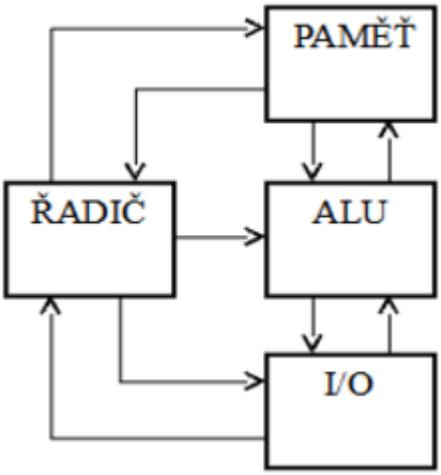
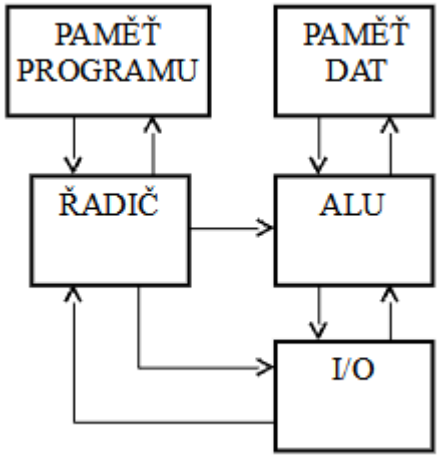
1. Koncepce a architektura číslicových počítačů

```
Pro koncepce číslicových počítačů se používají historicky dvě.
```

- Hardwardská koncepce
- Von Neumannova koncepce



Von Neumann Koncepce



Harvard Koncepce

Základní principy počítače Von Neumanova

A)

- ALU (Aritmeticko-logická jednotka)
- Řadič
- vstupní a výstupní jednotky
- paměť

B)

Algoritmus je převeden do programu (posloupnost instrukcí) => Technické prostředky jsou nezavíslé na řešené úloze.

C)

Všechny data, instrukce a také adresy jsou vyjádřeny dvojkovými číslicemi.

D)

Data a instrukce se uchovávají ve společné paměti na místech označených adresami. Přístup k datům v paměti trvá stejnou dobu pro různá paměťová místa.

E)

Zpracování dat řízené programem probíhá automaticky.

```
Druhá základní koncepce je Harvardská
```

- Ta se liší od von Neumanovy v bodu d: počítač má dvě oddělené paměti - jednu jen pro uložení programu, druhou jen pro data. (Paměťové místo není tedy jednoznačně určeno jen adresou - na stejné adrese jsou dvě, případně i více než dvě, různá paměťová místa.)

Základní pojmy

- **Instrukce se skládá z operačního znaku (= druh činnosti s operandy) a z adresy (= určení operandů).**
- **Operand je číslo, s nímž se vykonává příslušná operace**

Průběh výpočetního cyklu:

1. Načtení instrukce z operační paměti do řadiče. Adresa instrukce je v PC, instrukce se přesune do RI. Zvětšení obsahu PC (přičte se „1“), aby ukazoval na další instrukci.
2. Dekódování instrukce v DI. Zjištění, zda se budou načítat operandy z OP.
3. Natažení operandů z paměti do datových registrů v ALU, adresy jsou součástí instrukce nebo zakódovány, na paměťové místo ukazuje řadič (jeden z jeho registrů).
4. „Výpočet“ v ALU, získání výsledku a informací o něm (příznaky).
5. Uložení výsledku do paměti na místo, které je určené (zakódované) instrukcí, ukazuje na OP řadič.

2. Zobrazení údajů v číslicovém počítači, kódování

Proč používáme v hw ČP dvojkovou soustavu

V běžném životě používáme desítkovou soustavu, ale u hardwaru obecně nejde reprezentovat každé číslo stejně. Používat několik stupnic napětí a udržovat je tak aby byly čitelné a bezchybné by byl nemožný úkol.Místo toho se počítače zaměřují na používání pouze dvou hodnot a nuly a jedničky.Když se nachází vyšší napětí než např. 0.9 voltu jedná se o logickou jedničku. A samozřejmě se také tento systém nejlépe ukládá na magnetické a optické nosiče.

převod celého čísla DES do čísla BIN /pomocí dělení/

Příklad 71 desítkově. 71/2 = 35 jelikoz je se zbytkem tak 1 35/2 17 jelikoz je se zbytkem tak 1 17/2 8 jelikoz je se zbytkem tak 1
8/2 = 4 jelikoz je beze zbytku tak 0

4/2 = 2 jelikoz je beze zbytku tak 0
2/2 = 1 jelikoz je beze zbytku tak 0
1/2 = jelikoz je se zbytkem tak 1
a vezme to odspoda
71 desitkove je 01000111 dvojkove

převod necelého - reálného čísla BIN do čísla DES /pomocí mnohočlenu/

číslo kupříkladu 00010110 dvojkově převedu pomocí

nula krát dva na nultou+ jedna krát dva na prvou + jedna krát dva na druhou + nula krát dva na třetí + jedna krát dva na čtvrtou + nula krát dva na pátou nula krát dva na šestou + nula krát dva na sedmou což se rovná 21 desítkově

převod celého čísla DES do čísla BIN /pomocí řádové mřížky/ - příklad u IP adresy 138.72...

