

PAMĚTI

1. V hlavní paměti von Neumann typu se nachází instrukce a data.
 Paměťové místo - závisí na architektuře, v podstatě nejmenší jednotka, kterou je možná adresovat celočíselnou adresou.
 SLABKA 8bitů; Slovo - různé, nejčastěji 16 bitů (několik slabik).

2. Typy dat - instrukce a data

Numerická data: typ float

little endian - nižší slabiky na nižších adresách
 big endian - vyšší slabiky na nižších adresách

Ne numerická data: například ASCII kód, logické hodnoty, pixely

4. 0x000123DF na adresu 5000:

	5000	5002	5004	5006
little endian	0F	23	01	00
big endian	00	01	23	0F

5. RAM - random access memory, paměť s libovolným přístupem čtení/zápis

SRAM - Static RAM, je založena na stavtech klopových obvodů, protože má menší proudy oproti dynamické, používá se v menších kapacitách, je rychlejší než DRAM, používá se

DRAM - Dynamic RAM, paměťový bit tvoří tranzistor a kapacita protože se postupně ztrácí náboj v C, musí se obnovovat, čtení je destruktivní.

ROM - read only memory

PROM - jednou programovatelný obvod

EPROM - masově ROM např. osvětlením UV zářením obvodu

EEPROM - elektricky mazatelná EPROM

} energeticky nezávislé

6. Štěrnicí styku s hlavní pamětí:

ADRESOVA' - přenáší adresu

DATOVA' - přenáší data

RIDIČI' - signály: Write Enable, Output Enable, Chip Select

STAVOVA' - např. signal WAIT (z paměti do procesoru)

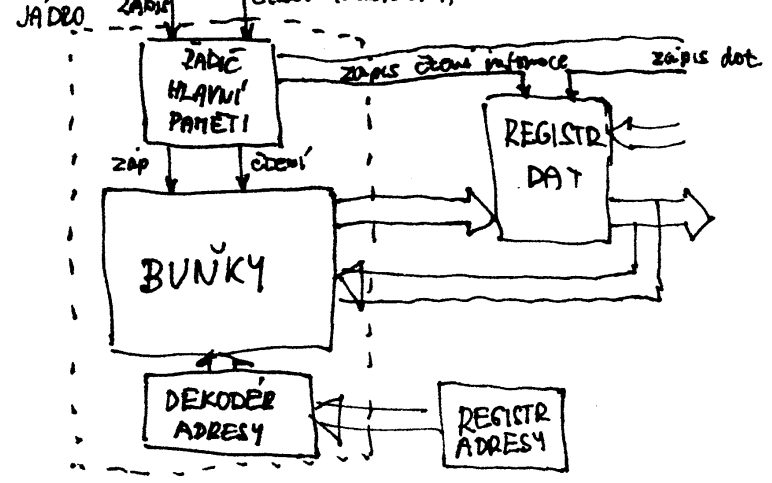
8. Hierarchie paměťového systému

několika úrovně asociativní paměti různých kapacit a rychlosti s útlum dosáhnout vyhovujícího poměru výkonu a kapacity

Typy: registry, cache, hlavní paměť, mější, záložní (DRAM) HDD FLOPPY

Kriteria: doba přístupu, kapacita, cena, ...

9. Bloky paměťového obvodu



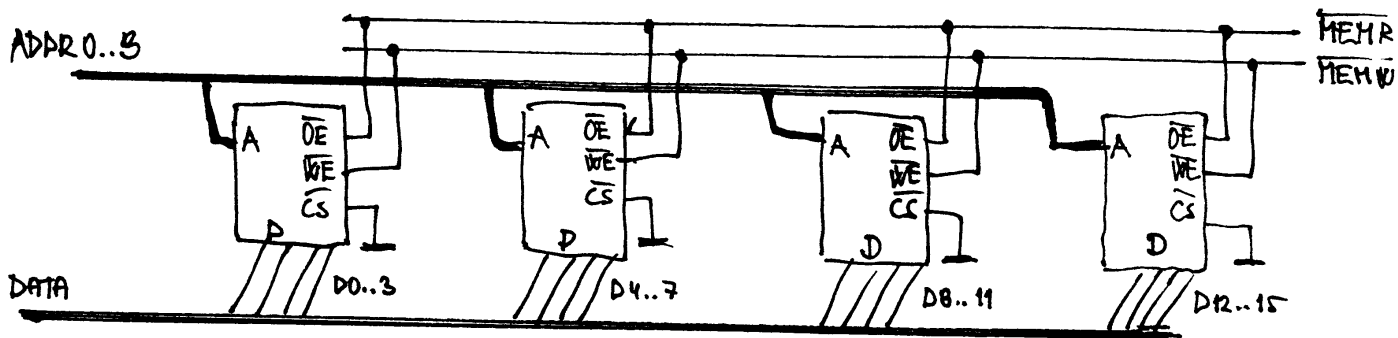
Vstup:

