla. Jedná se o záporný logaritmus poměru dopadajícího a propuštěného (nebo odraženého) světla. Dokonale propustný materiál má denzitu 0, materiál, který není průsvitný, má vysokou denzitu.

9.4 Scannery

Scanner je zařízení pro snímání předlohy do počítače pracující na principu digitalizace odstínu barvy, a to v modelu RGB. Rozlišujeme tyto typy scannerů:

- stolní,
- ruční, včetně scannerů čárových kódů,
- archové (průchodové) – skenovaný objekt prochází přístrojem,
- tužkové – na řádek textu,
- 3D,
- bubnové – na skenovaný objekt je nanesen speciální olej (během skenování vyprchá), který umožní objektu přesně přilnout k bubnu, buben se otáčí, od objektu se odráží světlo, které je zachycováno (používá se zejména pro filmy).

9.4.1 Snímací technologie

Snímací technologie scannerů je založena na osvětlení předlohy, sejmutí světlocitnými prvky a následném zpracování výstupu těchto prvků. Princip je hodně podobný tomu, co se děje v digitálních fotoaparátech, a vlastně se můžeme setkat i se stejnými snímacími prvky.

CCD (Charged Coupled Device) je technologie, kdy je předloha osvětlena bílou zářivkou umístěnou na snímací hlavě, doplněnou o optický systém, který zajišťuje stejnoměrné osvětlení. Snímání probíhá takto:

- hlava se pohybuje pod skleněnou deskou, na které je předloha
- pomocí soustavy zrcadel je obraz odrážen do snímacího prvku s třemi řadami CCD prvků (každý pro jednu základní barvu), snímá se vždy celá šířka najednou (pixely na stejném řádku)

Zářivka („lampa“) se musí nejdřív nahřát, což znamená časovou ztrátu. Nejčastějším problémem u scannerů s CCD prvky bývá nízká životnost lampy.

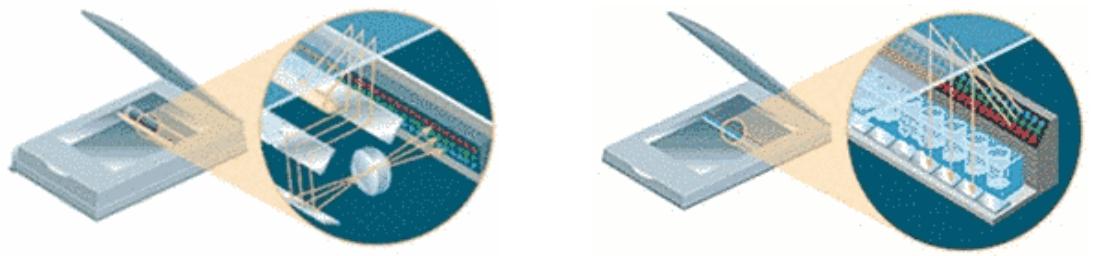
Možnosti sejmutí obrazu:

1. předloha je snímána třikrát, vždy s jiným filtrem, z těchto tří výsledků se poskládá finální obraz,
2. předloha se snímá jednou, ale každý řádek bodů je osvětlen postupně přes tři různé filtry – rychlejší,
3. rozklad světla odraženého od předlohy se provádí přes optický hranol, výsledné tři paprsky se vedou k třem řádkovým CCD snímačům.

CIS (Contact Image Sensor) osvětluje snímaný objekt LED diodami. Na snímací hlavě jsou *tři řady LED diod* (každá svítí jednou ze základních barev), diody musí být co nejblíž předloze.

Výhody CIS technologie jsou možnost použití nižšího napětí (proto USB scannery bez dalšího napájení bývají vybaveny CIS), vyšší životnost, možné rozměry snímače, spolehlivost, cena.

Naopak nevýhodou je, že snímací hlava musí být blízko snímaného objektu (LED diody mají mnohem nižší výkon než zářivka), proto u „plastičtějších“ předloh jsou výsledky horší, taktéž se moc nehodí pro snímání částečně průsvitných předloh (negativy, diapozitivy apod.).



Obrázek 9.5: CCD a CIS technologie

CCD s bílými LED jsou v poslední době čím dál častější. Tyto scannery fungují na stejném principu jako výše popsaný CCD, ale místo zářivky se jako zdroj světla používají bílé LED diody. Výhodou je především menší spotřeba energie a tím i možnost napájení přes USB, a výrazně kratší doba nahřívání oproti zářivkám.

9.4.2 Optické rozpoznávání znaků

Optické rozpoznávání znaků (OCR – Optical Character Recognition) úzce souvisí s používáním scannerů. Velmi často skenujeme text, který bychom chtěli buď přímo v upravitelné podobě, nebo alespoň ve vektorových fontech.

Kdybychom nepoužívali OCR, výsledek by byl vždy rastrový. U samotných obrázků to tolik nevadí (i když se tím připravujeme o možnosti téměř neomezeného zvyšování rozlišení při zvětšování obrázku), ale u textu může jít o velký problém, protože při změně rozlišení nebo třeba tisku písmen se může zhoršit čitelnost.

Jak OCR funguje?

- metoda se v naskenovaném obrázku pokouší rozpoznat znaky, účelem je převést obrázek do formy dokumentu s vektorovými fonty, případně upravitelného dokumentu,
- předloha musí mít dostačující kvalitu,
- OCR pracuje nejen s možnými tvary znaků, ale používá databázi nejpoužívanějších slov jazyka.

Pro OCR se používají různé alternativní metody (neuronové sítě, expertní systémy, fuzzy apod.).

9.5 Monitory

Monitor je výstupní elektronické zařízení sloužící k zobrazování textových a grafických informací. Bývá propojen s grafickou kartou nebo s jinými zařízeními (reproduktory), může být integrován do zařízení.

Monitory mívají různou úhlopříčku (udává se v palcích) – například 14", 15", 19", 21". Poměr stran je obvykle 4:3 (klasický) nebo 16:9 (širokoúhlý), příp. 16:10 nebo jiný, může mít zaoblené strany.

9.5.1 Parametry monitoru

Když kupujeme monitor, zajímáme se o jeho parametry. Co nás má (zatím obecně bez určení typu) zajímat?

Rozlišení je počet pixelů (zobrazitelných bodů) na šířku a výšku. Obvyklé hodnoty jsou

640×480 (VGA),	1280×800 (WXGA),	1366×768 (HD Ready),
800×600 (SVGA),	1280×1024 (SXGA),	1920×1080 (Full HD, 1080p),
1024×768 (XGA),	1600×1200 (UXGA),	4096×2160 (4K),...

Často se používají slovní názvy:

- HD: 1280×720, také 720p
- Full HD (FHD): 1920×1080, také 1080p
- Quad HD (QHD): 2560×1440, 4× obrazovka HD
- Ultra HD (UHD): 3840×2160
- 4K: 4096×2160 (přesněji: DCI 4K je 4096×2160, 4K UHD je 3840×2160)
- 8K, 16K, atd.

Použitelný výběr závisí na tom, zda jde o klasický nebo širokoúhlý monitor. Horní mez závisí na použité obnovovací frekvenci.

Doba odezvy je doba, za kterou pixel změní barvu z černé na bílou a zpět. Parametr „rise“ znamená dobu potřebnou pro rozsvícení bodu na bílou, „fall“ dobu pro zhasnutí do černé.

Ovšem více vypovídající je parametr „grey to grey“ – změna z tmavě šedé do světle šedé a zpět.

Obnovovací frekvence je frekvence, s jakou se obnovuje obsah obrazu, bývá v jednotkách Hz nebo FPS (Frames per second). Například 60 Hz (tj. 60 FPS) je hodnota typická pro běžné displeje, herní monitory mají spíše kolem 120 Hz nebo vyšší.

Input lag je zpoždění mezi zdrojem grafických dat (tj. grafickou kartou či její obdobou) a cílem (zde obrazovkou). Není to totéž jako doba odezvy, protože zde připočítáváme i cestu do monitoru. Na této hodnotě se podepisují rozhraní, kvalita a délka kabelu, atd. Typické hodnoty kvalitních běžných sestav jsou mezi 40 a 60 ms, pro hraní rychlých her jsou potřeba nižší čísla.

Poznámka

Hodnoty udávané výrobcem musíme v každém případě brát s rezervou, protože jde o údaje naměřené v ideálních laboratorních podmínkách, kterých v běžném prostředí s různými světelnými podmínkami při zobrazování běžných obsahů obrazovky dosáhneme jen výjimečně. Navíc může nastat problém při přenosu obrazu – přeslechy v kabelu, problém na straně grafické karty, apod.

Další parametry

- spotřeba v pracovním režimu/režimu spánku,
- rozteč pixelů,

- pozorovací úhly (při sledování obrazovky šikmo se deformuje především zobrazení barev),
- rozměry (především hloubka), hmotnost, atd.,
- použitá technologie – CRT, LCD (TN, IPS, VA, apod.), OLED, FED, SED, atd.

Konektory na monitorech (VGA, DVI, HDMI, Display Port, USB) jsou v kapitole 3 Rozhraní a konektory od strany 22, zde je už nebudeme rozebírat.

Poznámka

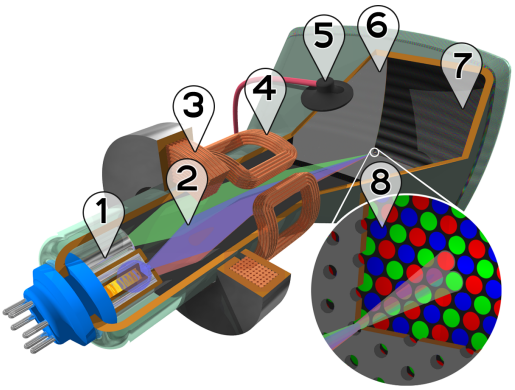
Záleží na konkrétním rozhraní a jeho verzi, jak moc dat přes ně za časovou jednotku přeneseme. Čím vyšší rozlišení a čím vyšší obnovovací frekvence, tím větší propustnost potřebujeme. Vzpomenete si, jaká rozhraní a v které verzi bychom potřebovali pro přenos 4K obrazu při frekvenci 60 Hz?

9.5.2 CRT monitory

CRT (Cathode Ray Tube) monitory ještě donedávna byly jediným typem monitorů pro osobní počítače. Jejich základem (objevujícím se i v názvu) je *katodová trubice*.

