

Y36SAP - 12

Příklad návrhu výkonné a řídicí
jednotky
Kódování instrukcí ADOP

27.5.2009

Y36SAP-processor

1

Zkoušky 2009

Ut. 9. 6. 8:30 K1

St 10. 6. 11:00 K1

Ut. 16. 6. 9.00 K1

Ut. 23. 6. 8:30 K1

Ct. 2. 7. 8:30 K1

POZOR! K1 = KN E107

Náhradní testy

Po. 8. 6. 16:00 K1

Přihlašování v KOSu na „Jednorázovou akci“

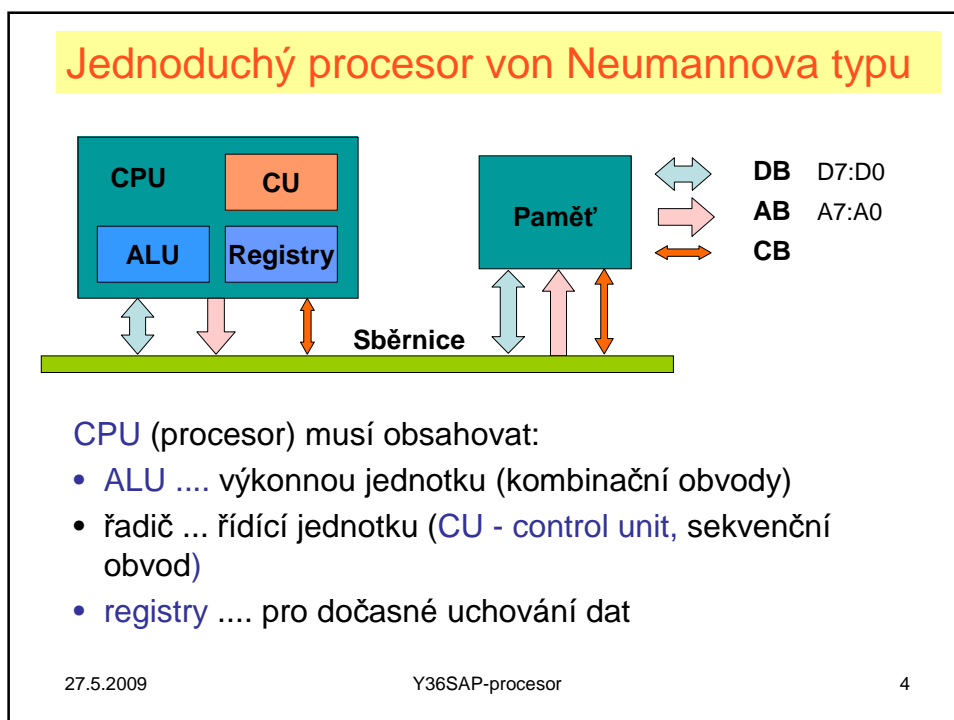
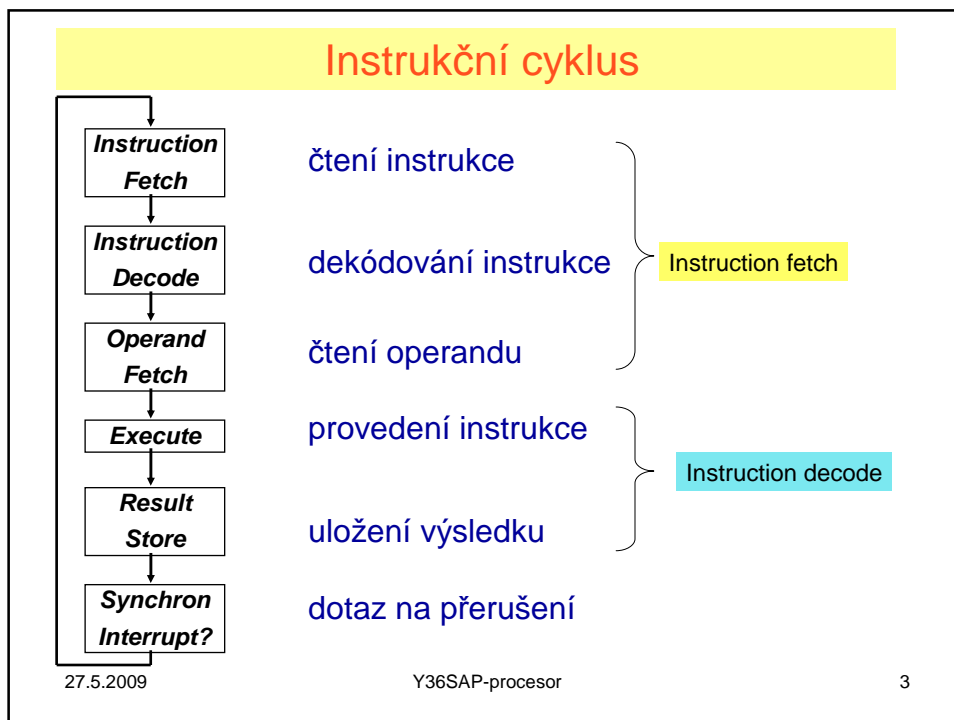
T1-Y36SAP a T2-Y36SAP