- řez adresy - adresa
- registr dot. data (zápis dot.)
- zápis

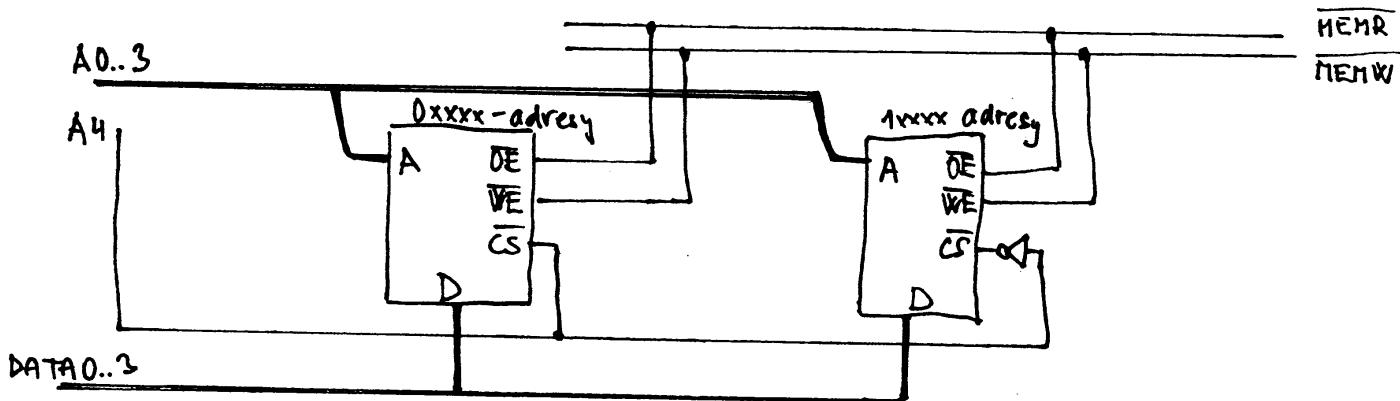
Výstup:

- nastavení adresy
- čtení
- vyčtení dot. z registru dot.

12. Paměťový modul s kapacitou 16 x 4 bity
 - vytvořit paměť 16 x 16 bitů ~ ROZŠÍŘENÍ SLOVA HLAVNÍ PAMĚTI



13. První modul 16 x 4 bity vytvořit paměť s kapacitou 32 x 4 bity
 ~ ROZŠÍŘENÍ ADRESOVÉHO PROSTORU



PAMĚTI CACHE & VIRTUÁLNÍ PAMĚTI

- VIRTUÁLNÍ PAMĚŤ** - vytváření adresových prostorů z různých typů pamětí.
 Např. RAM + HDD (souprození). Kapacita může být větší než je skutečná fyzická.
 virtuální paměť pracuje s logickými adresami.
- FYZICKÁ ADRESA** - hodnota signálu na adresní sběrnici hlavní paměti
LOGICKÁ - segment + offset
VIRTUÁLNÍ - typ logické adresy která je převedena posléze na fyzické adresy
- Rozdíl mezi **STRÁNKOU** A **SEGMENTEM**
SEGMENT - souvislá část paměti RAM s pevnou velikostí, mohou se překrývat (v hlavní paměti)
STRÁNKA - mohou se překrývat, úseky $S = 2^q$, kde q je počet bitů adresy, stránky začínají u adresy dělitelné S . (musí být ve větší paměti)
- Umístění fyzického a logického adresového prostoru
FYZICKÝ - je rozdělen na stejné velké úseky pevné délky nazývané **STRÁNKOVÉ RAMCE** (hlavní paměť)
LOGICKÝ - je rozdělen na úseky pevné délky nazývané **STRÁNKY**, je realizován pomocí větší paměti a data jsou přesouvána do hlavní paměti po stránkách, jsou-li požadována programem a pokud v hl. paměti ještě tato stránka není.
- Dvouúrovňová organizace tabulky stránek**
 Tabulka obsahuje informaci, zda se kopie příslušné stránky nachází v hlavní paměti.
 Log. adresa - [číslo stránky : adresa ve stránce]. Tabulka je docela rozsáhlá, takže se řeší dvou úrovně: V první úrovni je pouze část této tabulky a ve druhé úrovni se rozkládá zbytek adresy.

6. Co je to CACHE?

Rychlá paměť mezi hlavní pamětí a procesorem. Do cache se proběhne ukojení data z hlavní paměti a při přístupu se zjistí zda už nejsou v cache, pokud by se našla, rychleji. Pokud v cachej přístupu data nejsou musí se načíst standardním způsobem. Realizovaná ASOCIATIVNÍ PAMĚTI. Obsahuje nejčastěji používané položky paměti z hlavní paměti.