například čislo 183 desítkově

128 64 32 16 8 4 2 1
1 0 1 1 0 1 1 1

10110111 dvojkově

Vyberu si nejblíže mocninu dvou na n a jednoduše jestli je to číslo menší jak čislo nad ním tak ho odečtu napíšu jedničku a jdu na další.

kódování čísel DES do n-bitového BINÁRNÍHO kódu /n = 4, 5, 16, ... co to je váhový kód a jeho význam, hodnoty rozsahu čísel DES v závislosti na n= xx/

BCD kód je to tzv. "váhový" kod a reprezentuje stavy 0-9 jiné hodnoty nereprezentuje.Při převodu např. 2390 desítkově musíme převést každé čislo samostatně tedy 0010 0011 1001 0000 v bcd kodu. 2 desitkove tedy= 8 4 2 1
0 0 1 0

převod čísla BIN do čísla HEX /vysvětlit vznik Hexadecimálních čísel a jejich význam při zkráceném zápisu operandu nebo adresy, BIN např. jen 10 místné pro převod/

00011110 dvojkove tak to je 1E seštnáctkově převod je jednoduchy HEX využivá {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F} a kodovani je pres ctyr mistnou mrizku tedy 8 4 2 1 Jestliže máme jen desetimístné čislo přídáme před něj další dvě nuly.HEX zápis čísel se využívá zhlediska zkráceného zápisu jinak velkých čísel.

převod čísla HEX do čísla BIN

2F šestnáckově vezmu každé číslo a rozeberu ho na čtyři bity
2 = 8 4 2 1

0 0 1 0
F = 8 4 2 1
1 1 1 1
2F HEX = 0010 1111 BIN

kódování čísel DES do kódu BCD /vysvětlit vznik kódu BCD, co to je váhový kód a jeho význam/ BCD kód je to tzv. "váhový" kod a reprezentuje stavy 0-9 jiné hodnoty nereprezentuje.Při převodu např. 2390 desítkově musíme převést každé čislo samostatně tedy 0010 0011 1001 0000 v bcd kodu. 2 desitkove tedy= 8 4 2 1
0 0 1 0

kódování čísel do kódu GRAY /vysvětlit vznik kódu GRAY, co to je neváhový kód, význam u PLC/

kod kde se po sobe jdouci hodnoty liší v bitovém vyjádření změnou pouze jedné bitové pozice.původně navržen kvůli řušení z elektromagnetických přepínačů . Navržen tak aby eliminoval jednoznačnost.

kódování znaků v ČP (písmena, číslice, řídící znaky komunikace, …)

Používá se kodování ASCII což je kódová tabulka, která definuje znaky a převádí na reprezentaci ve dvojkové soustavě.

pravidla kódu ASCII vč. jeho rozsahu, zavedení národního prostředí /srovnání En – CZ/ a spec. znaky semigrafiky

První ASCII tabulka měla rozsah 128 znaků a byla americká a přídavek o dalších 128 znaků přidal české znaky.

- pravidla pro kódy UNICODE, WINxxx, UTF /každý jednou větou, zkouší se v PV ústní, DS ústní/

S každou verzí unicodu přibývali nové a nové znaky první verze z roku 1991 měla 7129 znaků. Každá z těchto kodovacích sad je omezena i když už je docela vysoká např UTF32 využívá 4 bajty. Unicode je reprezentace, enkodovani a práce s textem Základní kódování Unicode jsou: UTF-8 UTF-16 (UTF-16BE, UTF-16LE) UTF-32 (UTF-32BE, UTF-32LE) WINxxx je take textova reprezentace vyrobena microsoftem

3. Aritmetické a logické operace v číslicovém počítači, logické funkce

- uveďte -množiny čísel DES / BIN „přirozená“, „celá“, „racionální“, „iracionální“, formáty čísel „INTEGER“, „REAL“, „DOUBLE“ a kde se zpracovávají (celočíselná ALU, NEU z koprocesoru) přirozená by se dala reprezentovat unsigned int,pro celá integer a proc racionální a iracionální používame float a double. Double v C++ využívá dvojnásobného místa 64 bitů oproti floatu, ale má více místa na floating point tedy místa za čárkou. Tedy celočíselná

čísla se zpracovávají aritmeticko logické jednotce a s desetinnou čárkou v floating point unit tedy v matematickém koprocesoru