(po dohodě s cvičicím možno opravit 1 z testů)

27.5.2009

Y36SAP-processor

2



Příklad

Navrhněte číslicový obvod pro násobení
dvou čtyřbitových nezáporných čísel
včetně řízení

- výkonná jednotka ... sčítačka, obvody pro posuv
- řadič ... generování signálů pro zápis do registrů, posuv a okamžik sčítání
- registry pro násobenec, násobitel, dočasný výsledek a celkový výsledek

27.5.2009

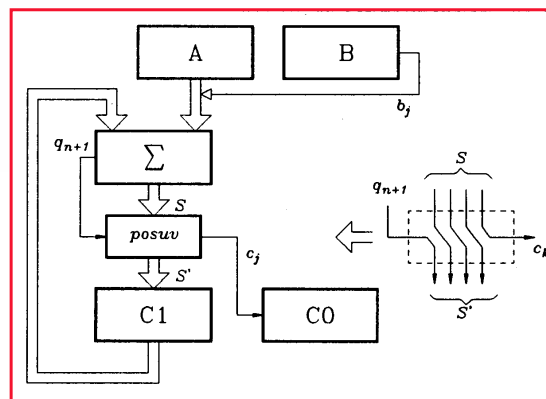
Y36SAP-procesor

5

Násobení ... SAP-7-arit

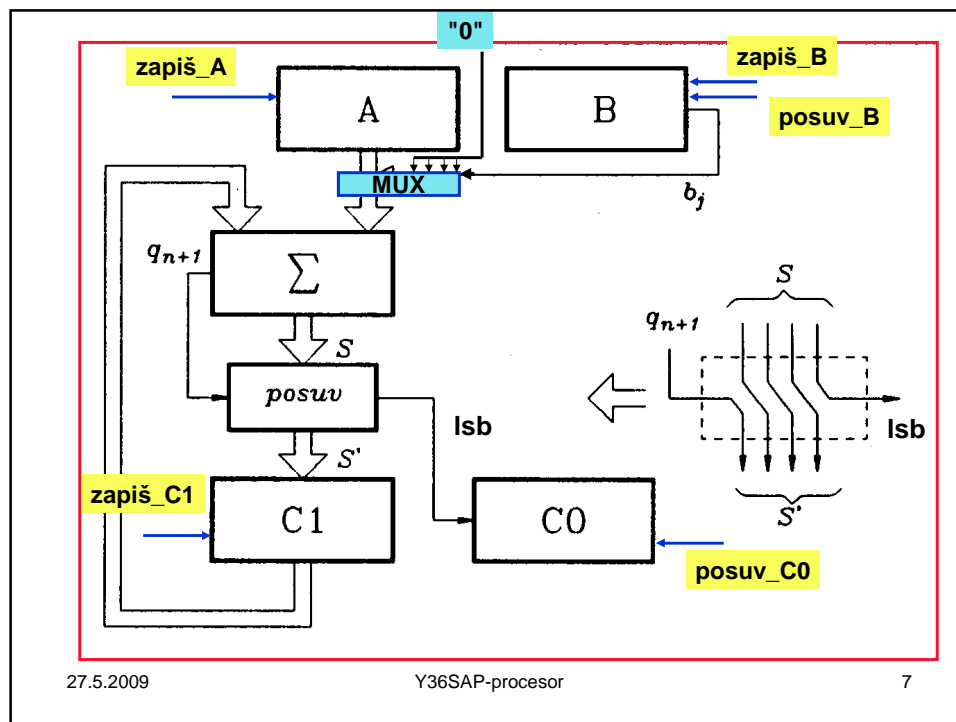
$$\begin{array}{r}
 \begin{array}{r}
 A \quad \times \quad B \\
 111 \times 101 \\
 \hline
 000 \quad \downarrow \downarrow \downarrow \\
 111 \quad \leftarrow \\
 011x \\
 000 \quad \leftarrow \\
 001x \\
 111 \quad \leftarrow \\
 \hline
 1000
 \end{array}
 = \begin{array}{c}
 C \\
 100011x \\
 \begin{array}{cc}
 C1 & C0 \end{array}
 \end{array}
 \end{array}$$

včetně řízení !!!