7. Co je klíč a v jakém doje v rychlé vyrovnávací paměti?

ASOCIATIVNÍ PAMĚT - není adresována adresou, ale částí datové položky, která se má vyhledat. Do paměti se zapisuje - DATA, KLÍČ, BIT PLATNOSTI

KLÍČ - obdoba adresy, její v paměti položka se zakonjen klíčem s bitem platnosti. Přičte se, jinak se kópi, a položka v paměti není.

8. Příklad virtuální adresy na fyzickou u stránkové a segmentové paměti.

Stránka musí začínat na násobek 2^q kde q je počet bitů adresy.

9. ASOCIATIVNÍ PAMĚT - není adresována adresou ale částí datové položky ... viz 7.

Jak se pozná že je položka v paměti? Přítomnost se pozná porovnáním klíče s klíči uloženými v adresář (případně několika klíči podle stupně asociativity)

10. Asociativní paměť vs. paměť s omezeným stupněm asociativity

Plně asociativní paměť - nelze použít standardní obvody RAM, větší rozměr, při stejné kapacitě oproti sta. RAM.

OMEZENÝ STUPĚŇ ASOCIATIVITY: Každá položka z hlavní paměti je podle adresy přidělena jedno první místo (případně víc, podle stupně), kde se v cache může nacházet. Adresář cache lze pak realizovat běžnou pamětí RAM, přítomnost položky se zjistí porovnáním klíče.

11. Obsah tabulky stránek u virtuální paměti

Pro každou stránku obsahuje informaci zda se kopie příslušné stránky nachází v hlavní paměti a pokud ano tak ve kterém stránkovém rámcu, je uložena v hlavní paměti. Začátek je v registru začátku tabulky stránek. Příznak indikující známou obsah.

12. Čtení z hlavní paměti s pamětí CACHE:

Současně se čte z cache a z hlavní paměti, pokud jsou data v cache tak se cyklus čtení z hl. paměti nedokončí. Jinak se čtou z hlavní paměti a pak až z cache.

ZAPIS: pokud položka není v cache zapíše se zpravidla jen do hlavní paměti

Write through - PRŮBĚŽNÝ ZAPIS - zápis současně do cache i hlavní paměti

Write back - ODLOŽENÝ ZAPIS - nová hodnota se zapisuje pouze do cache

13. Příklad virtuální adresy na fyzickou u stránkové paměti.

Virtuální adresa ...

ČÍSLO STRÁNKY	ADRESA VE STRÁNCE
---------------	-------------------

→ register + číslo

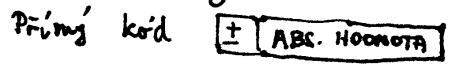
Podle čísla stránek se v registru začátku tabulek stránek vybere číslo stránkového rámece. Uvnitř konkrétního rámece se adresuje adresa ve stránce.

Co se stane, není-li domá virtuální adresa v paměti?

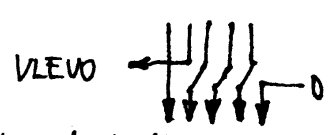
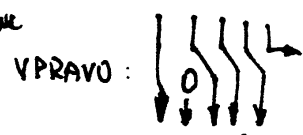
Domá-li přítomna vyvola se přeručení, jehož obsluha vyvola načtení stránky z vnější paměti.

Posuvy a ROZŠIŘOVÁNÍ ŘÁDOVÉ MŘÍŽKY

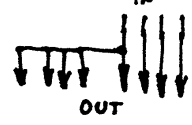
1. Aritmetický posuv v přímém kódu



0a+
1a-



2. Obvod 4 vstupy, 8 výstupu, vstup číslo IN v doplňkovém kódu což výstup bude též číslo (8 bitové)



3. $Y = 5001$ je obrazem $D(x)$ čísla X v doplňkovém kódu (10tkové). Modul mřížky 1000.

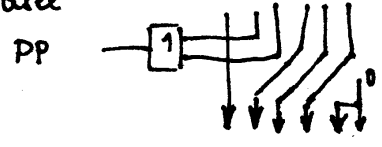
• Určit $X = Y - \text{MODUL} = -4999$ nebo $[\text{negace}(Y) + 1]$

• Aritmetický posuv $D(x)$ o 1 místo vpravo: Na první místo jednotka řádové mřížky $D(x) = 9500 \sim -500$

• Je výsledek $X/10$? Proč? $-500 \neq -499,9$ ztráta přesnosti... vyznalo se jiné číslo než 0.