Obvykle jde o skleněnou trubici, uvnitř ní bývá vakuum nebo velmi řídký plyn. V trubici jsou elektrody (anoda a katoda), mezi nimiž po připojení k elektrickému napětí prochází elektrický proud. Katoda tvoří „elektronové dělo“, které emituje elektrony, ty jsou v trubici urychlovány směrem k anodě a vychylovacími cívkami nasměrovány na stínítko. Stínítko ve fluorescenční vrstvě obsahuje luminofory, které po dopadu elektronů svítí. Intenzita proudu elektronů znamená intenzitu svícení bodu, u černého bodu je intenzita nulová (žádné elektrony).

Barevné monitory (RGB) mají tři elektronová děla, pro každou základní barvu jedno, před stínítkem je maska filtrující pro jednotlivé základní barvy.



1. elektronová děla
2. svazky elektronů
3. zaostřovací cívky
4. vychylovací cívky pro jednotlivé barvy
5. anoda
6. maska oddělující barvy
7. fosforová vrstva s R, G, B zónami
8. detail vnitřní části obrazovky

Obrázek 9.6: Princip CRT monitoru[4]

Obnovovací frekvence je rychlost obnovování obrazu. V jednom okamžiku je vysvícen jen jeden pixel. Proud elektronů ze všech tří děl je synchronně vychylován po řádcích, po celém průchodu jsou postupně vysvíceny všechny body obrazovky. Pixel však svítí jen po určitou omezenou dobu; aby obraz neblikal, musí být proud elektronů vychylován dostatečně rychle.

Vychylování je zajištěno vychylovacími cívkami horizontálními a vertikálními. Rozlišujeme horizontální frekvenci (frekvence vysvícení jednotlivých řádků) a celkovou obnovovací frekvenci (pro celou obrazovku, přes všechny řádky).

[4] Zdroj: http://www.absoluteastronomy.com/topics/Cathode_ray_tube

Obnovovací frekvence nám tedy říká, kolikrát za sekundu se vykreslí všechny řádky obrazovky (vertikální frekvence). Nesmí být moc nízká, aby obraz neblikal (bolely by z toho oči, udává se nejméně 60 Hz), ale ani moc vysoká (zvláště při vysokém rozlišení by mohly vzniknout problémy s neostrým obrazem, navíc to zvyšuje spotřebu).

9.5.3 LCD panely

LCD (Liquid Crystal Display) je nazvána podle tekutých krystalů. Tekutý krystal je materiál, který při průchodu elektrického proudu mění svou molekulární strukturu. LCD jsou většinou založeny na propouštění či blokování světla (z vnějšího zdroje nebo světla ze zadního či bočního podsvícení).

Podsvícení. LCD panel sám o sobě nezáří, jen určuje, kterými místy světlo projde a s jakou intenzitou. Potřebujeme tedy nějaký zdroj světla. Pro podsvícení existují dvě základní technologie:

1. CCFL (Cold Cathode Fluorescent Lamp) – studené katodové zářivky, a to buď několik horizontálně nad sebou nebo je zářivka umístěna na straně panelu a její světlo je vedeno přes celý panel pomocí světlovodu a reflexní vrstvy (často více reflexních vrstev). Na panelu používajícím CCFL prakticky nelze zobrazit černou barvu, vždy se na pixel odráží trochu světla. S touto technologií se dnes moc nesetkáváme.
2. LED – světlo je generováno LED diodami. Rozlišujeme několik typů lišících se hlavně umístěním LED diod. Dnes je běžné *Edge LED*, kde jsou LED diody umístěny na boku panelu a jejich světlo je rozváděno po celém panelu světlovody a odrazovými prvky. Existuje také *RGB LED* (Sony), kde jsou diody v matici po celé ploše panelu, každý pixel má čtyři LED diody (červenou, modrou a dvě zelené), a dále *Direct LED*, které je podobné, ale LED diody jsou jen bílé.

Edge LED postrádá výhodu možnosti zobrazení čistě černé barvy, ale má jinou výhodu – takto osvětlované panely mohou být velmi tenké, což je výhodné zvláště u notebookových panelů. Edge LED má ještě jednu výhodu: nižší cenu. Původně byly LED diody na dvou protilehlých stranách obrazovky, ale dnes se setkáme spíše s umístěním jen na jednu stranu, čímž výrobci snižují cenu na úkor rovnoměrnosti podsvícení.

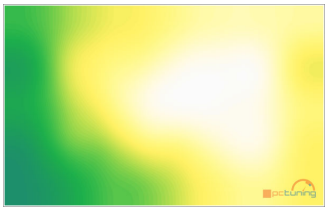
Také se můžeme setkat s pojmem WLED (White LED), což je prakticky totéž (bílé LED diody jsou použity jako podsvícení, barva je tvořena jinak, například barevným filtrem u subpixelu).

U všech technologií je důležité, jak kvalitně je světlo rozvedeno po celém panelu (tento parametr se nazývá *homogenita (rovnoměrnost) podsvícení*).

Poznámka

Rovnoměrnost podsvícení se testuje pomocí optické sondy. Výsledky se často vyskytují v srovnávacích testech ve formě 3D grafu nebo 2D obrazu, kde je intenzita podsvícení vyjádřena různými barvami.

Na obrázku vpravo vidíme výsledek pro obrazovku IPS panelu Dell U2311H. Bílá barva označuje místa podsvícená maximálně (100 %). V tomto konkrétním případě není podsvícení zcela rovnoměrné, ale ani v „horších“ místech to není až tak špatné, zelená barva označuje 80 %.



Obrázek 9.7: Ukázka homogenity podsvícení[5]

[5] http://pctuning.tyden.cz/hardware/monitory-lcd-panely/18123-dell-u2311h-e-ips-panel-za-velice-prijemnou-cenu

Další informace

- http://www.zive.cz/clanky/jak-testujeme-lcd-monitory/sc-3-a-139067/default.aspx
- http://www.tvfreak.cz/metodika-testovani-lcd-televizi/2248-2

Také u LCD se nastavuje *obnovovací frekvence*. Podobně jako u CRT monitorů, i zde je obraz v pravidelných intervalech (podle obnovovací frekvence) obnovován podle obsahu videopaměti, jen místo „řádkování" by se při nižší frekvenci mohl projevovat spíše jako pravidelné snižování a opětovné zvyšování jasu.

Tedy i u LCD panelů bychom měli mít nastavenu vhodnou obnovovací frekvenci. Příliš nízká unavuje oči, příliš vysoká znamená zbytečně zvýšenou spotřebu energie. Pro běžný provoz je ideální frekvence přibližně 60 Hz, pro hraní her je lepší vyšší frekvence.

Aktivní a pasivní LCD. Rozlišujeme dva základní typy LCD displejů – aktivní a pasivní.

TFT (Thin-Film Transistors) jsou „aktivní" LCD, dnes nejběžnější. Pod jednotlivými pixely jsou tranzistory, některé technologie mají i dva tranzistory na každou základní barvu, tj. 6 na pixel. Je vyžadováno aktivní podsvícení, a to většinou CCFL trubicemi nebo LED diodami. Speciálním případem jsou OLED displeje, které jsou samoemitující (vytvářejí samy světlo), tedy nepotřebují zvláštní podsvícení.

Pasivní LCD fungují trochu jinak. Každý řádek má jeden tranzistor, každý sloupec má jeden tranzistor. Tento typ LCD má pomalejší reakce, méně barev, vyskytuje se rušení mezi vodiči, ale zato je levnější a má menší spotřebu energie.

Dále se budeme věnovat většinou TFT (aktivním) LCD.

TFT se dělí do podskupin podle použité technologie – funkčnosti vrstvy tekutých krystalů. Základní druhy jsou TN, IPS, VA a OLED, druhý a třetí typ se opět vyskytují v několika variantách. Odlišnosti jsou především v principu zacházení s tekutými krystaly, následně v kvalitě zobrazení barev, pozorovacích úhlech, době odezvy, atd.

Základní princip. Tekutý krystal je materiál, který při průchodu elektrického proudu mění svou molekulární strukturu. LCD funguje na principu *propouštění a blokování světla* (z vnějšího zdroje nebo světla ze zadního či bočního podsvícení), blokování či naopak propouštění světla provádějí právě tekuté krystaly změnami své molekulární struktury do propustné nebo nepropustné formy (nebo něco mezi tím). Molekuly tekutých krystalů se při průchodu elektrického proudu srovnají s mikroskopickými drážkami na elektrodách do spirálové či jiné pravidelné struktury (tak vytvoří krystal).

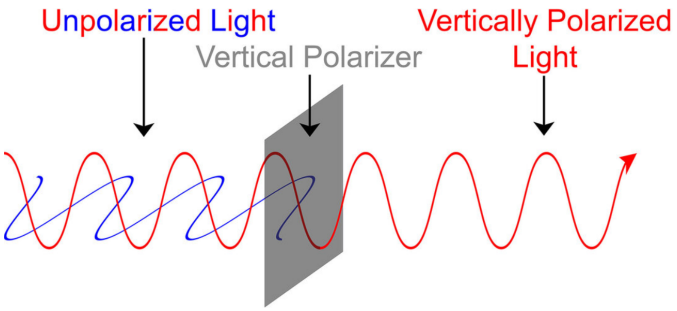
Barva jednoho pixelu (resp. subpixelu pro základní barvu) je ovlivněna následující posloupností:

světelný zdroj ⇒ vertikální polarizační filtr ⇒ průhledná elektroda ⇒ tekuté krystaly ⇒ průhledná elektroda ⇒ horizontální polarizační filtr ⇒

Poznámka

Trochu fyziky na vysvětlení: světlo není nic jiného než stream fotonů, které se pohybují kmitavým pohybem (v určité frekvenci) v rovině (jsme v 3D, tedy tato rovina může být různě nakloněná, fotony nemusejí nutně kmitat svisle) v určitém směru. Vertikální polarizační filtr je vlastně bariéra s vertikálními (svislými) štěrbinami. Jen ty fotony, které (mířící vodorovně) kmitají ve svislém směru (tj. rovina, ve které oscilují, je vertikální souběžně se štěrbinami) projdou – napasují se do některé štěrbiny, ostatní neprojdou. Vertikální polarizační filtr vidíme na obrázku 9.8.

