

Programiranje za elektrotehnike 1

Osnove programiranja v zbirnem jeziku

Vsebina poglavja

- Primerjava - Zbirni jezik : C
- Zbirni jezik
 - Ukazi v zbirnem jeziku
 - Delovanje zbirnika (prevajalnika)

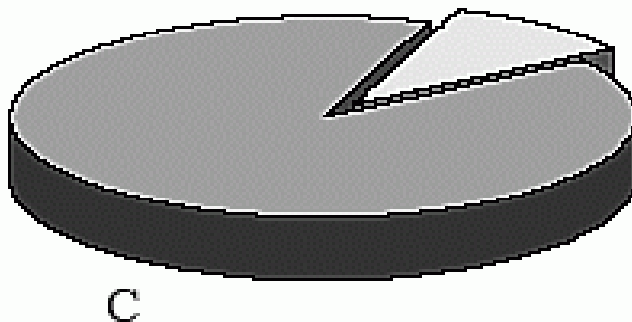
Primerjava - Zbirni jezik : C

- Zbirni jezik:
 - Hitrejše izvajanje
 - Zasede manj pomnilnika
 - Zahteva manj zmogljive procesorje
 - Natančno določimo obnašanje programske opreme
 - Možno je izkoristiti vse prednosti nabora ukazov procesorja
- C:
 - Krajši program
 - Bolj pregleden program
 - Preprostejše vzdrževanje
 - Hitrejši razvoj aplikacij
 - Bolj fleksibilen razvoj aplikacij
 - Včasih se koda ne prevede tako, kot smo predvideli

Uporaba C in zbirnega jezika

Tradicionalni programerji

Zbirni jezik



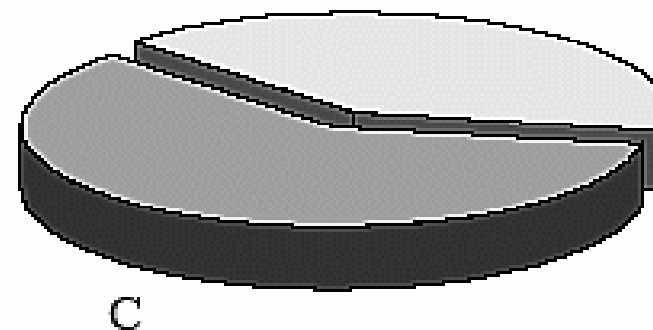
10% programerjev uporablja zbirni jezik!

DSP – več zbirnega jezika – maksimalna hitrost izvajanja

Široka potrošnja – optimalna cena – prevladuje zbirni jezik

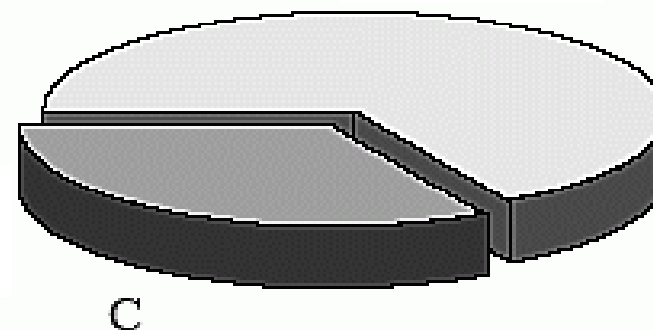
DSP programerji

Zbirni jezik



DSP programi za široko potrošnjo

Zbirni jezik



Zbirni jezik

- Ukazi v zbirnem jeziku:
 - Deli ukaza v zbirnem jeziku
 - Nalaganje in shranjevanje
 - Aritmetične operacije
 - Inkrement/dekrement
 - Logične operacije
 - Pomiki
 - Brisanje in postavljanje bitov
 - Brezpogojni skoki
 - Pogojni skoki
- Delovanje zbirnika (prevajalnika)

