

Procesor z pohledu programátora

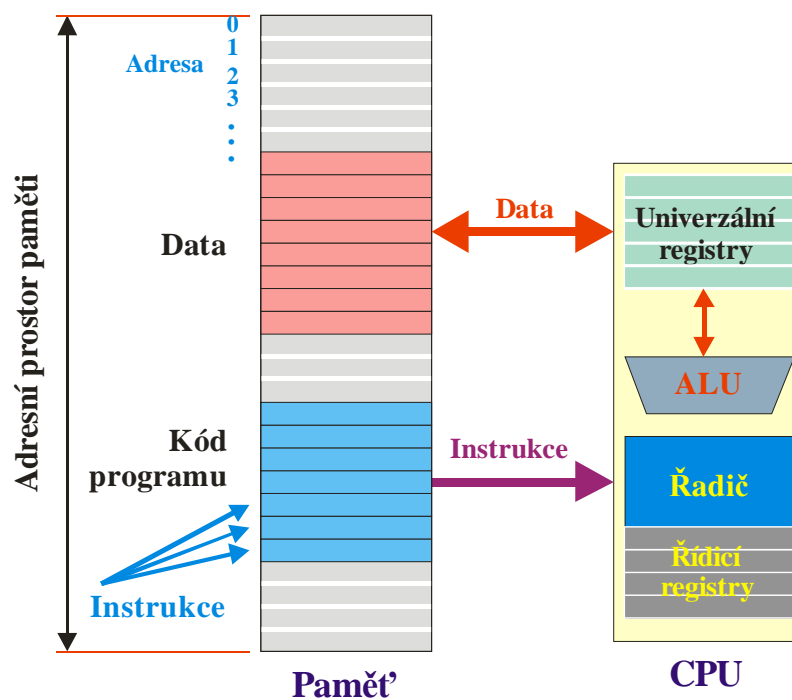
Terminologie

- **Procesor** (CPU) = řadič + ALU .
- **Mikroprocesor** = procesor vyrobený monolitickou technologií na 1 čipu.
- **Mikropočítač** = počítač postavený na bázi mikroprocesoru.
- **Mikrokontrolér** = mikropočítač se speciálními periferiemi vyrobený na 1 čipu.

„Programátorsky zajímavé“ vlastnosti procesoru

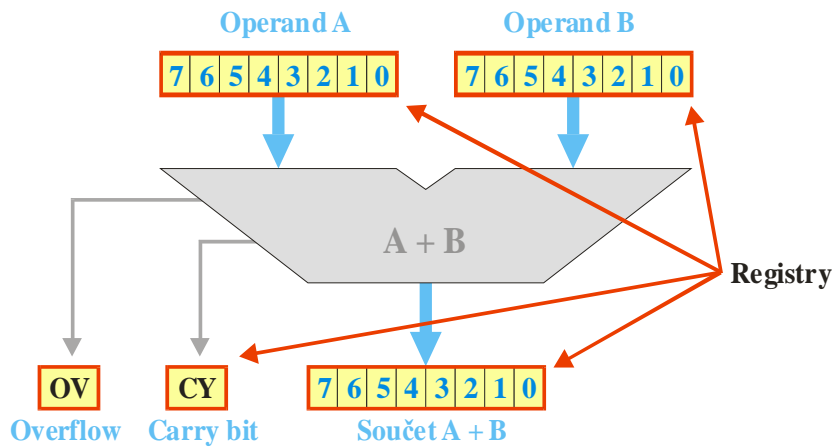
- ISA (Instruction Set Architecture):
 - Registrová sada procesoru
 - Instrukční soubor
 - Adresní prostory (paměti a IO)