[6] Zdroj: https://www.diyphotography.net/what-is-cross-polarization-and-how-you-can-use-it-in-your-photography/



Obrázek 9.8: Funkce vertikálního polarizačního filtru[6]

Horizontální polarizační filtr má vodorovné štěrbiny, tedy projdou jen ty fotony, které kmitají ve vodorovné rovině. Tedy pokud foton projde vertikálním filtrem a bez jakéhokoliv dalšího zásahu přijde k horizontálnímu, logicky nemůže projít. Ovšem jestli v oblasti mezi oběma filtry změní polarizaci (dojde k odrazu od tekutého krystalu), pak je určitá pravděpodobnost, že se jeho polarizace změní o 90 stupňů a tedy projde i tím horizontálním. Natočení krystalů se u TFT řídí pomocí tranzistorů pod pixely.

Poznámka

LCD panely jsou digitální, proto je lepší je připojovat přes digitální rozhraní (DVI, HDMI, DisplayPort). Pokud jsou připojeny přes analogové rozhraní (VGA), dochází ke zbytečným konverzím: D→A→D, což může mít vliv na kvalitu výstupu (ovšem v kanceláři a často ani doma to celkem nevadí).

Údržba LCD panelu: Rozhodně nepoužíváme prostředky na čištění oken ani papírové ubrousky! Postupujeme následovně:

- panel vypneme a necháme *zcela vychladnout* (kdybychom to neudělali, získali bychom „dekorativní“ šmouhy),
- čistíme buď speciálním vlhčeným čisticím ubrouskem určeným pro LCD panely, anebo použijeme kvalitní jemnou utěrku z mikrovlákna (kvalitní, pozor na obšití hrubou nití, mohlo by poškrábat povrch), kterou buď namočíme ve vlažné vodě a důkladně vyždímáme, anebo použijeme sprej určený k těmto účelům,
- necháme proschnout, pak teprve můžeme používat.

Pokud povrch panelu není skleněný (především u notebooků se používá částečně pružný plast), neměli bychom moc tlačit, větší tlak může poškodit krystaly.

CRT monitory lze čistit podobně, včetně vychladnutí.

9.5.4 Druhy TFT panelů

TN (Twisted Nematic), B-TN je nejstarší masově rozšířená TFT technologie, vyznačuje se horšími pozorovacími úhly, což je způsobeno šroubovitým uspořádáním molekul tekutých krystalů.

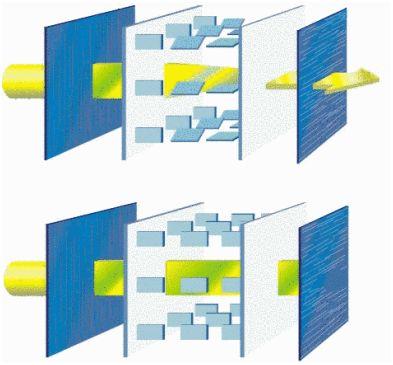
Podle výše uvedené posloupnosti ovlivňující barvu pixelu platí u TN následující:

- Neprochází proud: krystaly jsou v „chaotickém stavu“, světlo od prvního filtru je jimi rozptýleno do všech stran a odráží se všude, včetně původního směru v různých polarizacích – projde i druhým filtrem. Pixel tedy svítí.

- Pokud je (maximální) napětí: tekuté krystaly jsou natočeny tak, že fotony procházejí kolem nich s neměnnou polarizací, tedy druhým filtrem neprojdou (protože filtry jsou na sebe kolmé, frekvence, které prošly prvním, neprojdou druhým). Pixel nesvítí.

Z toho vyplývá, že při snížení napětí projde více světla. Mezi elektrickým napětím a jasem je *nepřímá úměrnost*, tedy se zvyšujícím se napětím klesá světlost bodu (bez napětí = bílá). Vadné pixely svítí bílou barvou, což je poměrně nepříjemné.

Pro TN je typické horší podání barev – gammut Adobe RGB ani náhodou (s tím souvisí menší barevná hloubka, obvykle jen 6 bitů na barvu/kanál, na větší hloubku je barva dopočítána ditheringem). Na druhou stranu mají tyto panely obvykle lepší odezvu než jiné typy TFT panelů.



Obrázek 9.9: Princip LCD[7]

Poznámka

Shrnutí: výhodou TN panelů je nízká cena a slušná doba odezvy (zvládá rychlé změny obrazu bez duchů), nevýhodou vše ostatní – horší podání barev, pozorovací úhly jsou malé (zejména vertikální – při pohledu zdola barvy dost výrazně tmavnou), vadný pixel svítí. S těmito panely se setkáváme spíše u velmi levných notebooků.

IPS (In-Plane Switching) vyvinutá společností Hitachi je také nazývána Super-TFT. Oproti TN nejsou molekuly při zobrazení bílé uspořádávány šroubovitě, ale podélně – důsledkem jsou lepší pozorovací úhly (kolem 170 stupňů), ale na druhou stranu kontrast lepší nebývá (dokonce i naopak) a doba odezvy je u (původní) IPS obecně nízká. Novější IPS panely už mají odezvu poměrně slušnou, ale záleží na výrobci a konkrétním modelu. Nevýhodou může být o něco vyšší energetická náročnost než u TN.

Podání barev je mnohem lepší než u TN. Barvy i ve srovnání s následující technologií vypadají živěji, jsou věrnější (až na velmi tmavé tóny), bohužel kontrast tak výborný není. Vadný pixel je černý, neruší v obraze. Technologie výroby je náročnější, proto (kromě jiného) bývají IPS panely dražší. Typické využití je v DTP (práce s grafikou), nicméně IPS panely čím dál více najdeme v běžných monitorech a noteboocích.

PLS (Plain-to-Line Switching) je varianta IPS od Samsungu, která se ve svých vlastnostech vpodstatě kryje s výše uvedeným o IPS panelech. Na rozdíl od jiných IPS má ještě o něco lepší pozorovací úhly, podání barev ve světlých tónech, a Samsung tuto technologii lépe připravil pro ohebné či ohnuté panely. PLS může být zajímavé také pro dotykové displeje (při dotyku se nemění barva pixelů, po přejetí prstem či perem tedy nezanechává stopu). Také výroba je údajně finančně optimálnější.

Poznámka

Shrnutí: výborné podání barev, vadný pixel je černý (neruší obraz), pozorovací úhly jsou slušné, doba odezvy je mírně horší než u TN (papírově), ale novější varianty tento parametr slušně dotahují. Dokonce můžeme narazit na IPS monitory s herními parametry. Nevýhodou je o něco vyšší cena.

VA (Vertical Align) je dílem společnosti Fujitsu. Na rozdíl od TN a IPS jsou molekuly (krystaly) uspořádány do „stromečků“, které se při průchodu proudu mírně „rozšiřují“. Původní VA technologie se vyznačovala špatnými pozorovacími úhly (lepšími než TN, horšími než IPS), což se hodně zlepšilo u nových

[7] Zdroj: http://pctuning.tyden.cz/component/content/4110?task=view

variant – MVA a PVA. Je typické, že u VA technologií jsou vertikální a horizontální pozorovací úhly stejné.

Výhodou oproti TN je vyšší kontrast, vadný pixel je tmavý, barvy jsou o něco lepší než u TN (záleží na variantě, některé varianty VA jsou dokonce srovnatelné s IPS). Obecně se dá říct, že v mnoha parametrech jsou VA prostě někde mezi TN a IPS (pozorovací úhly, podání barev, odezva). Ovšem lepší než obojí jsou v kontrastu a naopak horší než obojí jsou v odezvě (ale u obou parametrů záleží také na konkrétních zařízeních).

Ovšem v jedné věci VA panely excelují: je možné je celkem snadno použít i pro jiné tvary než „desky“, tedy mezi prohnutými monitory najdeme většinou VA panely.

Varianta *S-PVA* (Super-PVA) je v současné době považována za špičku v LCD technologiích, často mívá barevnou hloubku 10 bitů na kanál, velmi dobrý kontrast a podání černé.

Poznámka

Shrnutí: oproti TN mají VA panely řádově lepší podání barev (ty nejlepší dosahují v barevném podání kvalit IPS panelů), vadný pixel je černý, kontrast je obecně lepší než TN i IPS, doby odezvy přibližně odpovídají IPS, pozorovací úhly jsou v různých směrech stejné a dosahují téměř parametrů IPS.

VA panely mají obecně lepší poměr mezi cenou a kvalitou při velmi dobrém podání barev a kontrastu, a pokud nepotřebujeme až tak dobrou odezvu, jsou výbornou volbou. Pokud chceme prohnutý panel, pak určitě volíme VA.

QLED (Quantum LED) od Samsungu je ve skutečnosti vylepšení TN technologie o tzv. kvantové tečky (quantum dots), technologie je určena zejména pro televize a má konkurovat OLED displejím (viz dále).

Některé vlastnosti TN dokázal Samsung v QLED velmi zlepšit: například pokrytí barevného prostoru je excelentní, v průměru lepší než u OLED, dále dosahují vysokého jasu (tj. v prosvětlených místnostech je obraz na displeji celkem dobře viditelný). Naopak některé nevýhody TN zůstávají a v těchto parametrech je QLED horší než OLED, IPS atd., například zobrazování černé barvy je problém (obrazovka prostě pořád alespoň trochu svítí) a kontrast taky není nejlepší. Problém je i „vypalování“ statického obrazu.

Další informace

- https://www.gamersnexus.net/guides/1890-panel-comparison-tn-ips-pls-va-crt
- http://notebook.cz/clanky/technologie/2010/obrazovky-notebooku-cast1
- http://notebook.cz/clanky/technologie/2010/obrazovky-notebooku-cast2
- http://www.svethardware.cz/technologie-tft-lcd-displeje/7555-2
- http://www.tftcentral.co.uk/articles/panel_technologies.htm

Čeho si všímat u LCD panelů:

- použitá technologie,
- úhlopříčka, poměr stran obrazovky a rozlišení,
- odezva v ms (rozhodně pod 10 ms, čím menší, tím lepší),
- pozorovací úhly, kontrast, jas (v cd/m^2, vyšší číslo znamená vyšší jas), počet zobrazovaných barev (barevná hloubka), atd.,
- spotřeba v provozu a v pohotovostním režimu,
- vestavěné reproduktory, polohovatelnost stojanu, rozhraní pro připojení ke grafické kartě, zda obsahuje USB hub, atd. další přídavné funkce.