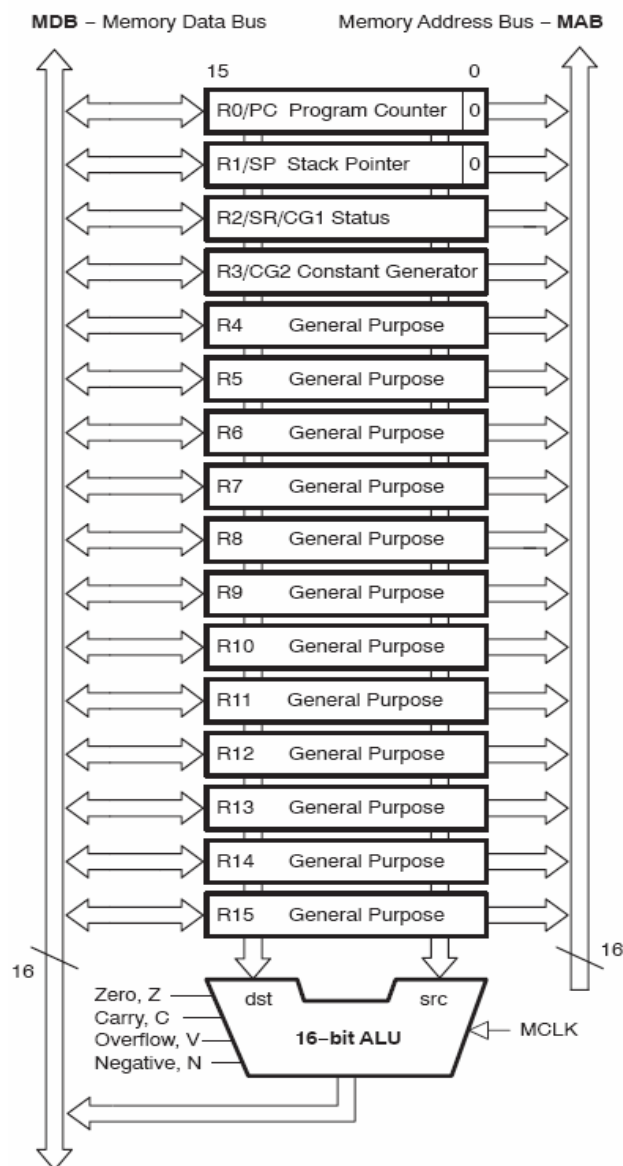
Naslov in vsebina

Naslov	Vsebina
0xEB01	0x0041
0xEB02	0x0032
0xEB03	0x0FF1
0xEB04	0xC011
0xEB05	0xF000

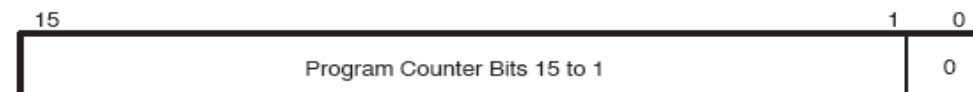
Opis ciljnega sistema

- Primer preprostega zbirnega jezika
 - Mikrokontroler MSP430F2013
- Osnovni ukazi (RISC)
 - 27 ukazov
 - 7 tipov naslavljanj
- Spremenljivke in konstante
 - Simboli:
 - src Izvorni operand (source), določen z As in S-reg.
 - dst Ciljni operand (destination), določen z Ad in D-reg.
 - .B/.W Operacija nad zlogom (byte) ali besedo (word):
 - Rx (R0-R15) Delovni registri
 - PC Programski števec
 - SP Kazalec na sklad
 - TOS Vrh sklada
 - SR Statusni register
 - WDT Watchdog Timer
 - y(Rx) Vsebina lokacije, ki jo poda (Rx + y), npr. 2(R5), indeksno naslavljanje
 - SPR Spremenljivka, simbolično.
 - &SPR Vsebina na naslovu, ki ga podaja spremenljivka SPR.
 - @Rx Vsebina na naslovu, ki ga podaja vsebina registra Rx, indirektno naslavljanje.
 - #k Konstanta, npr. #45h pomeni konstanto 45 v šestnajstičnem številskem sistemu.