Poznámka: CPU a paměť



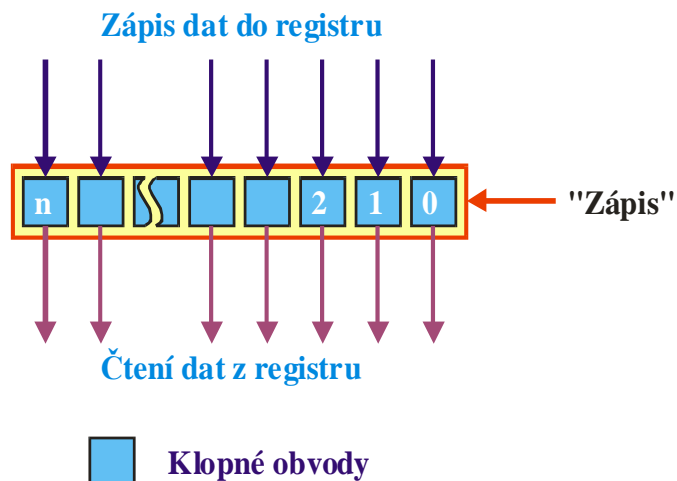
Poznámka: Registr

- Registr slouží v počítači jako dočasná paměť pro uložení určité hodnoty (operandy pro aritmetické operace, kód instrukce, ...).

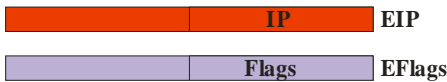
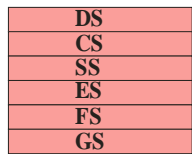
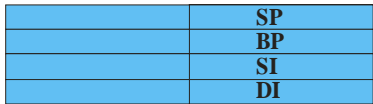
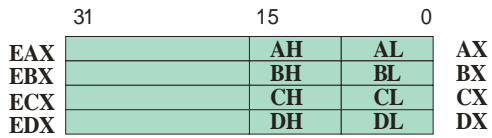


Poznámka: Registr

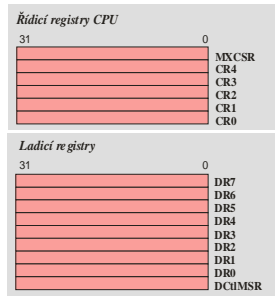
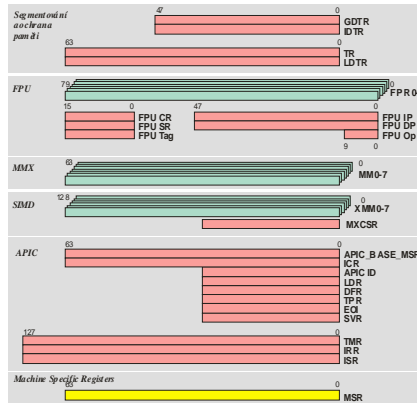
- Technická realizace: Registr je sestaven z klopných obvodů.



Registrová sada procesoru (Intel IA-32)

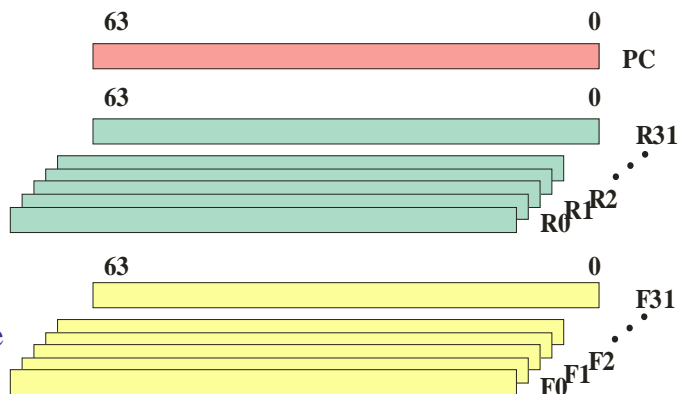


- 4 ks univerzálních registrů 32 bitů
- 4 ks bazových a indexových registrů
- 6 ks segmentových registrů
- IP (PC) a příznakový registr
- 8 ks FP registrů 80 bitů
- 8 ks SIMD registrů 128 bitů
- 8 ks MMX registrů 64 bitů



Registrová sada procesoru (Alpha)