9.5.5 OLED technologie

OLED (Organic Light-Emitting Diode, organické LED) využívají technologii organických elektroluminiscenčních diod. Organický proto, že tato technologie je založena na organickém materiálu, uhlíku. Obrazovka je tvořena diodami, které po zavedení elektrického proudu vyzařují světlo, tedy na rozdíl od předchozích jmenovaných *není nutné podsvícení* ani filtr.

Mezi anodou a katodou je emisní organická vrstva (luminofor), která při průchodu elektronů „pouští" fotony. To vše je buď uzavřeno mezi dvě skla, nebo sklo a jiný materiál, nebo třeba naneseno na fólii, čímž vznikne velmi tenký a pružný displej.

S touto konstrukcí souvisejí výhody jako široké pozorovací úhly, možnost dosažení vysokého jasu, rychlá doba odezvy, dobrý kontrast, a dokážou zobrazit skutečně černou barvu (to je při používání podsvícení celkem problém, což zde nehrozí), obecně reprodukce barev je výborná, OLED panel může být tenčí než běžné LCD.

Protože různé pixely mají vlastní diody, je spotřeba závislá na tom, jak momentálně tyto diody svítí. Při zobrazování „průměrného obrazu" jsou OLED poměrně úsporné, ale pokud se delší dobu emituje hodně světlý obraz, je to na spotřebě znát.

Pixel je tvořen modrým, červeným a zeleným subpixelem s vlastními LED. Nevýhoda je nižší životnost zejména modrých subpixelů, což se může řešit třeba tak, že modrý subpixel je větší (tj. jeho opotřebení nebude tak brzy znát). Nevýhodou je také dražší výroba než u výše jmenovaných.

Existují dva základní typy OLED: pasivní a aktivní.

PMOLED (Passive Matrix OLED) je pasivní varianta, tedy máme mřížku vodičů (řádky a sloupce). Pixel ja vždy na průniku řádku a sloupce, na místě pixelu jsou na vodiče napojeny anoda a katoda a mezi vodiči je organický materiál. Vodiče určují, zda má anoda emitovat elektrony a tedy organický materiál vytvářet fotony. Čím větší proud, tím více fotonů, a tedy pixel více září.

Nevýhodou je, že pro aktivaci třeba jen jediného pixelu je nutné pustit poměrně velký proud do celého řádku a sloupce (dokonce tolik, že u jiného typu displeje by to znamenalo hodně vysoký jas) a zobrazovací frekvence nemůže být moc vysoká (doba aktivace vodičů je závislá na počtu řádků, resp. sloupců). (Nejen) proto je toto řešení vhodnější spíše pro menší displeje s nižším rozlišením.

AMOLED (Active Matrix OLED) je aktivní varianta, kde každý pixel má vlastní tranzistor (alespoň jeden, spíše více spolupracujících) a jeho svítivost je řízena tímto tranzistorem nezávisle na jiných pixelech na řádku/sloupci.

Na rozdíl od pasivních variant má AMOLED kratší reakční dobu a spotřebovává méně energie, také má obecně lepší kontrast (černá by měla být opravdu černá, nesvítit). Ale na druhou stranu jsou AMOLED displeje na slunečním světle hůře čitelné a jejich výroba je náročnější.

Existují další varianty podle způsobu převádění energie na světlo, použitého nosného materiálu, atd. Například PHOLED (fosforeskující) s mnohem vyšší účinností při nižší spotřebě, pružné FOLED pro různě tvarované povrchy (včetně hledí helmy), oboustranné TOLED (transparentní), a další.

9.6 Tisk

Tiskárny patří k zobrazovacím zařízením stejně jako monitory, ale na rozdíl od nich zobrazují (tj. tisknou) výhradně v barevném modelu CMYK nebo některém odvozeném modelu. Na trhu existuje mnoho druhů tiskáren, které se navzájem liší především technologií použitou při tisku.

Mechanické tiskárny pracují na podobném principu jako psací stroje. *Typy* (znaky) se přetiskují přes barvicí pásku (i vícebarevná – různé barvy na stopách). Rozlišujeme

- jehličkové (dot-matrix) – bod je tisknut určitým počtem jehliček (9 nebo 24),
- znakové – pouze znaky, používány typy stejně jako u psacího stroje.

Jehličkové tiskárny dokážou pomocí jehliček utvořit nejrůznější znaky podle potřeby (včetně pseudografiky).

Mechanické tiskárny jsou velmi levné, ale hlučné, kvalita tisku je horší. Na druhou stranu dokážou pracovat i v prašnějším prostředí. V současné době je najdeme spíše v technických provozech.

Inkoustové tiskárny vytvářejí tisknuté objekty z malých kapiček inkoustu vystřikovaného na papír. Rozlišujeme dva typy:

- *piezoelektrické tiskárny* (InkJet) – v tiskové hlavě je trubička nebo destička z materiálu, který při průchodu elektrického proudu mění tvar; kapka inkoustu, která dopadne na prudce se vypouknuvší trubičku nebo destičku, je vymrštěna na papír,
- *tryskové* (BubleJet) – inkoust je v trysce nahříván, čímž se jeho část odpaří (plyn se rychle roztahuje), zbytek inkoustu je vymrštěn na papír.

Inkoust je v tiskové náplni – *cartridge*, barvy C (cyan, modrozelená), M (magenta, červenofialová), Y (yellow, žlutá) a K (black, černá). Obvykle jsou základní barvy odděleně.

Levnější tiskárny mají opravdu jen čtyři náplně (cyan, magenta, yellow, black), ale lepší tiskárny mívají víc zásobníků s různými barvami. Důsledkem je dokonalejší míchání barev, což člověk ocení zejména u fototisku. Trendem jsou také velké plnící zásobníky (tanky), což je výhodné v případě, že hodně tiskneme. Výhodou inkoustových tiskáren je kvalitní tisk (i fotografií), nevýhodou je občasné zanášení trysek (každá inkoustová tiskárna má mechanismus čištění trysek, který je startován obvykle stisknutím určité kombinace tlačítek nebo softwarově, rychlé čištění trysek se také provádí při zapnutí tiskárny).

Poznámka

U inkoustových tiskáren hrozí nebezpečí *vyschnutí inkoustu*, zvláště v tiskových hlavách. Proto je důležité občasné čištění trysek (od nánosu zaschlého inkoustu), a z toho důvodu také inkoustové tiskárny nejsou vhodné tam, kde se tiskne například jen jednou za půl roku (i když netisknutí po delší odstávce lze často vyřešit několikanásobným spuštěním čisticího mechanismu).

Většina tiskáren při svém vypnutí „parkuje“ tiskovou hlavu do takové polohy, ve které je riziko vyschnutí nižší. Pokud je inkoustová tiskárna vypnuta příliš náhle (například výpadkem proudu nebo použitím centrálního vypínače), může se stát, že tiskárna nestihne hlavu zaparkovat. Pak může opravdu dojít k poměrně rychlému zatuhnutí inkoustu v tryskách a je třeba je vyčistit. Proto bychom měli mít na paměti, že před použitím centrálního vypínače je nutné tiskárnu skutečně vypnout tlačítkem.

Cartridge do inkoustové tiskárny může být *originální* (od výrobce tiskárny) nebo *neoriginální* (od jiného výrobce). O originálních náplních se všeobecně ví, že určitě nepoškodí tiskárnu, ale na druhou stranu bývají drahé.

O neoriginálních náplních se tvrdí, že existuje určité riziko poškození tiskárny (většinou to není pravda, ale výjimka potvrzuje pravidlo), ale cena bývá nižší. Když chceme koupit neoriginální náplň, měli bychom si třeba na internetu ověřit, zda s výrobky příslušného výrobce náplně nebyly problémy. Mezi výrobci neoriginálních náplní také najdeme solidní firmy, situace se zlepšuje.

Měli bychom si uvědomit, že mnozí výrobci do dokumentace tiskárny připisují (více či méně zřetelně), že použití jiných než originálních náplní bude mít za následek ztrátu záruky. Dokonce se objevily případy (HP a další), kdy po detekování neoriginální náplně tiskárna odmítla tisknout.

Poznámka

Bohužel se setkáváme také s jiným problémem – obsahem cartridge. Výrobce náplně udává určitý obsah, ale může se jednat o nepravdivou informaci. Zvláště když je obal neprůhledný, nelze si pohledem ověřit, kolik barvy je v náplni ve skutečnosti.

U inkoustových tiskáren jsou dnes hodně skloňovaným tématem jejich *tiskové náklady*. Vždy, když kupujeme novou tiskárnu, bychom se kromě dalších parametrů (včetně ceny tiskárny) měli zajímat o tiskové náklady, zvláště když budeme tisknout hodně. Může totiž nastat situace, kdy náhradní náplně do tiskárny stojí stejně nebo dokonce více než samotná (obvykle velmi levná) tiskárna.

Poznámka

Výrobci často prodávali inkoustové tiskárny pod cenou a pokoušeli se zisk dorovnat právě z prodeje originálních tiskových náplní, které však nedokázaly konkurovat levnějším, třebaže méně kvalitním neoriginálním náplním. Situace se postupně stávala neúnosnou jak pro výrobce, tak i pro zákazníky, tedy výrobci postupně přišli s jinými řešeními. Například u některých tiskáren se můžeme setkat se samostatně prodejnými inkousty v lahvičkách nebo celými plnícími sadami, kdy si uživatel může sám do cartridge nalít inkoust.

Tepelné tiskárny s přímým tiskem (termotiskárny) používají speciální papír, který působením tepla ztmavne. Princip:

- u monochromatických (jednobarevných) se jehličky v tiskové hlavě nahřejí a úderem na papír se vytvoří tisknutý symbol,
- u barevných se používají fólie s barevným voskem (pro různé základní barvy), které se zároveň s papírem postupně protahují kolem tiskové hlavy, vosk se pak teplem přenese na papír.

V současné době se tento typ tiskáren používá v monochromatické podobě především ve faxech, supermarketech (pokladny) nebo automatech pro „potvrzenky“.

Nevýhodou tohoto typu tisku je malá stabilita tisku (postupně bledne), výhodou je, že jediné, co musíme doplňovat, je papír (obvykle v kotoučích, případně kotouč se samolepícími štítky). Papír ovšem musí být speciální, s vrstvou citlivou na světlo.

Tepelné tiskárny termosublimační slouží především k tisku fotografií. K zobrazení se také používá teplo, ale kromě papíru (obvykle fotopapír) potřebujeme také fólie s barevnou vrstvou. K jednomu papíru se vždy spotřebuje jedna fólie.