4. Obvod pro posuv 6 bit čísla v přímém kódu o 2 místa vlevo. Detekovat přeplnění.

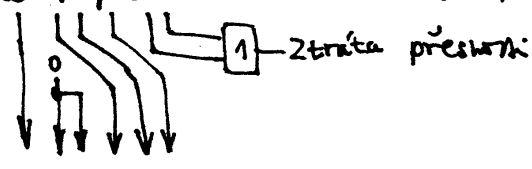
Určit rozsah čísel



Rozsah zobrazitelných:

-32..32

5. Ar. posuv 6 bit. čísla v přímém kódu o 2 místa vpravo

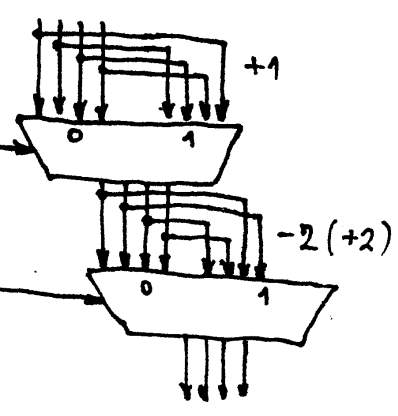


Rozsah zobrazitelných:

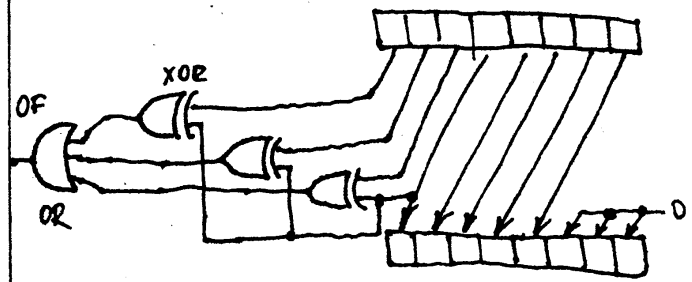
-32..32

6. Cyklický posuv 4 bit čísla o N míst. N je určeno dvěma řídicími vstupy Q a platí $-2 \leq n \leq 1$; $Q = n+2$. Multiplexory 2×4 dot. vstupy a 4 výstupy.

Q_1	Q_0	n	$Q+2$
0	0	0	-2
0	1	1	-1
1	0	-2	0
1	1	-1	1



8. Arit. posuv 8 bit čísla v doplňk. kódu o 3 místa vlevo. Detekce přeplnění



7. Arit. posuv 4bit VPRAVO Oaž 3 místa. MPX 2×4 vstupy (doplňkový kód)

Q_1	Q_0	Posun
0	0	0
0	1	1
1	0	2
1	1	3

Schéma je shodné jako u 6.

ŠČÍTAČKY A ODCÍTAČKY A DESÍTKOVÉ KÓDY

1. Jak se pozná přepnutí při sčítání nezáporných čísel (čísel bez znaménka)?
 k PŘEPNUTÍ dochází pokud se sečtou dvě čísla se stejným znaménkem a výsledkem je číslo s opačným znaménkem. U nezáporných čísel, pokud je přenos z nejvyššího řádu rovný 1.

2. Sčítačka s výhybkami. (paralelní sčít., menší spoždění než bez výhybek)

Zavodíme $G_i = a_i \cdot b_i$, $P_i = a_i \oplus b_i$ - přenos se generuje, přenosy produkuje

Platí-li $P_i \cdot P_{i+1} \cdot \dots \cdot P_j = 1$ - přenos produkuje všechny řády

Sčítačky se rozdělí do sekcí a přenosy produkuje pro výhybkách, zpoždění se zmenší na spoždění výhybek v rámci dvou sekcí v nichž přenos vzniká a zaniká.

Viz strana 57.

3. Princip asynchronní sčítačky

- koncepce vychází z toho, že ve většině případů neprodukuje přenos ani zdaleka všemi řády
- interval mezi hodinovými pulzy je přizpůsoben nejhoršímu případu (přechod přes všechny řády) u normální sčítačky a proto se u asynchronní:
- sčítačka detekuje akcím předchozího děje (na výstupu správné hodnoty všech přenosů), příslušný signál je použit pro řízení generátoru hodin pulsu

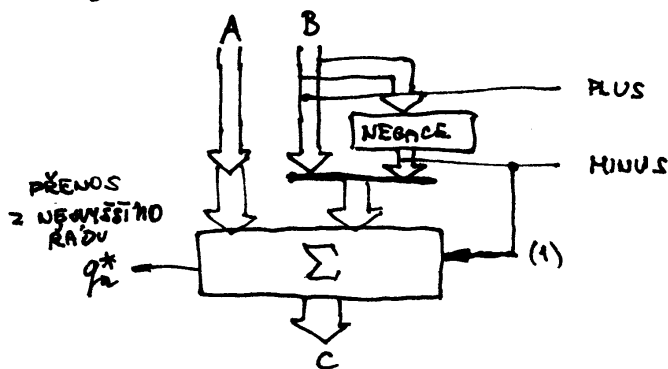
4. Princip sčítačky s podmíněnými součty

- místo úplných sčítaček se v ní používají obvody, které vytváří podmíněné součty a podmíněné přenosy

5. Doplnit paralelní sčítačku pro provádění odčítání nezáporných čísel?

Budou se přičítat záporná čísla vyjádřená v doplňkovém kódu.