Delovni registri



- PC (Program Counter)
 - Programski števec



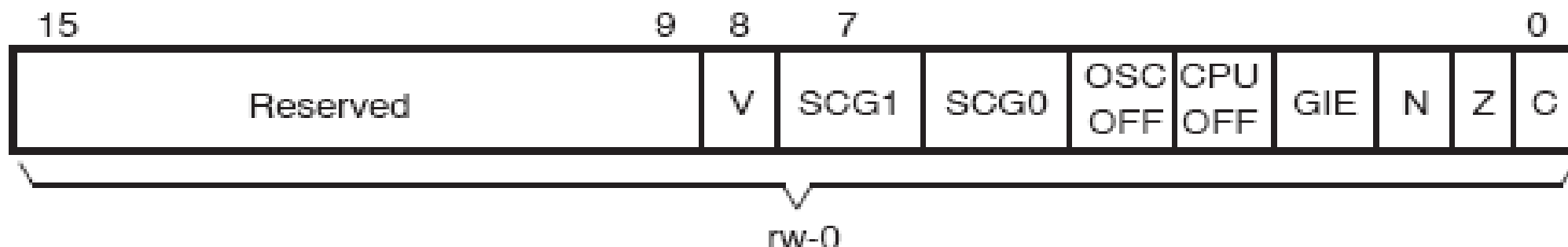
- SP (Stack Pointer)
 - Kazalec na sklad



- Prosto uporabni registri:
 - R4, ..., R15

Statusni register

- SR



- C – Bit za prenos (Carry bit)
- Z – Ničelni bit (Zero bit)
- V – Bit za prekoračitev obsega (Overflow bit)
- N – Bit za negativno vrednost (Negative bit)
- GIE – Bit za omogočitev prekinitev (General Interrupt Enable)

Deli ukaza v zbirnem jeziku

- labela
- operacijska koda
- operandi
- komentarji

OUTPUT	MOV.W	R6, R7	;prenesi vsebino R6 v R7
	CLRC		;zbriši bit C v SR

Nalaganje in shranjevanje

- Nalaganje v akumulator oz. pomnilnik

- MOV.W src, dst

C: dst = src;

- (src) → (dst)

- Registrsko naslavljanje:

MOV.W R10,R11

C: R11 = R10;

- Indeksno naslavljanje:

MOV 2(R5),6(R6)

C: *(R6+6) = *(R5+2);

- Simbolno naslavljanje:

MOV SPR1, SPR2

C: *SPR1 = *SPR2;

- Absolutno naslavljanje:

MOV &SPR1, &SPR2

C: *SPR1 = *SPR2;

- Indirektno registrsko naslavljanje:

MOV.B @R10,0(R11)

C: *R10 = *R11;

- Indirektno registrsko naslavljanje z avtoinkrementom:

MOV.B @R10,0(R11)

C: *(R10++) = *R11;

- Trenutno naslavljanje:

MOV.W #45, SPR1

C: *SPR1 = 45;

Aritmetične operacije

- Seštevanje:

- ADD src, dst

- $\text{src} + \text{dst} \rightarrow \text{dst}$

- Primer: ADD.W R4, R5

C: $\text{dst} = \text{src} + \text{dst};$

C: $R5 = R4 + R5;$

- ADDC src, dst

- $\text{src} + \text{dst} + C \rightarrow \text{dst}$

- Primer: ADDC.W R4, R5

C: $\text{dst} = \text{src} + \text{dst} + C;$

C: $R5 = R4 + R5 + C;$

- Odštevanje:

- SUB src, dst

- $\text{dst} - \text{src} \rightarrow \text{dst}$

- Primer: SUB.W #10h, R5

C: $\text{dst} = \text{src} - \text{dst};$

C: $R5 = R5 - 0x10;$

- SUBC src, dst

- $\text{dst} - \text{src} - 1 + C \rightarrow \text{dst}$

- Primer: SUBC.W Spr1, R6

C: $\text{dst} = \text{src} - \text{dst} - 1 + C;$

C: $R6 = R6 - *Spr1 - 1 + C;$

Inkrement in dekrement

- Inkrement:

- INC dst

- $dst + 1 \rightarrow dst$

- Primer: INC.W Spr1

C: $dst = dst + 1;$

C: *Spr1++;

- Dekrement:

- DEC dst

- $dst - 1 \rightarrow dst$

- Primer: DEC.W Spr1

C: $dst = dst - 1;$

C: *Spr1--;

Logične operacije

- **IN:**
 - **AND src, dst**
 - $\text{src} \& \text{dst} \rightarrow \text{dst}$
 - Primer: `AND.W #1h, R5`

C: $\text{dst} = \text{src} \& \text{dst};$
C: $\text{R5} = \text{R5} \& 0x1;$
- **ALI:**
 - **BIS src, dst**
 - $\text{src} | \text{dst} \rightarrow \text{dst}$
 - Primer: `BIS.W Spr1, R6`