Fotografie vytisknuté termosublimační tiskárnou jsou téměř tak kvalitní jako od nejkvalitnějších inkoustových tiskáren (detaily nebývají tak dobře zvládnuté, ale fotografie působí celkově přirozeněji), navíc je fotografie opatřena ochrannou povrchovou vrstvou, která zajišťuje delší životnost a chrání proti důsledkům otisků prstů. Nevýhodou je větší cenová náročnost tisku.

S termosublimačními tiskárnami se dnes setkáváme především ve fotokioscích (pro samoobslužné focení). Lze pořídit také termosublimační tiskárnu na domácí použití.

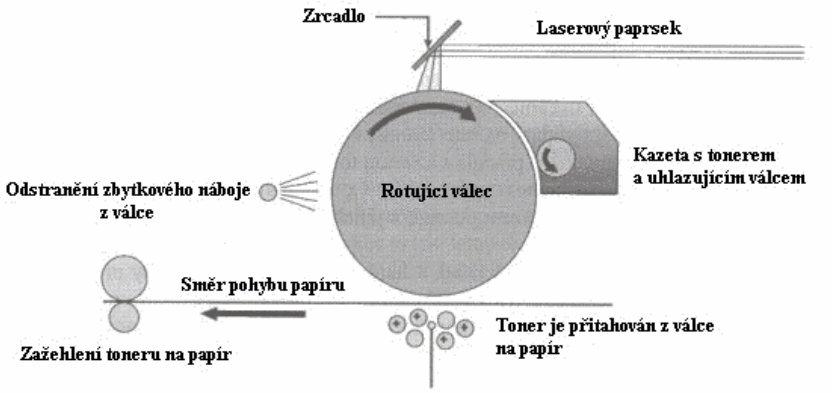
Laserové tiskárny používají pro podporu zápisu laser (ten ale rozhodně nemíří na papír, došlo by ke vznícení). Tisk probíhá takto:

- laserový paprsek dopadne na válec potažený fotocitnou vrstvou, na místě dopadu se změní elektrický náboj,
- na místech se změněným elektrickým nábojem ulpívá *toner* (barvivo ve formě jemných zrnek, prášek),

- válec s barvou na „ozářených“ místech se obtiskuje na papír, na něm ulpí barvivo,
- barvivo se na papíru zafixuje (zažehlí) zahřátím (používá se zahřátý válec).

Laserové tiskárny jsou tiché, rychlé a tisk je obvykle kvalitní, alespoň co se týče textu a jednoduché grafiky. Nevýhodou může být nízká životnost fotocitlivého válce (to je ovšem relativní, záleží na vytíženosti tiskárny).

Laserové tiskárny (ani barevné) obecně nejsou vnímány jako tiskárny pro tisk fotografií (fotografie lze tisknout, ale pouze u nejkvalitnějších modelů jsou opravdu pěkné), především fotografie nevypadají moc přirozeně. Odolnost tisku (zvláště ve vlhkém prostředí) není vysoká.



Obrázek 9.10: Schéma laserové tiskárny[8]

Typické použití je tam, kde se velmi hodně tiskne – samotné tiskárny bývají dražší, ale tiskové náklady jsou nižší než u inkoustových tiskáren. Takže obvyklým místem pro jejich využití jsou především kanceláře.

LED tiskárny pracují na podobném principu jako laserové, ale místo laseru je použita řada LED diod (pro každý bod na řádku jedna, typicky několik tisíc na řádek) se zaostřovacím mechanismem. Válec je fotocitlivý, tedy světlo o příslušné vlnové délce stačí na změnu elektrického náboje, teplo není generováno.

Oproti laserovým tiskárnám mají LED tiskárny výhodu ve vyšší spolehlivosti (méně pohyblivých prvků), menších rozměrech a nižší spotřebě energie.

Multifunkční zařízení jsou vlastně kombinace tiskárny a scanneru, spojením těchto dvou funkcí je lze použít i ke kopírování z papírové předlohy na jiný papír (text se nejdřív naskenuje do vyrovnávací paměti zařízení, a pak vytiskne). Další obvyklou doprovodnou funkcí je fax. Existují multifunkce založené na inkoustových nebo laserových tiskárnách, v různých cenových a kvalitativních relacích.

Vlastnosti tiskáren. Co nás u tiskáren zajímá:

- hustota tisku (DPI – dots per inch),
- rychlost tisku (PPM – pages per minute),
- tiskové náklady (cost per page – náklady na tisknutí jedné stránky),
- tisk fotografií – speciální papír, barvy,
- podavače papíru, taky gramáž papíru (jestli dokážou tisknout i na silnější papír),
- rozhraní – většinou USB, případně síťové rozhraní (funkce síťové tiskárny),
- displej (také jeho velikost), způsob ovládání, ergonomie ovládání,
- zda dokáže tisknout z USB, SD karty, jaké formáty souborů tiskne z těchto médií (jestli jen obrázky nebo například i PDF), atd.

[8] Zdroj: http://www.fi.muni.cz/usr/pelikan/ARCHIT/TEXTY/TISK2.HTML

Kapitola 10

Napájení a chlazení

Rychlý náhled: V této kapitole se věnujeme napájení a chlazení – jak funguje zdroj, jaké jsou jeho obvyklé parametry, jak fungují akumulátory („baterie“) notebooků, co je důležité při chlazení komponent počítače.

Klíčová slova: Zdroj, ATX, TFX, SFX, certifikace zdroje, účinnost, PFC, akumulátor, Ni-MH, Li-Ion, Li-Pol, aktivní a pasivní chlazení, aktivně-pasivní chladič, heatpipe, teplovodivá pasta, ACPI.

Cíle studia: Po prostudování této kapitoly se naučíte, jak jsou napájeny jednotlivé komponenty v počítači nebo jiném výpočetním zařízení, jak funguje akumulátor v mobilních zařízeních a jak s ním zacházet, a také jak se chladí komponenty v počítači.

10.1 Napájení

Každý počítač má *zdroj*, který energii nevyrábí, ale *transformuje* ze střídavého proudu (AC, z elektrické zásuvky) na stejnosměrný proud (DC, ten potřebuje počítač).

Důležitou funkcí zdroje je také řešení občasných *energetických výkyvů* (hlavně přepětí, které by mohlo poškodit komponenty v počítači, ale také podpětí). Pokud zdroj dokáže ustát i skutečný výpadek proudu, označuje se jako UPS (nepřerušitelný zdroj napájení). Ovšem ani UPS nedokážou „táhnout“ napájení věčně, ale určitá doba zajištění napájení při výpadku je garantována.

Zdroje se nacházejí v několika různých provedeních. Kromě běžného zdroje do desktopu (ATX) existují nižší varianty (TFX), zdroje podle standardu SFX jsou zase určeny pro minipočítače, u notebooků a některých All-In-One počítačů najdeme externí zdroj (to je ta „krabička“ na připojeném kabelu, trafo).

10.1.1 Jak zdroj funguje

Z energetické sítě zdroj odebírá střídavé napětí na 230 V, které transformuje na stejnosměrné napětí, kterým se průběžně nabíjí *kondenzátor* (tato součást zdroje uchovává energii pro odběr dalšími komponentami). Napětí v kondenzátoru však stále ještě zůstává hodně velké (několik set V), ale integrované obvody, které jsou obvykle v počítači napájeny, vyžadují naopak napětí velmi nízké (několik voltů). Proto za kondenzátorem následují ještě součástky měnící vysoké stejnosměrné napětí na nízké podle požadovaných hodnot.

Ze zdroje jde energie do tzv. *větví*, na které se připojují jednotlivé komponenty. Každá větev je specifická především napětím, které distribuuje. Obvykle existují 3,3V větve, 5V větve (+5V a -5V) a 12V větve (+12V a -12V). U každé z větví je také důleživým údajem možné zatížení, které zvládne. Například větev 3,3V obvykle unesou zátěž někde mezi 20 a 30 A.

Vedle větví je důležitá také kabeláž, resp. konektory. U kabeláže je důležitým parametrem délka kabelů (kam až „dosáhnou“), minimální vyžadovaná délka ze specifikace je 26–28 cm, což na desktopu opravdu nedostačuje (ideálně 45–50 cm, u bigtower více). Konektory bývají většinou typu *molex*, tedy sdružení sudého počtu „kostiček“. Kabely se skládají z více žil, do každého pinu v molexu vede jedna žíla. Žíly jsou barevně odlišeny podle větví.

Hlavní napájecí kabel pro základní desku má obvykle 24pinový konektor. Dále potřebujeme dostatek 4pinových, 6pinových nebo 8pinových (nebo 4+4) konektorů 12V větve (procesor, PCIe karty, apod.) a další. Napájet je třeba také pevné disky, optické mechaniky a cokoliv dalšího v počítačové skříni. Mnohé dedikované grafické karty vyžadují dodatečné napájení. Čipová sada je obvykle napájena z 3,3V větve, větev 5V je určena především pro PCI karty a staré procesory (novější využívají 12V větev).

10.1.2 Standardy

V současné době se setkáme především se zdroji podle specifikace *ATX* – jak vidíme, standardizace zdrojů je ve svých počátcích úzce provázána s form factory základních desek (a taky s nimi funkčně souvisí).

Standard ATX pro zdroje existuje ve více verzích. Nejčastěji se setkáme s verzemi ATX v2.x (místo „x“ číslo podverze) a ATX 12V x.y (opět místo „x“ a „y“ konkrétní čísla). Jednotlivé verze se liší minimálními požadovanými hodnotami pro možné zatížení větví, minimální požadovanou účinností, poskytovanými konektory, požadavky na ochranu před přepětím zkratem, přetížením přehřátím apod. Seženeme také zdroje podle jiných specifikací, například do hodně nízkých skříní (například Mini-ITX) existují zdroje TFX. ATX 12V 2.x používá „kostičkové“ konektory: 24pinový hlavní pro napájení základní desky, 4pinový pro přídavné napájení základní desky, další PCIe konektory pro grafické karty a samozřejmě SATA, Molex a další.

V roce 2020 se objevil nový standard ATX12VO („O“ jako „only“), jehož hlavní konektor je 10pinový s menšími „kostičkami“.