Odčítaný operand se zneguje a do nejvyššího řádu se přivede, kotka jednička.



Jak se pozná záporný výsledek?

ŠČÍTANÍ $q^* = 1$

ODČÍTANÍ $q^* = 0$

6. Jako 5. pro desítkovou sčítačku

Desítková negace $\bar{X} = 9 - X$ jinak je to stejné jako 5.

7. Princip sčítačky s predikcí přenosu CARRY LOOK-AHEAD

Přenosy se vytváří ve zjednodšeném obvodu místo úplné sčítačky a přivádí se přímo na příslušné vstupy úplných sčítaček.

Pomocné proměnné - PROPAGATE - přenos produkuje, GENERATE - přenos se generuje.

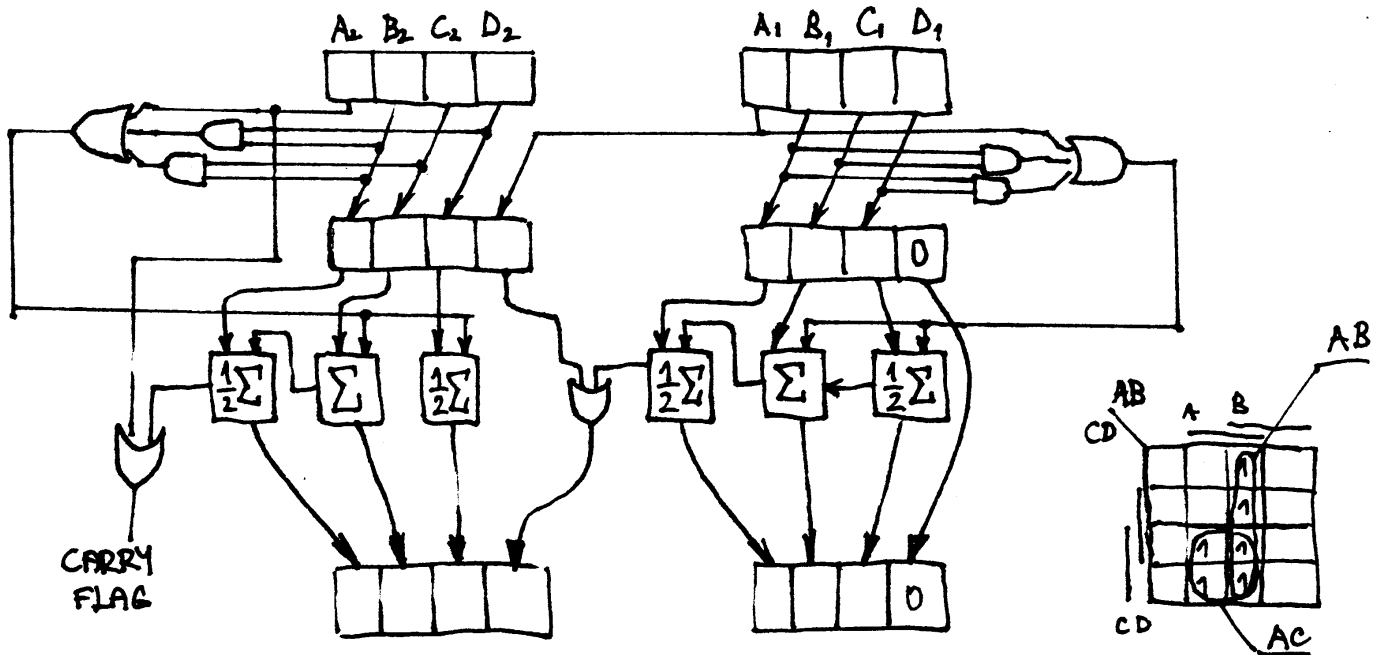
8. Obvod na jehoz 8 vstupech je dvoumístné BCD číslo, na výstupu by mělo být číslo $2X$. Doplnit indikací přepnutí.