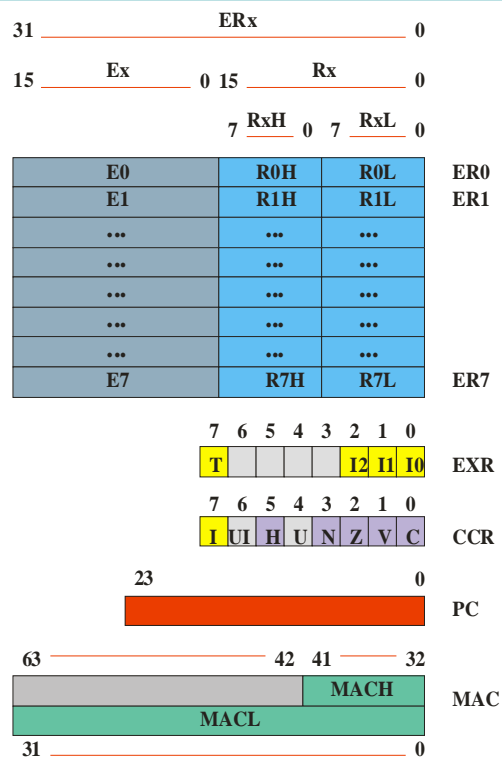
- 32 ks 64 bitových integer registrů,
- 32 ks 64 bitových float registrů,
- cca 50 řídicích registrů.
- Nemá příznakový registr.
- Všechny operace jsou typu registr ↔ registr.
- S pamětí pracují pouze instrukce Load a Store.



cca 50 řídicích registrů

Registrová sada procesoru H8S

- **RxL, RxH** – 8bitové registry
- **Rx, Ex** – 16bitové registry
- **ERx** – 32bitové registry
- **EXR** – řídicí registr
- **CCR** – příznakový registr
- **PC** – programový čítač (24 bitů)
- **MAC** – Multiply Accumulate
- **ER7** slouží implicitně jako **SP**



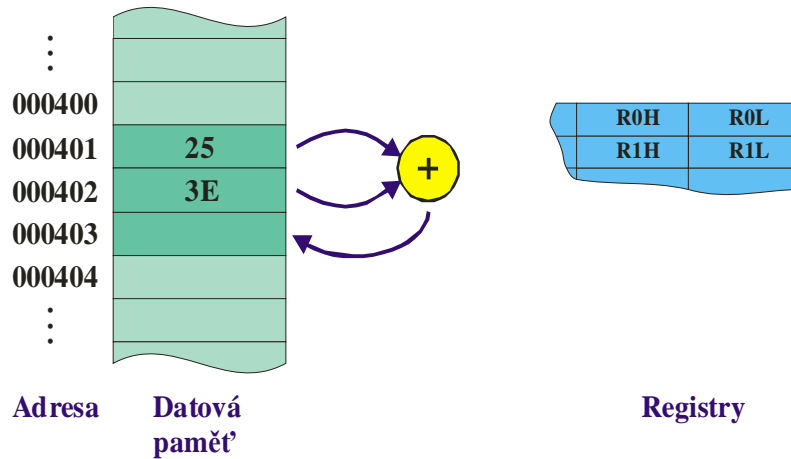
PC, SP, CCR, EXR

- **PC = Program Counter**
Při provádění určité instrukce obsahuje adresu následující instrukce.
- **SP = Stack Pointer**
Ukazatel na vrchol zásobníku.
- **CCR = Condition Code Register**
Obsahuje soubor příznaků pro větvení programu.
- **EXR = Extended Control Register**
Pro řízení přerušení a trasování.

