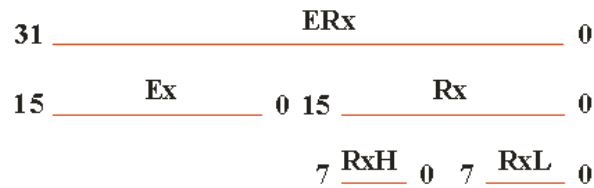


Programování H8S

Registry procesoru H8S volně k dispozici pro programátora



E0	R0H	R0L	ER0
E1	R1H	R1L	ER1
...	
...	
...	
...	
...	
...	
E7	R7H	R7L	ER7

Máme 8 registrů, které mají 32 bitů. Tyto registry můžeme rozdělit na 16 registrů po 16 bitech, nebo na 32 registrů s 8 bity.

Základní operace s registry

Přesuny mezi registry:

MOV.B zdroj, cíl byte – přesun (8 bitů)
MOV.W zdroj, cíl word - přesun (16 bitů)
MOV.L zdroj, cíl long - přesun (32 bitů)

Př.

```
mov.w R1,R0 ;přesun obsahu registru R1 do R0
```

Uložení konstanty do registru

```
mov.w #konst:16, R0
```

Uložení hodnoty z paměti do registru

```
mov.w @adresa:rozsah, R0
```

Případně se jako adresa může uvést proměnná, deklarovaná v datové části:

```
.h8300s
```

```
.data ;datová část
```

```
var1: .word 0x1234 ;proměnná, s definovanou počáteční hodnotou
```

```
.text ;programová část
```

```
.global _start
```

```
_start: mov.w @var1,R0 ;načtení hodnoty paměťového prostoru, kam ukazuje „var1“
```

```
lab1: jmp @lab1 ;nekonečná smyčka
```

```
.end
```

Základní matematické operace

Sčítání

Byte:

ADD.b #konstanta:8,registr

ADD.b registr,registr

Word:

ADD.w #konstanta:16,registr

ADD.w registr,registr

Long...

Inkrementace

Byte:

inc.b registr

Word:

inc.w #1,registr ;inkrementace o 1

inc.w #2,registr ;inkrementace o 2

Long...

Skoky

Nepodmíněný skok:

JMP @návěstí

JMP @registr (skok na adresu v registru)

Skok do podprogramu:

JSR @návěstí

JSR @registr (skok na adresu v registru)

Návrat z podprogramu:

RTS

Zásobník

Prostor pro zásobník musíme deklarovat v datové části paměťového prostoru a na začátku programu do *stack pointer* registru uložit jeho vrchol.

```
.data          ;datova část
.space 100     ; 100 bytu pro zásobník
stack:        ;deklarace ukazatele zásobníku (na konci pametoveho prostoru pro zasobnik)

.text         ;programova cast
.global _start
_start: mov.l #stack,ER7 ; hodnota stack do SP
```

Uložení čísla do zásobníku, operace PUSH

- Posuv zásobníku o odpovídající počet byte směrem dolů.
- Uložení čísla z registru do paměti (zásobníku).

Přečtení čísla ze zásobníku, operace POP

- Přečtení čísla z paměti z místa, kam ukazuje zásobník.
- Posuv zásobníku o patřičný počet byte směrem nahoru.

U H8S se jedná o příkazy:

PUSH.W registr

PUSH.L registr

POP.W registr

POP.L registr

Příklad:

Převod šestnáctkového čísla uloženého v proměnné jako znak (0..9,A..F příp. a..f)

```
.h8300s

.data ;datova cast
char1: .word 0x0066

.text ;
.global _start
_start: mov.w @char1,R0
        cmp.b #0x3A:8,R0I
        bgt znak:16
cislo:  sub.w #0x30:16,R0
        jmp @lab1

znak:   cmp.b #0x47:8,R0I
        bgt maly:16

velky:  sub.w #0x37:16,R0
        jmp @lab1

maly:   sub.w #0x57:16,R0

lab1:   jmp @lab1
        .end
```

Obrácený převod:

```
.h8300s

.data ;datova cast
char1: .word 0x000F

.text ;
.global _start
_start: mov.w @char1,R0
        cmp.b #0x09:8,R0I
        bgt znak:16
cislo:  add.w #0x30:16,R0
        jmp @lab1

znak:   add.w #0x37:16,R0

lab1:   jmp @lab1

        .end
```

Tabulka ASCII znaků použitých v šestnáctkové soustavě

znak	hodnota desítkově	hodnota šestnáctkově
0	48	0x30
1	49	0x31
...
9	57	0x39
A	65	0x41
...
F	70	0x46
a	97	0x61
...
f	102	0x66