1. Binární posun o 1 místo doleva $\sim 2X + CF$

2. Proveš korekci po sčítání

a) pokud je první nibble $2X$ větší než 9 přičíst 6

b) Pokud je druhý nibble $2X$ větší než 9 přičíst 60h a $CF=1$



$$Y = AB + AC + \text{PŘENOS V NIBBLE}$$

ZÁPORNÁ ČÍSLA

1. Které kódy jsou nejednoznačné?

PŘÍMÝ: kladná nula 0000, záporná nula 1000

INVERZNÍ: 0000 1111

2. Rozsah zobrazitelných čísel, základ soustavy 2, mřížka 8 bit, jednotka 1

Přímý: $-127 \dots +127$ $M = (10000000)_2 = 256$ $|X| < \frac{1}{2}M = 128$

Doplňkový: $-128 \dots +127$ $-\frac{1}{2}M \leq X < \frac{1}{2}M$

Inverzní: $-127 \dots +127$ $|X| < \frac{1}{2}M$

Aditivní: $-K \dots 255-K$ (K -posunutí) $-K \leq X < M-K$

3. Rozsah zobrazitelných, $M = 10000_{10}$, jednotka 1

Doplňkový: $-5000 \dots +4999$

Inverzní: $-4999 \dots +4999$

Aditivní: $-K \dots 9999-K$

4. Vztah mezi obrazem čísla v doplňkovém a aditivním kódu.

$$D(x) = \begin{cases} X & \dots X \geq 0 \\ M+X & \dots X < 0 \end{cases} \quad \text{ADITIVNÍ} \quad \begin{cases} X \geq 0: A(x) = D(x) + \frac{1}{2}M \\ X < 0: A(x) = D(x) - \frac{1}{2}M \end{cases}$$

$$A(x) = X + K = X + \frac{1}{2}M$$

5. Jak se mění obraz opačného čísla a abs. hodnoty? Binární soustava. E-jednotka

Přímý OPAČNĚ inverze nejvyššího bitu

Doplňkový $D(-X) = \overline{D(X)} + E = M - D(X)$

Inverzní $D(-X) = \overline{D(X)}$

ABSOLUTNÍ HODNOTA
nulování nejvyššího bitu

$$D(|X|) = \begin{cases} D(X) & \text{pro } X \geq 0 \\ \overline{D(X)} & \text{pro } X < 0 \end{cases}$$

$$D(|X|) = \begin{cases} D(X) & \text{pro } X \geq 0 \\ \overline{D(-X)} & \text{pro } X < 0 \end{cases}$$

6. viz 4.
 7. Jak se posuná přepnutí při sčítání?

Průměr: přenos z nejvyššího řádu je 1

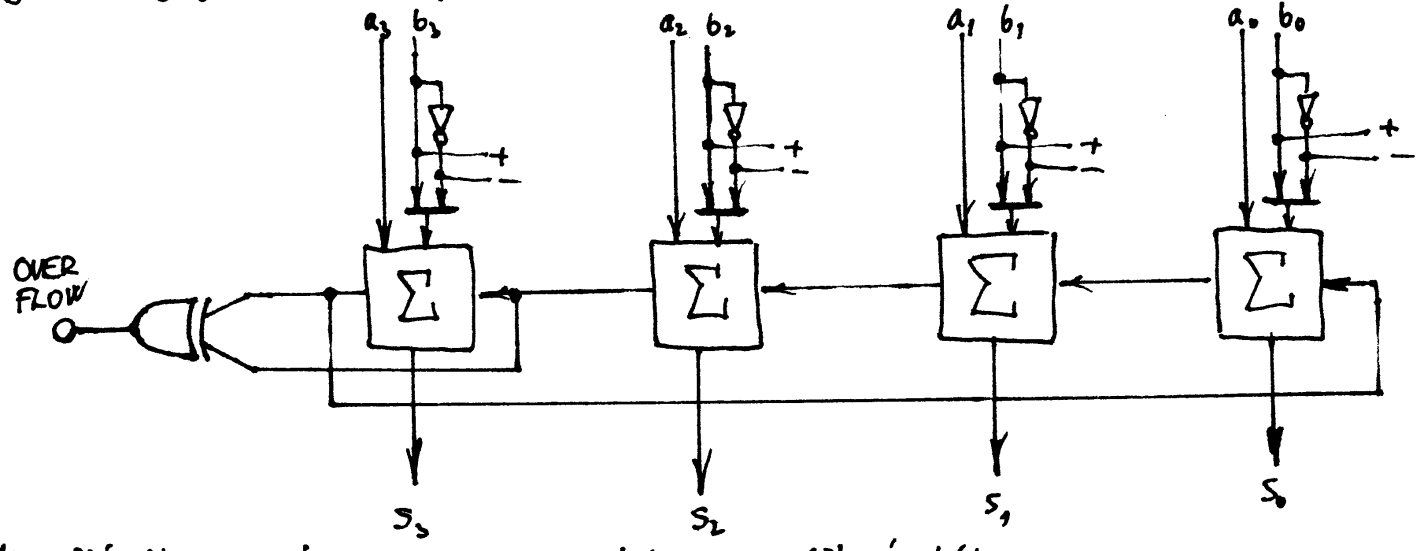
Doplňkový: $\oplus + \oplus \rightarrow \ominus$, $\ominus + \ominus \rightarrow \oplus$, a binární přepnutí pokud přenos z nejvyššího řádu se rovná přenos do něj.