Použití registrů

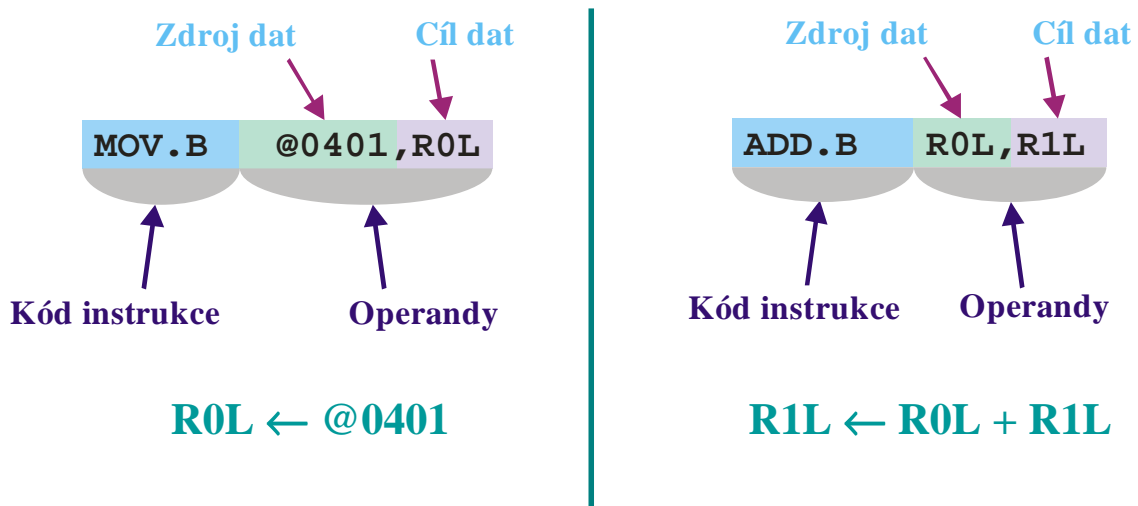
Jednoduchý program: sečtení dvou čísel v paměti, uložení výsledku do paměti

- H8S neumí pracovat s operandy v paměti → operandy se musí uložit do registrů,
- výsledek operace sčítání je v registru → musí se uložit do paměti.



Poznámka: Instrukce

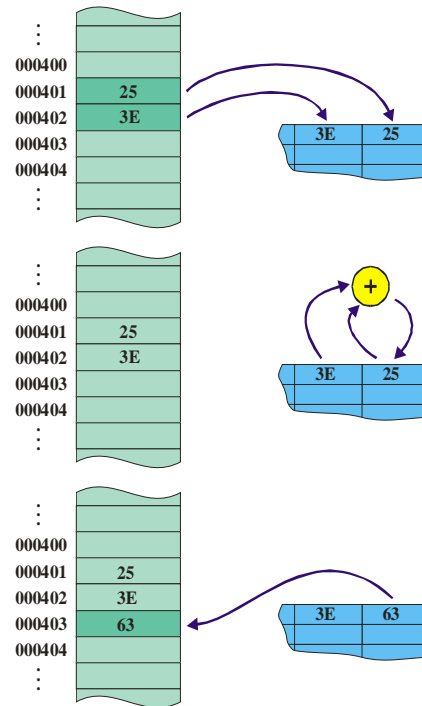
- Zjednodušeně:



Jednoduchý program

```

MOV.B @0401,R0L
MOV.B @0402,R0H
ADD.B R0H,R0L
MOV.B R0L,@0403
    
```

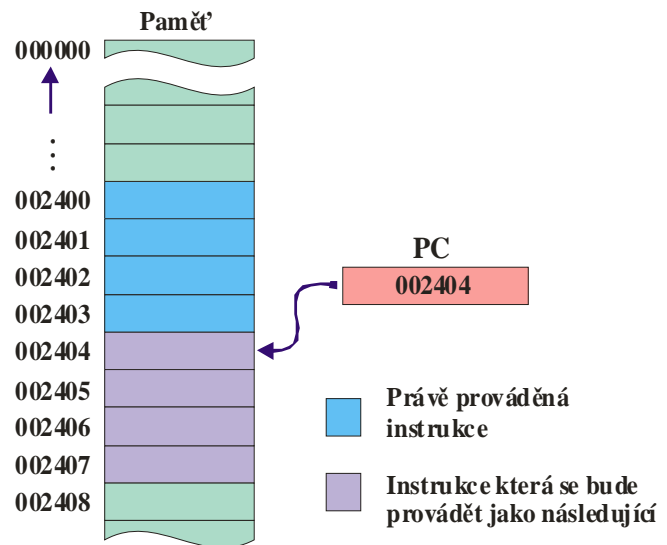


Registr PC

PC = Program Counter
(Programový čítač)

Při provádění určité instrukce obsahuje adresu následující instrukce.