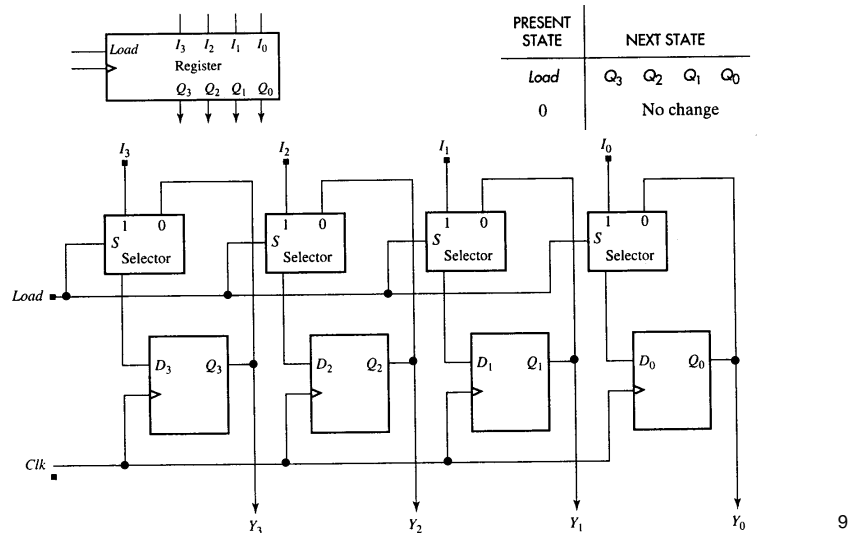
27.5.2009

6



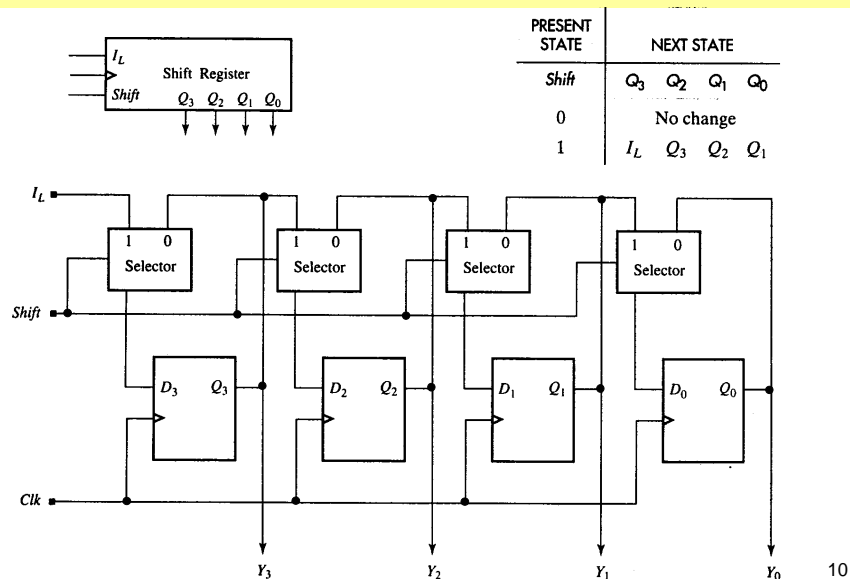
- výkonná jednotka ... 4 bitová sčítačka, obvody pro posuv ... dráty
- registry
 - A ... 4 bitový pro násobenec s paralelním vstupem
 - B ... 4 bitový pro násobitel, posuvný s paralelním vstupem
 - C1 ... 4 bitový pro dočasný výsledek s paralelním vstupem
 - C0 ... 4 bitový pro celkový výsledek posuvný
- řadič ... generování signálů pro zápis do registrů, posuv a okamžik sčítání signály ve žlutých boxech