Existují také *certifikace zdrojů*. Setkáváme se s několika certifikacemi lišícími se především požadovanou účinností zdroje při konkrétním zatížení. Hlavní vlastnosti certifikovaných zdrojů jsou naznačeny v tabulce 10.1, nejnověji existuje také certifikace 80 Plus Titanium určená pro tzv. redundantní zdroje[1] s účinností ještě vyšší než 80 Plus Platinum.

Pokud má zdroj vysokou *účinnost*, znamená to nejen, že stačí na svou práci a neodebírá zbytečně mnoho energie, ale také, že se méně zahřívá a tudíž větrák na něm se nemusí tak rychle otáčet (což má také vliv na hlučnost). Tedy účinnost můžeme chápat jako obrácené a normované (na procenta) číslo k množství energetických ztrát na jednotlivých součástkách zdroje (čím větší jsou energetické ztráty, které se mění v teplo, tím menší je účinnost zdroje).

Další informace

Jak zjistit, zda a kterou certifikaci má vybraný zdroj? Kromě toho, že výrobce se touto informací celkem „chlubí“, to také můžeme zjistit na http://80plus.org/, resp. https://www.clearesult.com/80plus/.

[1] Redundantní zdroj je přídavný zdroj, který se používá pro napájení z druhého přívodu energie nezávislého na prvním, anebo pro napájení z centrálního záložního akumulátoru, není to totéž jako UPS.

		80 Plus	80 Plus Bronze	80 Plus Silver	80 Plus Gold	80 Plus Platinum	80 Plus Titanium
		Účinnost při daném vytížení:					
Pro 115 V:	10% vytížení	–	–	–	–	–	90 %
	20% vytížení	80 %	82 %	85 %	87 %	90 %	92 %
	50% vytížení	80 %	85 %	88 %	90 %	92 %	94 %
	100% vytížení	80 %	82 %	85 %	87 %	89 %	90 %
Pro 230 V:	10% vytížení	–	–	–	–	–	90 %
	20% vytížení	82 %	85 %	87 %	90 %	92 %	94 %
	50% vytížení	85 %	88 %	90 %	92 %	94 %	96 %
	100% vytížení	82 %	85 %	87 %	89 %	90 %	94 %

Tabulka 10.1: Srovnání účinnosti certifikovaných zdrojů při různých zatíženích, 230V EU Internal Non-Redundant[2]

PFC (Power Factor Correction) je část zdroje, jejímž úkolem je „uhlazovat“ křivku proudu do tvaru sinusoidy (tj. křivka se sinusovým průběhem, funkci „sin“ určitě každý z nás zná). PFC není ani tak důležitý pro komponenty počítače, ale naopak pro okolní síť důležitý je, protože tady jde o střídavý proud. V odběru proudu se totiž vyskytují špičky (typicky při zapnutí počítače odběr najednou poskočí hodně nahoru), vzniklá deformace sinusového napětí v rozvodné síti může způsobit rušení v této síti.

PFC jsou tedy důležité hlavně tam, kde se v jeden okamžik zapíná (nebo zvyšuje svůj odběr) více zařízení najednou. Pokud například zapneme v jeden okamžik (plus mínus) více počítačů bez PFC, může dojít i k vyhození jističů.

V současné době prakticky všechny prodávané zdroje jsou PFC vybaveny, protože podle norem (norma EMC o elektromagnetické kompatibilitě) sice nepředepisuje povinnost mít PFC, ale požadavky na splnění některých parametrů a prahových hodnot se bez PFC jen těžko dají splnit.

Ve zdrojích najdeme buď pasivní nebo aktivní PFC (tento údaj je u zdroje vždy uveden). Pasivní jsou spíše v levnějších zdrojích, kdežto aktivní PFC jsou typické pro téměř všechny kvalitní zdroje. Oba typy dělají prakticky totéž, ale zatímco pasivní PFC je sestaven z pouze průchodových součástek bez vlastního odběru (obvykle tlumivka na vstupu zdroje), aktivní PFC obsahuje také aktivní součástky s vlastním odběrem (tranzistor v kombinaci s cívkou apod.). Aktivní PFC je účinnější než pasivní.

10.1.3 Vlastnosti zdroje

Výkon zdroje. Jak víme, důležitým parametrem je účinnost při daném zatížení (účinnost bývá často označena právě výše uvedenými certifikacemi).

Další důležitý parametr, o kterém rozhodujeme podle způsobu a intenzity používání počítače, je výkon. Jak jsme výše viděli, účinnost zdroje závisí na momentálním vytížení, a vytížení je zase do určité míry závislé na výkonu zdroje.

Výkon zdroje se udává ve Wattech a je to jeden z nejdůležitějších parametrů, o které bychom se měli zajímat. Pro nenáročné využití stačí 300W zdroj (hodně záleží na použitých komponentách a jejich energetické náročnosti). Pokud však máme výkonnější procesor s vyšší hodnotou TDP, pak volíme zdroj s výkonem kolem 400–500 W, a jestliže máme špičkovou (například herní) sestavu s výkonnou dedikovanou grafickou kartou nebo dokonce více grafikami, pak je třeba použít zdroj ještě výkonnější (700–800 W).

[2] Vytvořeno podle https://www.clearesult.com/80plus/program-details

Záleží také na počtu připojených pevných disků (například RAID má logicky vyšší spotřebu než jediný disk), optických mechanik a dalších.

Poznámka

To, že provozování počítače se blíží hranicím výkonu našeho zdroje, poznáme podle různých indikátorů, například zamrzání počítače v některých situacích (například při hraní graficky a výpočetně náročných her), občasné samovolné restartování disků, apod.

Mohlo by se zdát, že je lepší pořídit si raději zdroj s velmi vysokým výkonem, abychom měli co největší prostor pro další rozšiřování či vylepšování sestavy. Měli bychom však vědět, že zbytečně vysoký výkon obvykle znamená nižší účinnost zdroje (zdroj je jen málo zatěžován, a při zátěži, která je po většinu doby využívání menší než 50 %, je účinnost obvykle nízká), což znamená plýtvání energií.

10.1.4 Akumulátor

Akumulátor (lidově baterie) notebooku či jiného podobného zařízení slouží k akumulování (uschování) energie, která už prošla zdrojem (tj. stejnosměrné napětí, a to spíše nižších hodnot). V jednom akumulátoru je obvykle několik článků (například může jít o 6článkový akumulátor).

Podíváme se na některé pojmy související s akumulátory.

Článek (galvanický článek) váže elektrickou energii, kterou lze chemickou cestou snadno uvolňovat ve formě elektrického pole (tj. tato energie je využitelná pro napájení el. zařízení). Existují dva základní typy článků – primární (nelze znovu nabít, dokáže pouze uvolňovat energii) a sekundární (lze opakovaně nabíjet a vybíjet).

Baterie se skládá z několika vzájemně propojených galvanických článků. Pokud je baterie tvořena ze sekundárních článků (tedy takových, které lze nabíjet), hovoříme o *akumulátoru*.

Paměťový efekt akumulátoru je jev, se kterým se setkáváme u některých typů akumulátorů (viz dále), pokud je nabíjíme, třebaže nebyly zcela vybity. U těchto typů akumulátorů dochází ke snižování kapacity v důsledku opakovaného nabíjení bez úplného vybití. Ale pozor – moderní akumulátory reagují jinak!

Formátování akumulátoru je důležité především u těch typů, které jsou náchylné k paměťovému efektu, u nich dokonce může vést k mírnému napravení snížení kapacity způsobeného paměťovým efektem. Provádí se tak, že několikrát (třikrát) necháme akumulátor zcela vybít a pak ho nabijeme na plnou kapacitu. U typů akumulátorů, které nejsou náchylné na paměťový efekt, může být za určitých okolností formátování taky užitečné (řekněme tak jednou za rok nebo dva, pokud je akumulátor hodně používán), ale nesmíme články vybít zcela (vždy bychom měli několik procent nechat).

Typy akumulátorů. Existuje několik typů akumulátorů použitelných pro nejběžnější mobilní zařízení (notebooky, netbooky, mobilní telefony apod.):

- *nikl-kadmiové* (Ni-Cd) – historická záležitost, už se s nimi prakticky nesetkáme,
- *nikl-metalhydridové* (Ni-MH) – také se už moc nepoužívají v mobilních zařízeních, ale můžeme je koupit ve formě tužkových a jiných „nabíjecích baterií“ do spotřební elektroniky,
- *lithium-iontové* (Li-Ion) – v noteboocích se s nimi dnes setkáme prakticky výhradně,
- *lithium-polymerové* (Li-Pol) – v některých mobilních telefonech.

Ni-MH (a také staré Ni-CD) akumulátory trpí nebezpečím paměťového efektu popsaného výše. Proto se u zařízení s tímto akumulátorem doporučovalo nabíjet až po úplném vybití a čas od času provést tzv.

formátování. Dnes se s nimi setkáváme ve formě (mikro)tužkových baterií, jejichž nabíječky obvykle obsahují funkci vybití před opětovným nabíjením (ostatně, tyto baterie obvykle kupujeme vybité).

Poznámka

V dnešních počítačích se používají Li-Ion a Li-Pol akumulátory, které *netrpí paměťovým efektem*, tudíž před opětovným nabíjením se nevybíjejí, ba dokonce úplné vybití jim škodí! Pokud například notebookový Li-Ion akumulátor budeme opakovaně zcela natvrdo vybíjet, snížíme jeho životnost, nebo ho dokonce zničíme. Na webu se v diskusních fórech občas objevují rady o regulaci kapacity akumulátoru jeho úplným vybíjením, ale ty jsou u Li-Ion a Li-Pol baterií zcela vedle (platily by pro Ni-MH). Ostatně, to je důvod, proč modernější operační systémy v zařízeních s Li-Ion přecházejí do režimu spánku nebo se úplně vypnou při zjištění nízkého stavu baterie (několik procent). Pokud se rozhodneme akumulátor „jednou za jeho život“ formátovat, nepomůže to, ale ani to až tak moc neuškodí. Jestliže ho formátujeme pravidelně třeba i jednou za měsíc, zbytečně tím snižujeme jeho kapacitu.