-přehled aritmetických operací (+ - * /) v EU s celočíselnou ALU – sčítačka, násobička Binárně sčítat znamená pokud je tam jedna jednička je to jedna jestli obě čísla tak o řád nahoru Binárně odčítat znamená upravi menšitele na šířku menšence udělat doplněk menšitele což znamená negaci a k doplnku přičíst jedničku součet menšence a druhého doplnku menšitele a upravíme rozdíl na stejnou šířku Binární násobička 1101

- 1010

  0000 1101 0000 1101 10000010 Pokud to neni 1*1 tak nula a kazdy bit nasobim zvlast Binárně dělit 11011101/1010 = 1011 1111 1010 01 - zbytek -aritmetické sčítání BIN (A + B), vysvětlení přenosu mezi řády, doporučeno A B celá čís. bez znaménka max. 6 bitů poloviční binární sčítačka A B C S 0 0 0 0 0 1 0 1 1 0 0 1 1 1 1 0 S je suma prvního řádu a C je carry out což je o jeden řád nahoru

-aritmetické odčítání BIN (A - B) za pomoci sčítání s použitím druhého doplňku /pravidla pro jeho vytvoření/, doporučeno A B celá čísla max. 6 bitů a A>B

Binárně odčítat znamená upravi menšitele na šířku menšence udělat doplněk menšitele což znamená negaci a k doplnku přičíst jedničku součet menšence a druhého doplnku menšitele a upravíme rozdíl na stejnou šířku

1101111-1001000 = 0100111

1001000 znegovat 0110111 0110111 + 1 = 0111000

1101111 +0111000

10100111 a upravime na sirku mensence tedy ubereme prvni cislo 0100111

-zapsat doplněk do Bytu (např. -50 DES), znaménkový bit, rozsahy pro Byte (-,+) Znamenkova cisla 8bitu + je logicka nula a - je logicka jednicka jako prvni bit v bytu tedy hodnoty takoveho cisla jsou +127 az -127 +127 je 01111111 -18 je 10010010

-aritmetické sčítání BCD (A + B), jen slovně jak se dělá korekce výsledku, kdy je BCD+ výhodné

Korekce vysledku se provadi dalsi korekcni scitackou jenz po bcd detektoru a scitacke opravi vysledek na bcd za pouziti korekcniho cisla je jim bud 0 nebo 6 a ma pak i paty bit C

- výroková matematika (technické využití některých log.funkcí / operací), popsat operace „průnik“, „sjednocení“, „shodnost“, „implikace“, „negace“ s využitím Vennových diagramů

prunik logicky AND A B AND 0 0 0 0 1 0 1 0 0 1 1 1 jen pokud jsou obe v jednicce nebo prunik dvou mnozin je mnozina jen spolecnych prvku

sjednoceni logicky OR A B OR 0 0 0 0 1 1 1 0 1 1 1 1 kdyz alespon jednou jednicka tak vysledek je jedna.Sjednoceni dvou mnozin je mnozina vsechn techto prvku Implikace A B A->B 0 0 1 0 1 1 1 0 0 1 1 1

znamená vztah vyplývání nebo zahrnutí. Skutečnost nebo výpověď A implikuje nějaké B
Neshodnost logicky XOR

A B XOR 0 0 0 0 1 1 1 0 1 1 1 0

Jestli ze se nerovnaji tak jednicka

-logické násobení BIN (A * B), pravdivostní tabulka pro *, doporučeno A B max. 6 bitů, maskování

A B A*B 0 0 0 0 1 0 1 0 0 1 1 1

```
1101
```

- 1010

  0000 1101 0000 1101 10000010

- logické sčítání BIN (A + B), pravdivostní tabulka pro +, doporučeno A B max. 6 bitů

A B A+B 0 0 0 0 1 1 1 0 1 1 1 1 1111 +0100 010011

- logický doplněk BIN, pravdivostní tabulka pro negaci, doporučeno A max. 6 bitů A A' 0 1 1 0

100101 jeho negace 011010

-logická neshodnost BIN (A xor B), pravdivostní tabulka pro xor, doporučeno A B max. 6 bitů A B XOR 0 0 0 0 1 1 1 0 1 1 1 0

11001 XOR 00101 = 11100 kazdy bit pokud se neshoduji tak jedan

4. Kombinační obvody, jejich realizace z pravdivostní tabulky

Pravdivostní tabulky logických funkcí

- Logický průnik AND

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

- Logické sjednocení OR

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

- Logická neshodnost XOR

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

- Logický doplněk Negace

A	A'
0	1
1	0

Typy kombinačních obvodů

Nejdostupnějším kombinačním obvodem je NAND, neboli známé hradlo 74HC00.Existuje dvou vstupové až osmivstupové ale nikoli liché číslo. Jeho pravdivostní tabulka:

5. Sekvenční obvody, jejich realizace, použité klopné obvody

6. Programovatelné logické obvody

7. Mikrořadiče (MCU), jeho struktura, význačné integrované periferie

8. Program, programovací jazyky, příkaz, instrukce, druhy adresování, skoky

9. Mikroprocesor v reálném režimu, adresování LA a FA