Registr s řízením zápisu – „paralelní load“



9

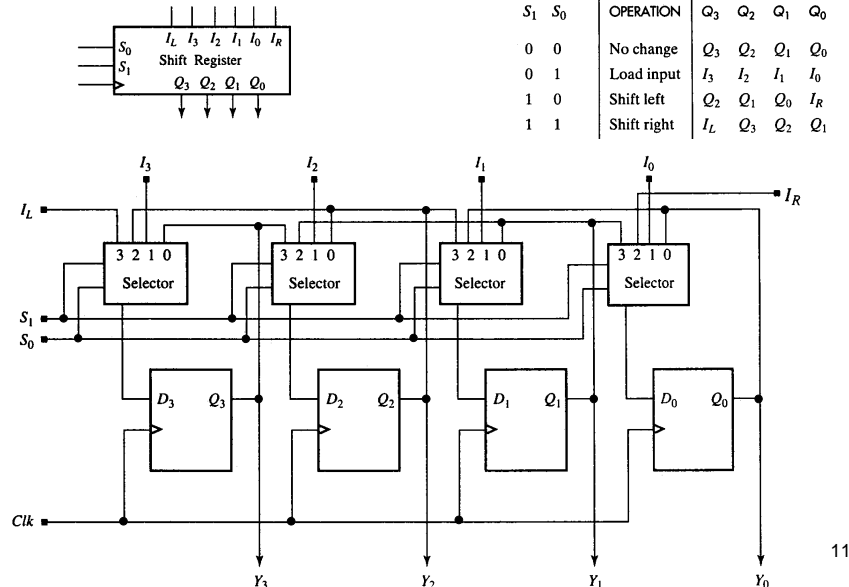
Posuvný registr ... nemá paralelní vstupy



10

Posuvný registr s řízením zápisu a směrem posuvu

.... nepotřebujeme posuv vlevo



Řadič

návrh řadiče graf přechodů

podle funkce

podle způsobu zadání vstupů a zobrazení
vstupu

podle toho, jak se to má realizovat

např. programovatelné obvody a stavebnice
(HW kit) ... LED diody, přepínače, tlačítka

problém času - ns x doba stisknutí tlačítka

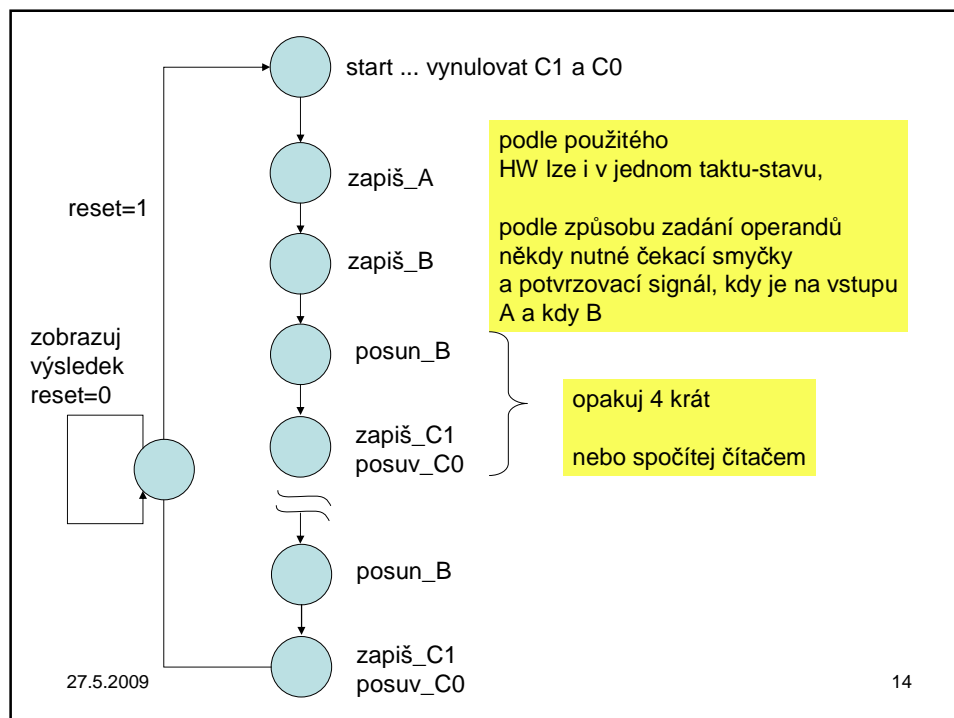
Násobení ... SAP-7-arit

$$\begin{array}{r}
 A \times B = C \\
 111 \times 101 = 100011 \\
 \hline
 000 \quad \downarrow\downarrow\downarrow \quad C1 \ C0 \\
 111 \quad \leftarrow \\
 0111 \\
 \hline
 000 \quad \leftarrow \\
 0011 \\
 \hline
 111 \quad \leftarrow \\
 1000
 \end{array}$$