Inverzní: stejné jako u doplňkového

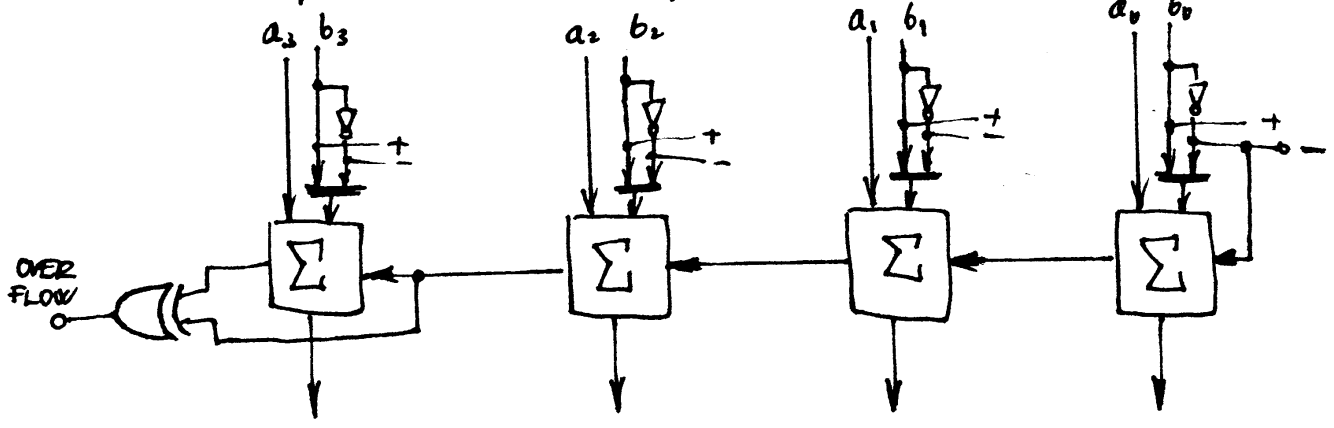
8. Přepnutí v doplňkovém a inverzním kódu v 10tkové soustavě?

Stejně u obou $\oplus + \oplus \rightarrow \ominus$, $\ominus + \ominus \rightarrow \oplus$

9. Sčítačka - odčítáčka pro 4bitová čísla v inverzním kódu s detekcí přepnutí.



10. Sčítačka - odčítáčka pro 4bit. čísla v doplňkovém kódu



POHYBLIVÁ ŘÁDOVÁ ČÁRKA

1. Vztah mezi číslem a jeho obrazem v pohyblivé řádové čárce

A.. číslo, m.. mantisa, e.. exponent, z.. základ soustavy
 $A = m \cdot z^e$

2. Mantisa - součet popisů čísla s pomocí desítkovou čárkou. Např. číslo 121000 se v této notaci vyjádří jako 1,21E+05, kde 1,21 je mantisa a E+05 je exponent (charakteristika).

3. Sčítání v pohyblivé řádové čárce?

Upravit operandy, aby měly stejný exponent a potom se mantisy sečtou.

5. Násobení/dělení

Vynásobit/dělit mantisy a sečíst exponenty

7. Kdy dojde k přeplnění a nenaplnění v polybitové řádové čárce.

Když výsledný exponent je větší než největší možný - PŘEPLNĚNÍ.
menší než nejmenší - NENAPLNĚNÍ

Přeplnění - je detekováno tehdy, je-li výsledek větší než maximální zobrazené číslo v použitém formátu dat.

NENAPLNĚNÍ - nastane při nulovém výsledku operace, který je tak malý, že jej nelze v daném formátu dat zobrazit.

8. Normalizovaný tvar čísel je takový tvar, že není nutné posunout mantisa dolů \Rightarrow jednodušší algoritmy aritmetických operací.

Zabezpečení práce s normalizovanými čísly - po provedení operace může být výsledek v nenormalizovaném tvaru, po operaci je nutné číslo převést na normalizovaný tvar.

9. Co je to princip skryté jedničky?

Nulové, normalizovaná mantisa v přímém kódu má na nejvyšším významovém bitu vždy jedničku, která se v zápisu vynechává. Exponent by měl být v aditivním kódu

Jak se zobrazí nula? -

Nezáleží na znaménku a zbytek jsou samé nuly (mantisa, exponent).