Údržba a používání akumulátoru. Každý akumulátor postupně ztrácí svou kapacitu, to je zcela normální. Konkrétní typ akumulátoru má přiřazen svůj údaj o maximálním počtu nabíjecích cyklů (tj. životnosti), ale i před dosažením této hodnoty kapacita postupně klesá. Tento pokles můžeme správným zacházením zpomalit (následující platí pro Li-Ion a Li-Pol baterie):

- Nikdy nenecháme akumulátor zcela vybít. Operační systémy jsou obvykle nastaveny tak, aby se včas před vybitím akumulátoru vypnul nebo uspal počítač, toto nastavení neměníme (nebo pokud tak operační systém není nastaven, provedeme nastavíme ho tak).
- Pokud často pracujeme připojeni k el. síti, nenecháváme akumulátor zbytečně přebíjet a raději ho vyjmeme (pokud to jde). Zbytečným přebíjením akumulátor není poškozován, ale je snižována jeho životnost (blížíme se k maximálnímu počtu nabíjecích cyklů, i když to není potřeba). To se projevuje postupným snižováním kapacity.
- Stejně jako u jiných komponent, také u akumulátorů platí, že v určitém rozmezí teplot pracují lépe než v jiném. Nejlépe pracují akumulátory při pokojové teplotě (kolem 20 °C), při vyšších teplotách se jejich účinnost a životnost zhoršuje. Akumulátor by při své činnosti neměl být ničím zakryt (vytváří teplo, to musí být odváděno vzduchem).

Pokud delší dobu akumulátor nepoužíváme (máme zařízení připojeno k el. síti), je třeba ho vhodně *skladovat*. Akumulátory Li-Ion skladujeme spíše v chladu (ale pozor, mráz naopak škodí), v teplotách nižších než pokojových. Přílišná vlhkost škodí, takže pokud se rozhodneme dát akumulátor do ledničky (ne mrazničky!), neprodyšně ho zabalíme. Před uskladněním bychom měli akumulátor nabít do kapacity cca 75 % a přibližně jednou za půl roku (podle celkové kapacity akumulátoru) znovu vybalit a dobít opět na 75 %. Akumulátory se postupně samy vybíjejí, proto je u Li-Ion důležité je takto udržovat.

Poznámka

V případě, že notebook s lithiovou baterií napájíme střídavě z elektrické sítě nebo z baterie, pak optimální zacházení spočívá v udržování stavu nabití v rozmezí 20–80 %, kdy se životnost (kapacita) téměř nesnižuje. Některé notebooky jsou vybaveny obvody pro inteligentní řízení stavu nabití, které baterii udržují přibližně kolem 50 % stavu nabití.

10.2 Chlazení

10.2.1 Princip chlazení počítače

Téměř všechny komponenty počítače potřebují ke svému běhu teplotu blízkou spíše pokojové teplotě, ale zároveň mnohé komponenty „topí“ – produkují teplo. Typickými topiči i při využívání chlazení jsou hlavně dedikovaná grafická karta (i přes 60 °C), procesor (s vyšším TDP i přes 50 °C), pevný disk (přibližně 35–50 °C), paměťové moduly a zdroj (přibližně 30–40 °C), a samozřejmě samotná základní deska (na některých místech může být ještě teplejší než procesor). Proto je chlazení počítače velmi důležitým tématem, teplota ve skříni by neměla překračovat 40–45 °C).

Pokud je chlazení nedostatečné, snižuje se životnost komponent, ale také se můžeme setkat s nestabilitami, zasekáváním a „padáním“ (obvykle samovolné restarty komponent či dokonce celého počítače).

Můžeme rozlišovat různé typy chlazení podle způsobu odvádění tepla (či „nosného média“). Nejběžnější je chlazení vzduchem nebo vodou (nebo jinou kapalinou s vhodnými vlastnostmi).

Základní opatření proti přehřívání:

- Důležité je *umístění počítače*. Kolem počítače by měl pokud možno proudit vzduch, topení by nemělo být hned vedle a sluníčko by nemělo na počítač svítit. Skříň počítače nepoužíváme jako odkladný prostor pro knihy, papíry a jiné izolační záležitosti. Skříň počítače může být podložena, aby nebyla přímo na podlaze (či dokonce koberci), pod ní by měla být pevná rovná podložka (třeba dřevěná).
- *Prach* je výborný tepelný izolant. Bohužel každé elektronické zařízení prach přímo přitahuje, proto i v dobře uklízené místnosti se dovnitř prach dostane. Proto je dobré občas počítač odprašnit (alespoň konzervou se stlačeným vzduchem – pozor, kam prach odfoukáváme, anebo vysavačem s nižšími otáčkami). Před odprašněním bychom měli počítač vypnout a odpojit od elektřiny.
- Proudění vzduchu uvnitř počítače často překážejí *kabely*. Naštěstí se už nepoužívají široké IDE (PATA) kabely, ale i tenčí kabely mohou být překážkou proudění vzduchu či působit turbulence. Proto bychom měli mít kabely uspořádané a ideálně svázané izolovanými drátky.

Jak odstranit prach zevnitř počítače:

- vysavač nastavený na nízké obrátky; pozor, nesmíme se hubicí dotknout žádného čipu (ideálně vůbec ničeho) – při proudění vzduchu hubicí dochází k tření a vzniká statická elektřina,
- vzduchové konzervy se dají koupit v každém obchodě s počítači, případně hypermarketu; jejich výhodou je silný proud vzduchu s tenkou násadou, dobře se „míří“,
- foukací balónek.

Poznámka

Pokud při odstraňování prachu z vnitřku skříně počítače použijeme vysavač, měli bychom předem zajistit všechny větráčky proti otáčením (například je oblepíme lepicí páskou), ale pokud to neuděláme, katastrofu to obvykle znamenat nebude. Po vysátí prachu nesmíme zapomenout všechny větráčky zase uvolnit a zkontrolujeme, jestli jsou všechny konektory tam, kde mají být, a že jsou zcela zasunuté.

Je důležité vědět, jak vlastně vzduch ve skříni proudí. V novějších skříních je obvykle vzduch nasáván bočními nebo čelními otvory (musí být volné!) a netěsnostmi skříně a vyfukován ventilátorem za zdrojem. Proudění vzduchu by neměly překážet jednotlivé komponenty, kolem kterých vzduch proudí a odebírá teplo, a samozřejmě ani kabely (nebo alespoň v co nejmenší míře). Rozmístění komponent je obvykle

alespoň částečně určeno Form Factorem základní desky (viz str. 55). Dedikované grafické karty s vlastním chladičem obvykle vyfukují teplý vzduch ven přes vedlejší slot v zadní stěně skříně.

10.2.2 Typy chladičů

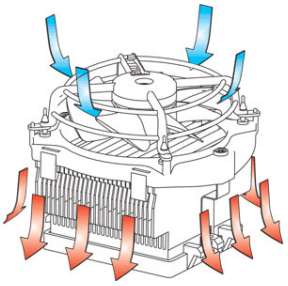
Pasivní chladiče jsou určeny pro tepelně méně náročné komponenty. Jsou obvykle založeny na využití konvekčního proudění vzduchu (teplý vzduch stoupá) nebo využívají jiný způsob nuceného proudění vzduchu (například v kombinaci s aktivním chladičem). Typickou a velmi důležitou vlastností pasivních chladičů je jejich tichý běh (jsou bez aktivních částí, nemají čím vytvářet hluk), poněkud horší vlastností je o něco nižší účinnost než u jiných způsobů chlazení.

U pasivních chladičů je důležitá velká chladicí plocha. Proto se obvykle jedná o celkem hustě žebrované součástky. Obvykle platí, že čím hustší žebrování, tím lepší chlazení, ale jen do určité míry. Příliš husté žebrování může vadit správnému průchodu vzduchu a působit turbulence.

Pasivní chladiče obvykle doprovázejí aktivní chladiče (viz níže), ale mohou být použity i samostatně (například na chipsetu).

Aktivní a aktivně-pasivní chladiče najdeme například u zdroje nebo často u procesoru, někdy i u dalších komponent. Jak bylo výše uvedeno, obvykle se vyskytují v kombinaci s pasivním chlazením.

Na obrázku 10.1 vidíme směr proudění vzduchu kolem kombinovaného chladiče nad procesorem (větráček nasává studený vzduch a fouká ho na procesor, ohřátý vzduch je pasivním chladičem odváděn pryč).



Obrázek 10.1: Aktivně-pasivní chladič[3]

Poznámka

Platí, že čím větší ventilátor (větší lopatky), tím účinnější je chlazení (ale záleží i na tvaru lopatek). Proto velké ventilátory se mohou při stejném chladicím výkonu otáčet nižší rychlostí. Souvisí to zejména s faktem, že lopatky ventilátoru vytvářejí aerodynamický hluk (rychle proudící vzduch je celkem hlasitý, s extrémem se setkáme například u letadel). Proto platí, že volíme spíše větší ventilátor (120×120 mm), který může pracovat na nižší rychlosti, a tím je i tišší než malé ventilátory (80×80 mm). Navíc menší ventilátory se rychleji zanášejí prachem. Ovšem větší ventilátor si můžeme dopřát jen tehdy, když se na dané místo vejde (tudíž před koupí větráku si změříme místo, do kterého ho chceme osadit).

Ventilátor se obvykle umísťuje za zdroj, kde slouží k vyfukování teplého vzduchu ven, a dále bývá často vepředu, kde naopak nasává studený vzduch a vhání ho dovnitř skříně. Dále bývá ventilátor na procesoru a případně i u dedikované grafické karty. Existují také přídavné ventilátory ve formě rozšiřující karty, které se mohou zasunout do PCI slotu, obvykle vedle dedikované grafiky.

Heatpipe [hi:tpaip] je hermeticky uzavřená měděná trubička naplněná takovou kapalinou, která dobře vede teplo (voda, ethanol apod.). Vnitřní povrch heatpipe je narušený, má jakousi houbovitou strukturu. Jedná se o pasivní chlazení.

Obrázek 10.2: Kombinace aktivního chladiče a heatpipes[4]

[3] Zdroj: http://pctuning.tyden.cz/navody/zaklady-stavba-pc/7167-zaklady_pc-chlazeni_a_tichy_pocitac
[4] Zdroj: http://pctuning.tyden.cz/navody/zaklady-stavba-pc/7167-zaklady_pc-chlazeni_a_tichy_pocitac

Bod varu kapaliny uvnitř heatpipe musí být spíše nízký (ideálně kolem 50 °C), čehož se dosahuje vhodnou volbou kapaliny (ethanol) nebo natlakováním (u vody). Na zahřívaném konci se kapalina odpařuje – tím odebírá teplo, na chladnějším konci pak kondenzuje a vnitřním „houbovitým“ povrchem se vrací k teplejšímu konci. Díky tomu nezáleží na konkrétní poloze a nasměrování trubiček, vypařování i vzlínání funguje i proti zemské gravitaci.