27.5.2009

Y36SAP-processor

13



14

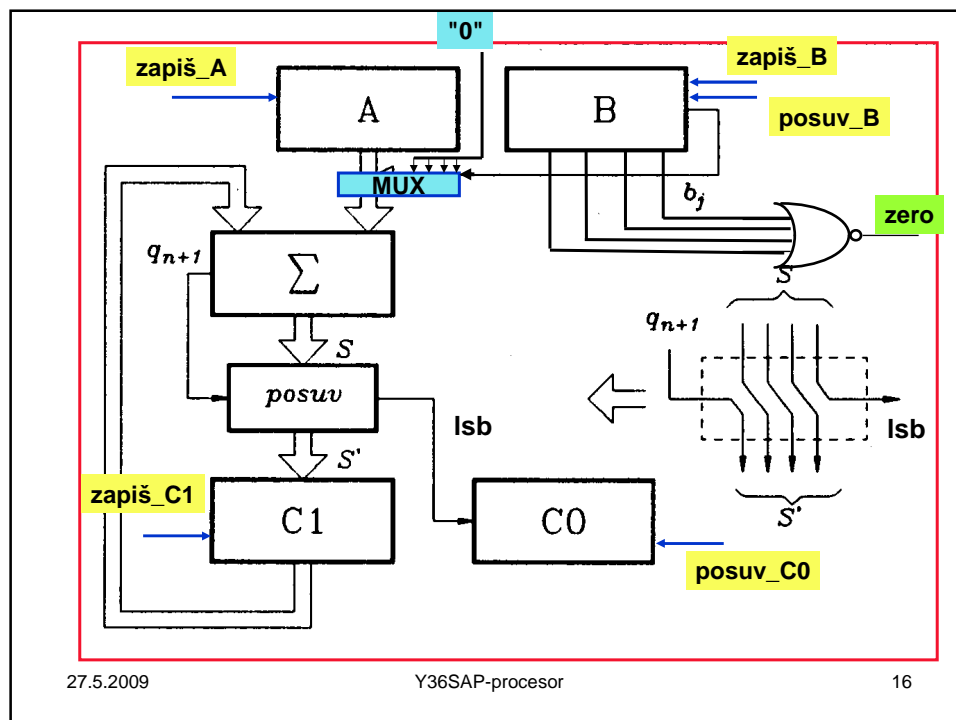
vylepšení, zrychlení

- když násobitel je "0", nemusí proběhnout celý cyklus (zde 4 takty)
- změna HW ... test na nulovost registru B
- změna řadiče jeho dalším vstupem je "stavový" signál o dílčích výsledcích, stavech, zde signál "zero"