C: $\text{dst} = \text{src} | \text{dst};$
C: $\text{R6} = \text{R6} | *Spr1;$
- **Ekskluzivni ALI:**
 - **XOR src, dst**
 - $\text{src} \wedge \text{dst} \rightarrow \text{dst}$
 - Primer: `XOR.B B'0001, R5`

C: $\text{dst} = \text{src} \wedge \text{dst};$
C: $\text{R5} = \text{R5} \wedge 1;$
- **NE:**
 - **INV dst**
 - $\sim \text{dst} \rightarrow \text{dst}$
 - Primer: `INV.W R5`

C: $\text{dst} = \sim \text{dst};$
C: $\text{R5} = \sim \text{R5};$

Pomiki (shift & rotate)

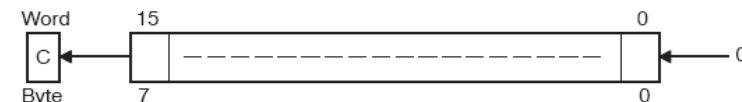
- Aritmetična rotacija v levo:

- RLA *dst*

- Primer: RLA.W Spr1

C: $dst = dst * 2;$

C: $*Spr1 = *Spr1 * 2;$



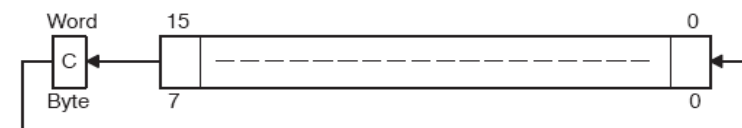
- Rotacija v levo skozi C:

- RLC *dst*

- Primer: RLC.W Spr1

C: $dst = dst \ll 1;$

C: $*Spr1 = *Spr1 \ll 1;$



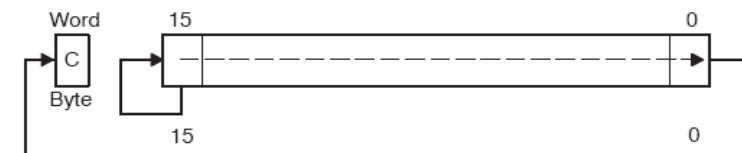
- Aritmetična rotacija v desno:

- RRA *dst*

- Primer: RRA.W Spr1

C: $dst = dst / 2;$

C: $*Spr1 = *Spr1 / 2;$



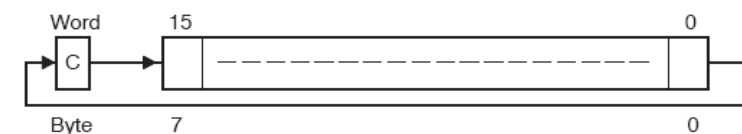
- Rotacija v desno skozi C:

- RRC *dst*

- Primer: RRC.W Spr1

C: $dst = dst \gg 1;$

C: $*Spr1 = *Spr1 \gg 1;$



Postavljanje in brisanje bitov

- **Brisanje bitov**

- BIC src, dst

- $\sim\text{src} \& \text{dst} \rightarrow \text{dst}$

- Primer: BIC.W #1h, Spr1

C: $\text{dst} = \text{dst} \& \sim\text{src};$

C: $*\text{Spr1} = *\text{Spr1} \& (\sim 0\text{x1});$

- CLR dst

- $0 \rightarrow \text{dst}$

- Primer: CLR.W Spr1

C: $\text{dst} = 0;$

C: $*\text{Spr1} = 0;$

- **Postavljanje bitov**

- BIS src, dst

- $\text{src} | \text{dst} \rightarrow \text{dst}$

- Primer: BIS.W Spr1,0

C: $\text{dst} = \text{src} | \text{dst};$

C: $*\text{Spr1} = *\text{Spr1} | 0\text{x1};$

Brezpogojni skoki in zanke

- Brezpogojni skok:

- JMP dst

- $PC + 2 * \text{offset} \rightarrow PC$

- Primer: JMP LABELA

C: goto LABELA ;

- Skok v podprogram:

- CALL dst

- $\text{dst} \rightarrow \text{tmp}; SP - 2 \rightarrow SP; PC \rightarrow @SP; \text{tmp} \rightarrow PC$

- Primer: CALL funkcija

C: funkcija();

- RET

- Vrnitev iz podprograma, $@SP \rightarrow PC; SP + 2 \rightarrow SP$

- Primer: RET

C: return;

- Brezpogojna vejitev:

- BR dst

- $\text{dst} \rightarrow (PC)$

- Primer: BR LABELA

Pogojni skoki

- Skoči, če je postavljen C / višje ali enako:
 - JC dst / JHE dst
 - if C == 1:
 - PC + 2 * offset → PC
 - if C == 0:
 - izvedi naslednji ukaz
 - Primer: JC LABELA

- Skoči, če je postavljen Z / enako:
 - JZ dst / JEQ dst
 - if Z == 1:
 - PC + 2 * offset → PC
 - if Z == 0:
 - izvedi naslednji ukaz
 - Primer: JZ LABELA

- Skoči, če je postavljen N:
 - JN dst
 - if N == 1:
 - PC + 2 * offset → PC
 - if N == 0:
 - izvedi naslednji ukaz
 - Primer: JN LABELA

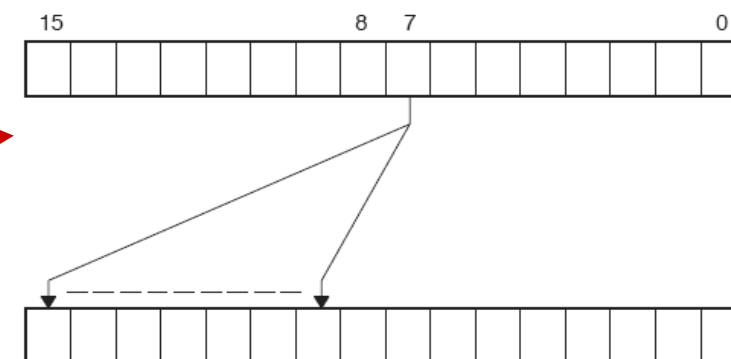
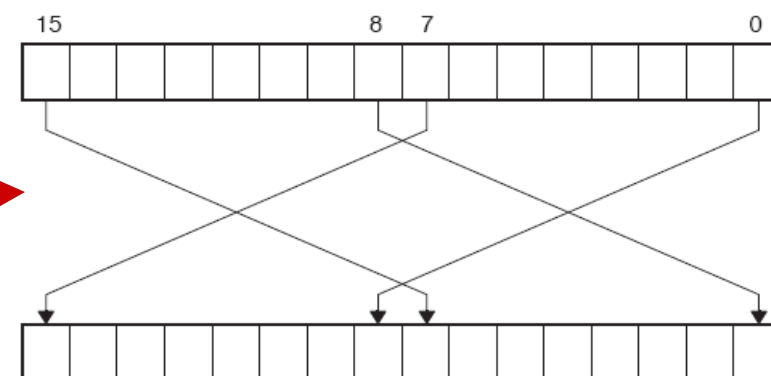
- Skoči, če ni postavljen C / nižje:
 - JNC dst / JLO dst
 - if C == 0:
 - PC + 2 * offset → PC
 - if C == 1:
 - izvedi naslednji ukaz
 - Primer: JC LABELA

- Skoči, če ni postavljen Z / ni enako:
 - JNZ dst / JNE dst
 - if Z == 0:
 - PC + 2 * offset → PC
 - if Z == 1:
 - izvedi naslednji ukaz
 - Primer: JZ LABELA