V H8S je PC dlouhý 24 bitů ⇒ může adresovat paměť v rozsahu $2^{24} = 16$ M bytů.



Funkce PC

Registr CCR

- Obsahuje příznakové bity (flag)
- Nastavuje se podle výsledku provedené operace (kromě I)
- Použití: ke větvení programu

Příznakové bity u H8S:

		7	6	5	4	3	2	1	0	
C	Carry	I	UI	H	U	N	Z	V	C	CCR
V	Overflow									
Z	Zero									
N	Negative									
H	Half Carry									
I	Interrupt Enable									
U, UI	Nevyužité									

Větvení programu podle příznaků (1)

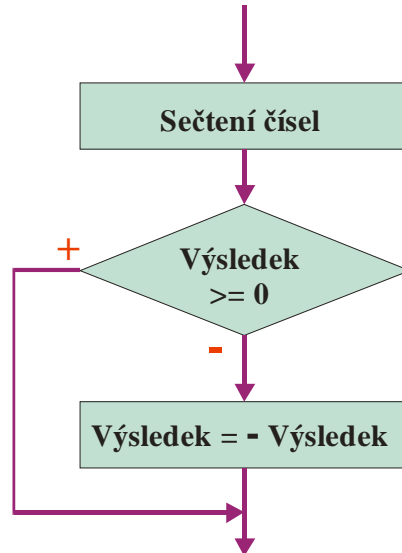
- K větvení programu slouží instrukce Bcc (Branch Conditionally).
- Testuje jeden nebo několik příznakových bitů.
- Podle výsledku testu buď provede nebo neprovede skok (přičtení určité hodnoty k PC).

Mnemonic	Meaning	cc	Condition	Signed/Unsigned*
BRA (BT)	Always (true)	0000	True	
BRN (BF)	Never (false)	0001	False	
BHI	High	0010	$C \vee Z = 0$	$X > Y$ (unsigned)
BLS	Low or Same	0011	$C \vee Z = 1$	$X \leq Y$ (unsigned)
BCC (BHS)	Carry Clear (High or Same)	0100	$C = 0$	$X \geq Y$ (unsigned)
BCS (BLO)	Carry Set (Low)	0101	$C = 1$	$X < Y$ (unsigned)
BNE	Not Equal	0110	$Z = 0$	$X \neq Y$ (unsigned or signed)
BEQ	Equal	0111	$Z = 1$	$X = Y$ (unsigned or signed)
BVC	Overflow Clear	1000	$V = 0$	
BVS	Overflow Set	1001	$V = 1$	
BPL	Plus	1010	$N = 0$	
BMI	Minus	1011	$N = 1$	
BGE	Greater or Equal	1100	$N \oplus V = 0$	$X \geq Y$ (signed)
BLT	Less Than	1101	$N \oplus V = 1$	$X < Y$ (signed)
BGT	Greater Than	1110	$Z \vee (N \oplus V) = 0$	$X > Y$ (signed)
BLE	Less or Equal	1111	$Z \vee (N \oplus V) = 1$	$X \leq Y$ (signed)

Větvení programu podle příznaků (2)

Příklad:

Výpočet absolutní hodnoty ze součtu dvou čísel.



Větvení programu podle příznaků (3)

Příklad:

Výpočet absolutní hodnoty ze součtu dvou čísel.

	MOV.B	@0401,R0L	; 1. operand do R0L
	MOV.B	@0402,R0H	; 2. operand do R1L
	ADD.B	R0H,R0L	; nastaví příznaky
	BPL	LAB1	; skok při N == 0
	NEG.B	R0L	; dvojkový doplněk
→	LAB1: MOV.B	R0L,@0403	; uložení výsledku

Poznámka:

LAB1 je návěští, které označuje instrukci **MOV.B R0L,@0403**.

Použití dalších registrů

- Také registry **SP** a **EXR** mají speciální použití.
- **SP** je pro nás důležitý, ale nejprve probereme obecné možnosti adresování dat a instrukcí v paměti a kódování instrukcí.
- **EXR** má využití při obsluze přerušení a při ladění programu → **pro nás** není důležitý.