Heatpipes se vyznačují velmi vysokou účinností, třebaže jde v principu o pasivní chlazení. Setkáváme se s nimi obvykle v kombinaci s aktivním chlazením, ukázku vidíme na obrázku 10.2 (na boku je ventilátor coby aktivní chladič, dále zde najdeme běžný pasivní chladič – žebrování, dole jsou ve formě písmene „U“ heatpipes).

Heatpipes pomáhají chladit procesor, často i základní desku a další komponenty. U kvalitnějších paměťových modulů je také najdeme uzavřené v pouzdře kolem desky s čipy s tím, že paměťové moduly se obvykle nacházejí blízko procesoru a tedy jsou často přichlazovány chladičem procesoru.

Teplovodivá pasta se nanáší na plochu mezi chlazenou komponentou a chladičem (nejčastěji u procesorů). Jejím úkolem je ideálně spojit komponentu a chladič, aby mezi nimi nebyl vzduch, čímž se zlepší odvod tepla. Pastu nanášíme jen v tenké vrstvě (obvykle stačí kapka doprostřed procesoru), tak, aby došlo k plnému propojení, příliš velká vrstva může naopak odvod tepla zhoršit.

10.2.3 Sledování teploty

Moderní počítače běžně mívají u „rizikovějších“ komponent teplotní senzory, které lze softwarově sledovat. Sledování lze provádět v různých programech, i specializovaných na konkrétní komponentu. Kromě sledování také často můžeme nastavovat rychlost otáčení ventilátorů (jejich otáčky) – když je ventilátor příliš hlasitý a zároveň nehrozí nebezpečí přehřátí komponenty, můžeme otáčky tohoto ventilátoru mírně snížit.

Ve Windows můžeme použít například:

- SpeedFan[5] – regulace otáček ventilátorů, monitoruje teplotní senzory a přistupuje ke S.M.A.R.T.,
- CPUID HWMonitor[6] – na webu je celá sada nástrojů pro monitorování systému od firmy CPUID,
- HDD Health, HDDLife – pro pevný disk,
- Notebook Hardware Control[7] – především pro notebooky, včetně správy napájení,
- SiSoft Sandra Lite.

Typičtí výrobci chladičů jsou Arctic Cooling, Evercool, Nexus, Noctua, Thermalright, Xigmatek, Thermolab, Zalman a další.

Další informace

Další informace o chlazení: http://pctuning.tyden.cz/navody/zaklady-stavba-pc/7167-zaklady_pc-chlazeni_a_tichy_pocitac

10.3 Power Management

Power Management (správa napájení) představuje možnosti řízení spotřeby komponent počítače v určitých situacích. Existují dvě možnosti, jak lze správu napájení provádět: *APM* (starší, zcela v režii BIOSu) a *ACPI* (novější, v režii operačního systému).

[5] http://www.almico.com/speedfan.php
[6] http://www.cpuid.com/softwares.html
[7] http://www.pbus-167.com/

APM (Advanced Power Management) se konfiguruje v BIOSu v sekci, která je obvykle nazvaná „Power Management Setup“ nebo podobně. Definuje několik režimů, mezi kterými může procesor přecházet (Enabled, Standby, Suspend to RAM, Suspend to Disk, Off).

ACPI (Advanced Configuration and Power Interface) sice také využívá BIOS (pro práci s konfigurací základní desky) a používá hardwarové registry (v čipsetu), ale konfigurace se provádí v operačním systému, například ve Windows to je v ovládacím panelu *Možnosti napájení* (může záležet na konkrétní verzi Windows). S notebookem můžeme dostat (a obvykle také dostaneme) přídavný software od výrobce notebooku (resp. základní desky nebo jiné komponenty), který také může sloužit k podrobnějšímu nastavení řízení spotřeby.

Komunikace mezi řízenými komponentami, BIOSem a operačním systémem probíhá ve speciálním jazyce *AML* (ACPI Machine Language), který je univerzální pro všechny typy komponent od různých výrobců. Pokud zařízení (komponenta) podporuje ACPI, musí rozumět tomuto jazyku.

Podobně jako APM, také ACPI rozeznává několik režimů činnosti.

- Working (S0) – plný výkon,
- Sleeping S1 (S1/POS) – obdoba Standby a Suspend to RAM z APM, pod proudem zůstává operační paměť a cache procesoru, ostatní komponenty jedou v šetřícím režimu,
- Sleeping S2 – podobně jako Sleeping S1, ale cache procesoru není napájena, obsah operační paměti zůstává zachován,
- Sleeping S3 (Save to RAM) – podobně jako Sleeping S2, ale všechny komponenty kromě operační paměti jsou zcela bez napájení, probouzení je o něco delší než v předchozích uspávacích režimech,
- Save to Disk (S4, Soft-Off) – veškeré komponenty jsou bez proudu, obsah operační paměti je uložen na disk, probouzení je nekolikrát delší než u S3, operační systém (zde reálně vypnutý, mimo RAM), který tento stav dokáže ošetřit, se po spuštění svého zavaděče dokáže obnovit z odkládacího prostoru na disku,
- Mechanical-Off (S5) – vypnutý systém.

Literatura

[1] *ABC Linuxu.cz*, sekce Hardware [online]. [cit. 2015-01-05]. Dostupné z: http://www.abclinuxu.cz/clanky/hardware

[2] Fog, Agner. Software optimization resources [online]. *Agner's pages* [cit. 2019-10-18]. Dostupné z: https://www.agner.org/optimize/

[3] *AMD: Product Resource Center* [online]. [cit. 2014-10-06]. Dostupné z: http://products.amd.com/

[4] *ARM Instruction Reference* [online]. *ARM Information center* [2019-01-30]. Dostupné z: http://infocenter.arm.com/help/topic/com.arm.doc.dui0068b/CIHEDHIF.html

[5] ARM Compiler v5.06 for mVision armasm User Guide. *Assembler User Guide* [online]. 2016 [2019-02-01]. Dostupné z: http://www.keil.com/support/man/docs/armasm/default.htm

[6] *CNews.cz* [online]. [cit. 2015-01-05]. Dostupné z: http://www.cnews.cz/

[7] Dembowski, Klaus. *Mistrovství v hardware.* Brno: Computer Press, 2009, 712 s. ISBN 978-80-251-2310-2.

[8] Hamacher, Carl, Zvonko Vranesic, Safwat Zaky a Naraig Manjikian. *Computer Organization and Embedded Systems.* 6th ed. New York, NY: McGraw-Hill, c2012. ISBN 978–0–07–338065–0. Dostupné z: http://www.staroceans.org/kernel-and-driver/Computeranization%20And%20Embedded%20Systems,%20Hamacher,%20Vranesic,%20Zaky,%20Manjikian,%206Ed,%20Mgh,%202012.pdf

[9] *Hardware Secrets: Uncomplicating the Complicated* [online]. [cit. 2015-01-05]. Dostupné z: http://www.hardwaresecrets.com/

[10] Horák, Jaroslav. *Hardware: učebnice pro pokročilé.* 4. aktualiz. vyd. Brno: Computer Press, 2007, 360 s. ISBN 978-80-251-1741-5.

[11] *HW.cz* [online]. [cit. 2015-01-05]. Dostupné z: http://www.hw.cz/

[12] *Intel: Architectures and Products* [online]. [cit. 2014-10-06]. Dostupné z: http://ark.intel.com/

[13] Intel 64 and IA-32 Architectures Software Developer's Manual: Volume 3 [online]. *Intel.com* [2019-03-19]. Dostupné z: https://www.intel.co.uk/content/www/uk/en/architecture-and-technology/64-ia-32-architectures-software-developer-system-programming-manual-325384.html

[14] *Intel Microarchitectures* [online][2019-04-02]. Dostupné z: https://en.wikichip.org/wiki/intel/microarchitectures

[15] *IXBTLabs: Computer Hardware in Detail* [online]. [cit. 2015-01-05]. Dostupné z: http://ixbtlabs.com/

[16] Jaitak, Er. Nagesh. *Computer Organization & Architecture*. Self publishing. ISBN 978-1-31-230609-7. Většina stránek dostupná z: https://books.google.cz/books?id=0ujYBgAAQBAJ&printsec=frontcover

[17] Lomont, Chris. Introduction to x64 Assembly [online]. *Intel Developer Zone*[2019-03-15]. Dostupné z: https://software.intel.com/en-us/articles/introduction-to-x64-assembly

[18] Lalonde, Jean-Francois. *ARM Instruction Set* [online]. [2019-02-01] Université Laval, Canada. Dostupné z: http://vision.gel.ulaval.ca/~jflalonde/cours/1001/h17/docs/arm-instructionset.pdf

[19] Mano, M. Morris. *Computer system architecture*. 3rd ed. Englewood Cliffs, N.J.: Prentice Hall, c1993. ISBN 978-013-1755-635. Dostupné z: https://faculty.psau.edu.sa/filedownload/doc-10-pdf-d171a71acbe44cd5cd2f78a40570a069-original.pdf

[20] *PC Tuning.cz* [online]. [cit. 2015-01-05]. Dostupné z: http://pctuning.tyden.cz/

[21] Stallings, William. *Computer Organization and Architecture: Designing for Performance*. 8th ed. Upper Saddle River, NJ: Prentice Hall, c2010. ISBN 978-0-13-607373-4. Dostupné z: https://inspirit.net.in/books/academic/Computer%20Organisation%20and%20Architecture%208e%20by%20William%20Stallings.pdf

[22] *Svět hardware* [online]. [cit. 2015-01-05]. Dostupné z: http://www.svethardware.cz/

[23] Tanenbaum, Andrew S. *Structured Computer Organization*. 5th ed. Upper Saddle River, N.J.: Pearson Prentice Hall, c2006. ISBN 01-314-8521-0. Dostupné z: https://eleccompengineering.files.wordpress.com/2014/07/structured_computer_organization_5th_edition_801_pages_2007_.pdf

[24] *Tom's Hardware: The Authority of Tech* [online]. [cit. 2015-01-05]. Dostupné z: http://www.tomshardware.com/

[25] *Think Computers: Computer Hardware Reviews* [online]. [cit. 2015-01-05]. Dostupné z: http://www.thinkcomputers.org/