27.5.2009

Y36SAP-procesor

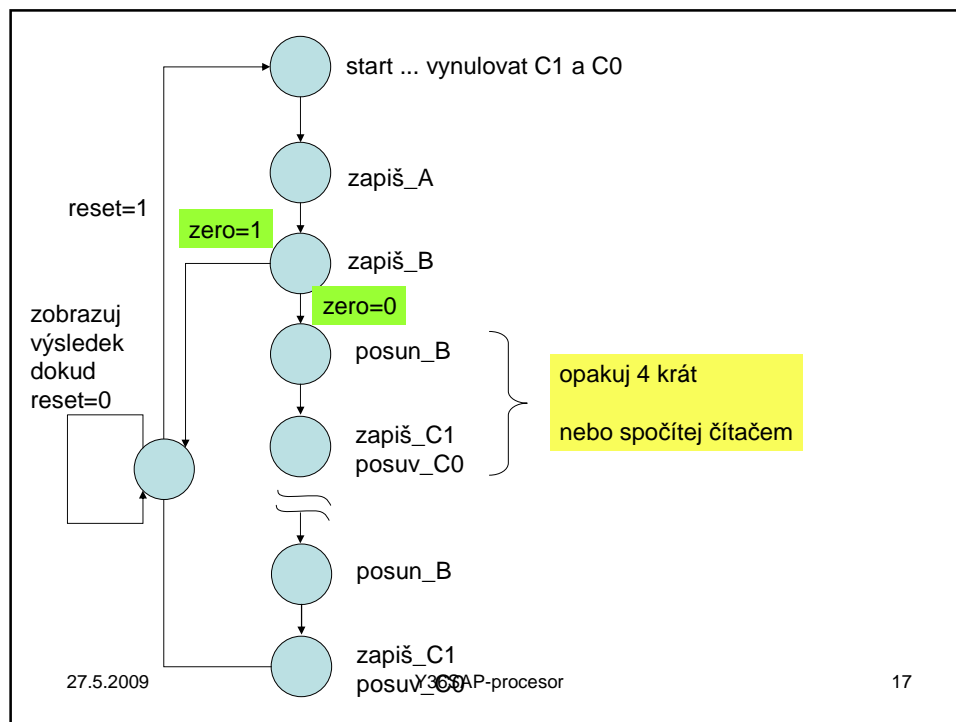
15



27.5.2009

Y36SAP-procesor

16



čítač M4 v binárním kódu

E \ q ₁ q ₀	0	1
00	00	01
01	01	10
10	10	11
11	11	00

běžné je použít hotový čítač
a
pracovat s moduly

E	q ₁	q ₀
0	1	0
1	0	1

$$D_{q_0} = q_0 \bar{E} + \bar{q}_0 E = q_0 \oplus E$$

E	q ₁	q ₀
0	0	1
0	1	0
1	0	1
1	1	0

$$D_{q_1} = \bar{q}_0 q_1 + q_1 \bar{E} + q_0 \bar{q}_1 E$$

27.5.2009

Y36SAP-procesor

18

Další konkrétní úpravy

- ovládání tlačítek
 - zadání dat
 - zobrazení výsledku
- podle toho, jak a čím
je to realizováno
např. na vývojové „stavebnici“
s programovatelným obvodem
- synchronní číslicový návrh - detekce konce či začátku posloupnosti jedniček či nul

27.5.2009

Y36SAP-procesor

19

Návrh procesoru ADOP

GPR architektura

Registry ... 16 registrů dostupných
programátorovi:

R0 – R11 universálních (datových) registrů

SP – ukazatel zásobníku

PC – programový čítač

PSW – stavový registr,

Příznaky Z ... zero, C ... carry, S ... sign, O ... overflow,
ES ... extended sign (znaménko 2. operandu v
binárních operacích)

ZR – obsahuje konstantní nulu

27.5.2009

Y36SAP-procesor

20

Návrh procesoru ADOP

- neviditelné registry - IR ... registr instrukce (pro dekódování instrukcí)
- jak dekódovat instrukce?
- jaké registry? jak přístupné?
- jak zaznamenat příznaky
- výkonné obvody - sčítačka
- organizace paměťového systému
- jaké vstupy a výstupy
- řadič

27.5.2009

Y36SAP-procesor

21

Kódování instrukcí

- 2 až 4 bytové instrukce
- Operační znak 2 slabiky, operand až 2 slabiky

3.půlbyte (15.-12.bit)	2.půlbyte (11.-8.bit)	1.půlbyte (7.-4.bit)	0.půlbyte (3.-0.bit)
---------------------------	--------------------------	-------------------------	-------------------------

Třetí půlbyte (nejvýznamnější bity operačního kódu) určuje skupinu instrukcí.

Nejvýznamnější tj. šestnáctý bit určuje délku instrukce:

0 – dvoubytové instrukce

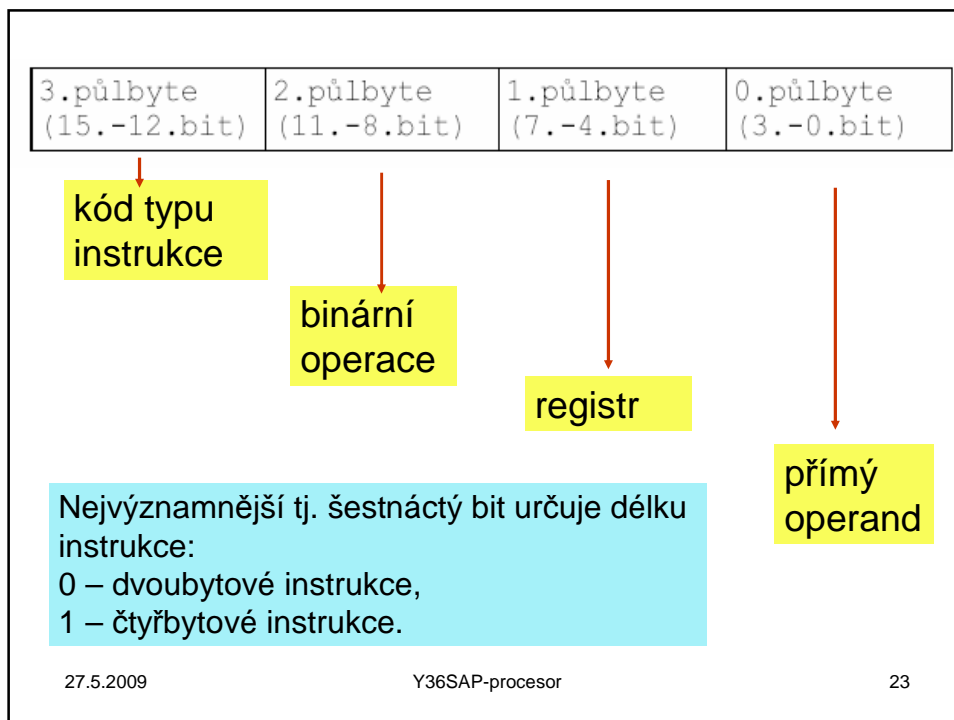
1 – čtyřbytové instrukce

Jména strojových instrukcí vyjadřují význam jednotlivých půlbytů operačního kódu.

27.5.2009

Y36SAP-procesor

22



Kód typu instrukce:

0. Basic - ...

- SHORTIMM_BOP_REGA_IMM4
- REGREG_BOP_REGA_REGB
- REGESIGN_BOP_REGA
- INDSHORT_BOP_REGA_REGB
-

27.5.2009 Y36SAP-procesor 24

Příklady

ADD R2, 2 kód: 1122
viz dokumentace.

1. SHORTIMM_BOP_REGA_IMM4 – binární operace mezi registrem **REGA** a operandem uloženým v 0. půlbytu operačního kódu. Výsledek je uložen do registru REGA – zde R2. Operand je 4-bitové číslo se znaménkem – zde 2

27.5.2009

Y36SAP-procesor

25

Binární operace BOP je kódována ve 2. půlbytu:

- 0. MOV,
- 1. ADD,
- 2. ADC,
- 3. SUB,
- 4. SBB,
- 5. AND,
- 6. OR,
- 7. XOR,
- 8. CMP,
- 9. SHL,
- 10. SHR,
- 11. ASR – aritmetický posun vpravo,
- 12. RRC,
- 13. RLC

Zde 1

27.5.2009

Y36SAP-procesor

26

40 02 MOV R0, [R2]

4. INDSHORT_BOP_REGA_REGB – binární operace mezi registrem REGA a operandem v paměti ležícím na adrese dané registrem REGB, výsledek je uložen do registru REGA. Operand v paměti je 8 bitové číslo se znaménkem.

MOV ... slide 24 – "0"

27.5.2009

Y36SAP-procesor

27

80 00 80 00 MOV R0, 32768

1000 0000 0000 0000 1000 0000 0000 0000

2.2. Čtyřbytové instrukce

0 LONGIMM_BOP_REGA_0_IMM – binární operace mezi registrem REGA a operandem uloženým v následujícím slově v paměti za operačním kódem. Výsledek je uložen do registru REGA. Operand je 16 bitové číslo.

2.4. Binární operace

Binární operace BOP je kódována ve 2. půlbytu:

0. MOV,

27.5.2009

Y36SAP-procesor

28

98 00 00 10 JGE 0010

1001 1000 0000 0000 absolutní adresa dosazená po překladu
v JSI návěští

1 JMP_COND_0_0_OFFS – podmíněný skok (relativní) na adresu OFFS, která je uložena v následujícím slově v paměti za operačním kódem.

Podmínka (COND) je kódována ve 2. půlbytu instrukcí skoku. Podmínky se rozdělují na přímé, které přímo testují příznaky procesoru a dále složené které testují kombinaci příznaků.

Přímé podmínky:

0. Z,
1. NZ,
2. C,
3. NC,
4. S,
5. NS,
6. O,
7. NO,

27.5.2009

Složené podmínky:

8. GE (greater or equal) – !SF && !OF || SF && OF
(výraz je ekvivalentní s SF == OF)
9. LT (less then) – negace podmínky GE,
10. GT (greater then) – GE && !Z
11. LE (less or equal) – negace podmínky GT,
12. AT (above then) – C && !Z
13. BE, (below equal) – negace podmínky AT
14. TRUE – podmínka vždy splněna

29

Simulátor

<http://service.felk.cvut.cz/jws/proc/procwww/>

umožňuje psát programy v JSI, překládat,
krokovat, spouštět a sledovat změny
obsahu registrů a paměti