Je zavedena konvence, podle níž se princip skryté jedničky nepoužije, je-li obraz exponenta roven nule (pak $e = -126$ a 8 bit)

10. 16 bit. číslo Y je obrazem čísla X v polybitové čárce. $Y = F D A 0$

$$Y = \underline{1111} \underline{0000} \underline{1010} \underline{0000}$$

\rightarrow Mantisa v dekadickém kódu, mířka 2, $-1 < M < 1$

\rightarrow exponent $A(E) = E + 8$

Y není normalizovaný tvar

$$X \text{ číslo} = \left(\frac{1}{2^4} + \frac{1}{2^6} \right) \cdot 2^7 = 10,00000$$

$$\text{Normaliz } Y = 1100 \ 0,101 \ 0000 \ 0000$$

$$Y = C \ 5 \ 0 \ 0$$

NAŠOBNÍ A DĚLENÍ

1. Rozdíl mezi dělením s návratem přes nulu a bez návratu bez přes nulu. Při dělení bez návratu se získá každou další operací (v jednom taktu) jeden bit podílu. V dělení s návratem přes nulu jsou vždy třeba dvě další operace - je-li bit podílu roven nule.

2. Kolik 8bitových registrů je třeba pro násobení dvou 8bit čísel a co budou obsahovat.

Registry: A, B, C0, C1

Na začátku: A, B činitele, C1 nula

Během operace: A: nemění se

B: v každém taktu posuv o 1 místo vpravo

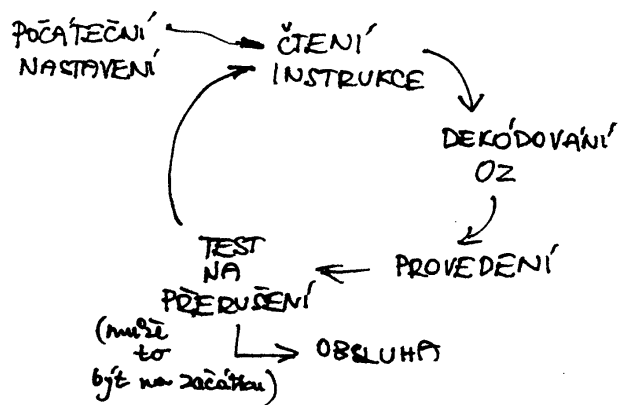
C1: V prvním taktu nula, v dalších taktách o jedno místo vpravo posunutý výsledek předchozího sečítání (výpadek definitivně učený typ)

C0: Ukládají se do něj definitivně učené byty součinnu v každém taktu se posuvá o 1 místo vpravo.

Po skončení: C1 - vyšších 8 bitů součinnu, C0 - nižších 8 bitů, A - nemění se, B nuly

NOVÉ OTÁZKY

1. ZÁKLADNÍ CYKLUS ŘADIČE POČÍTAČE



2. MIKROPROGRAMOVÝ ŘADIČ

Provedení 1 operace sestává z provedení dílčích operací - můžeme je zadat jako mikroinstrukce.

TYPY ŘADIČŮ

- vertikální: krátke mikroinstrukce obdobné instrukcím, 1 mikroinstrukce má několik taktů, používá se u 166 MHz mikroprogr. čítače
- horizontální: dlouhá mikroinstr. 64 bitů a více, 1 instrukce = 1 takt, omezená volba adresy následující instrukce, řídicí signály se určují v operačním znaku.
- diagonální: dlouhý mikrooperační znak a použití mikroprogr. čítače.

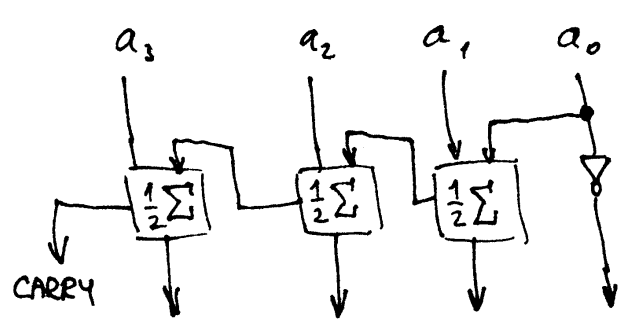
OPSAH HORIZONTÁLNÍ MIKROINSTRUKCE :

POZ	ADRESA NÁSLEDUJÍCÍ INSTRUKCE	VÝBĚR PODMÍNKY
-----	------------------------------	----------------

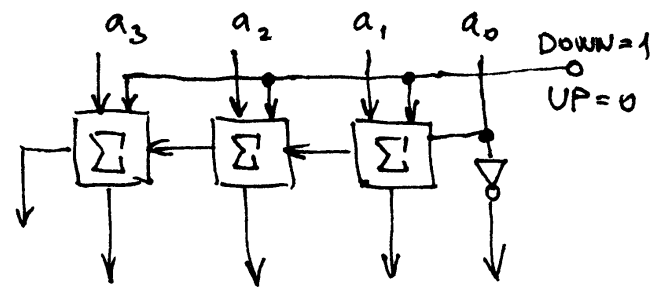
3. ZÁKLADNÍ REGISTRY ŘADIČE POČÍTAČE

Programový čítač, registr instrukcí, dekodér operačního znaku, ukazatel zásobníku

4. OBVOD PRO INKREMENTACI



UP/DOWN



5. IEEE Formáty čísel