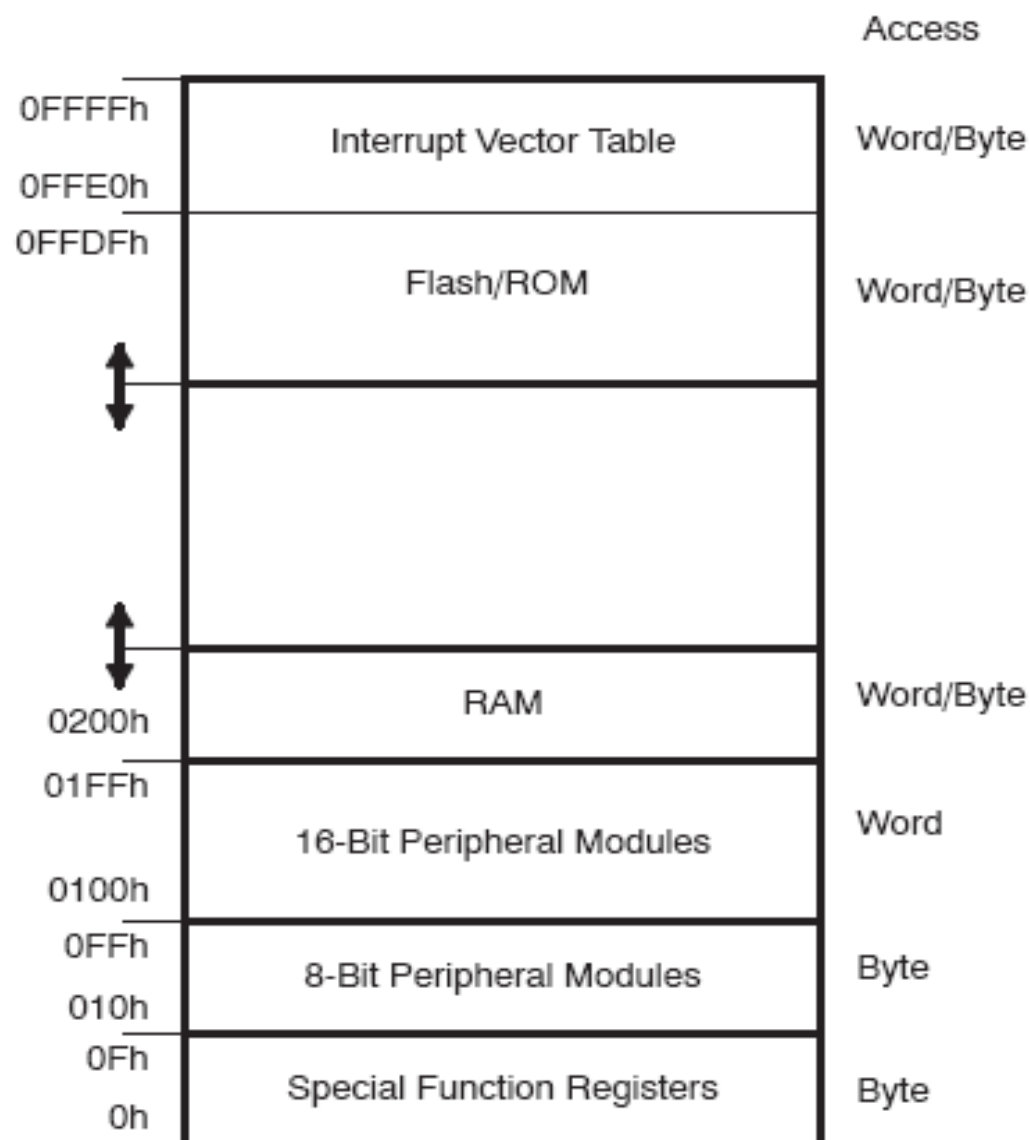
- Skoči, če je večje ali enako:
 - JGE dst
 - if (N ^ V) == 0:
 - PC + 2 * offset → PC
 - if (N ^ V) == 1:
 - izvedi naslednji ukaz
 - Primer: JGE LABELA

Še nekaj ukazov

- Ukaz za izvedbo primerjave:
 - CMP src, dst
 - dst - src
 - Primer: CMP R5, R6
- Ukaz za izmenjavo zlogov:
 - SWPB dst
 - Primer: SWAP Spr1
- Ukaz “Ni operacije”:
 - NOP
 - Ne naredi ničesar, samo poveča PC
 - Primer: NOP
- Ukaz za razširitev predznaka:
 - SXT dst
 - Primer: SXT R6



Organizacija pomnilnika



Delovanje zbirnika (prevajalnika)

- Prevajalnik (Assembler)
- Povezovalnik (Linker)
- Knjižnice (Library)
- Generator binarne kode (Executable file)

Domača naloga

Možna vprašanja na izpitu

- Podaj primerjavo med zbirnim jezikom in C!
- Opiši dele ukaza v zbirnem jeziku!
- Naštej tipe ukazov v zbirnem jeziku!