Na této adrese je podrobný popis procesoru
ADOP, ze kterého vychází jeho HW návrh

27.5.2009

Y36SAP-procesor

30

Von Neumannova architektura (SAP1)

- Instrukce a data jsou uloženy v téže paměti.
- Paměť je organizována lineárně (tzn. jednorozměrně) a je rozdělena na stejně velké buňky, které se adresují celými čísly (zprav. 0, 1, 2, 3, . . .).
- Data ani instrukce nejsou explicitně označeny.
- Explicitně nejsou označeny ani různé datové typy.
- Pro reprezentaci dat i instrukcí se používají dvojkové signály.
- V instrukci zpravidla není uváděna hodnota operandu, ale jeho adresa.
- Instrukce se provádějí jednotlivě, a to v pořadí, v němž jsou zapsány v paměti, pokud není toto pořadí změněno speciálními instrukcemi (nazývanými skoky).

27.5.2009

Y36SAP-control unit

31

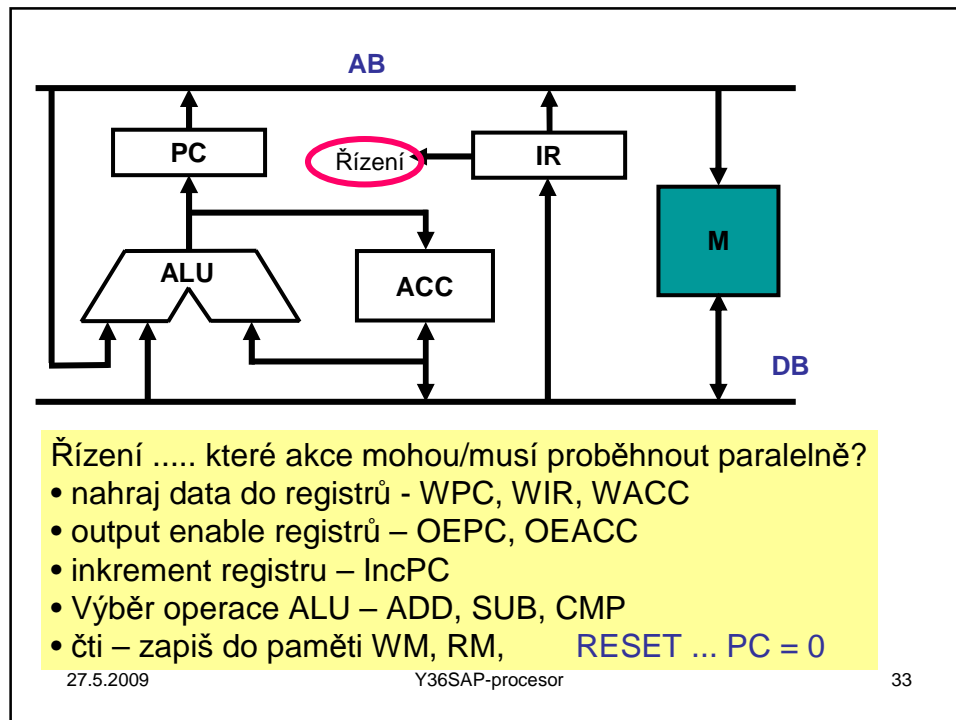
Řadič procesoru

- Pracuje podle instrukčního cyklu
- Řídí činnost všech výkonných jednotek počítače podle instrukcí a jejich kódu, podle **instrukčního cyklu**
- Je to sekvenční obvod – závisí na sekvenci vstupních (**stavových**) signálů, které generují výkonné jednotky (ALU, HP - instrukce) a vysílá jim **řídící** signály
- Pracuje v nekonečném cyklu – řídí zpracování instrukcí
- Navrhuje se podle instrukčního cyklu a výběru ISA – z grafu přechodů – vývojového digramu
- Podle způsobu jeho realizace existuje tzv. **obvodový** (klasický) řadič a **mikroprogramový** řadič

27.5.2009

Y36SAP-processor

32



HW návrh procesoru ADOP

GPR architektura

Registry ... 16 registrů dostupných programátorovi:

R0 – R11 universálních (datových) registrů

SP – ukazatel zásobníku

PC – programový čítač

PSW – stavový registr,

Příznaky Z ... zero, C ... carry, S ... sign, O ... overflow,
ES ... extended sign (znaménko 2. operandu v
binárních operacích)

ZR – obsahuje konstantní nulu

