

OPASKA!

- Ovi materijali namijenjeni su isključivo studenticama/studentima koji su upisali predmet “Računala i procesi” na FER-u u šk. g. 2002/2003.
- Za svako drugo korištenje potrebna je pismena suglasnost autora!
- Materijali služe kao pomoć u praćenju predavanja, a ne kao njihova zamjena te se ne mogu tumačiti izvan konteksta predavanja!

M. Žagar, 2002-10-01



Mario Žagar



Sveučilište u Zagrebu

**Fakultet elektrotehnike i računarstva
(FER)**

13. RIP - viši programski jezici i assembler

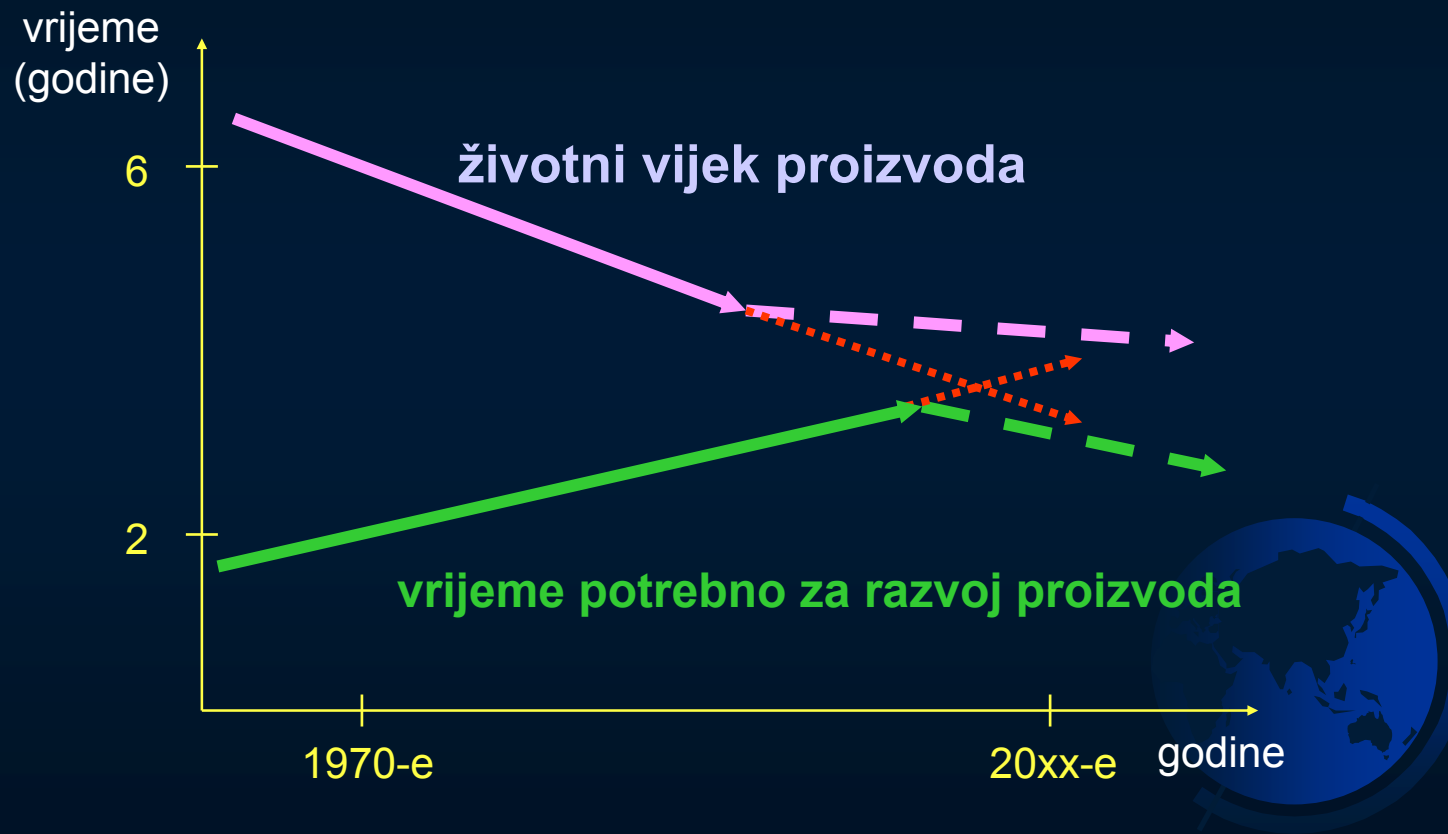


Ugrađena računala

- Sastavni dio nekog uređaja (konačne dimenzije!!)
- svojstva mikroračunala određena svojstvima uređaja
- pristup unikatni - “po mjeri” (vrlo rijetko konfekcijski)
- vrijeme razvoja uređaja u pravilu što kraće



Razvoj - životni vijek proizvoda



Korištenje HLL

☞ Prednosti

- veća efikasnost
- čitljiviji programi
- lakše testiranje
- bolja dokumentacija

☞ Nedostatci

- veći kod
- sporiji program



Viši programski jezici

- Formalni opisni jezik kojim programer opisuje što mikroprocesor treba napraviti.
- Notacija kojom se programer služi lakša je za snalaženje nego u assembleru.
- Svaka naredba u višem programskom jeziku odgovara nizu naredaba u strojnom jeziku.

U nastavku, primjeri programskog jezika C, C--



Primjer:

```
void main ( )
{
    int a;
    int data b;
    char c;

    a= -15;
    b= 0x1532;
    c= 10;

    a++;
    a=a+b;
    a=b+c;
```

```
    modul ( a );
    while ( b<1000 ) {
        b=b+a;
    }
}
int modul ( int x )
{
    if ( x > 0) {
        return ( x );
    } else {
        return ( -x );
    }
}
```


Z80

I8051

PIC16c54

a = -15; / a je tipa int */*

```
DEC SP
PUSH BC
PUSH BC
LD HL,3
ADD HL,SP
EX DE,HL;;
LD HL,-15
CALL CCPINT##
```

```
MOV R5,#0xF1
MOV R4,#0xFF
```

```
MOVLW 0F1h
MOVWF ?(a_main+0)
MOVLW 0FFh
MOVWF ?(a_main+0+1)
```

b = 0x1532; / b je tipa int kod i8051 se nalazi u DATA */*

```
LD HL,1
ADD HL,SP
EX DE,HL;;
LD HL,5426
CALL CCPINT##
```

```
MOV 0x08,#0x15
MOV 0x09,#0x32
```

```
MOVLW 032h
MOVWF ?(a_main+2)
MOVLW 015h
MOVWF ?(a_main+2+1)
```

Z80

I8051

PIC16c54

c = 10; /* a je tipa int*/

LD HL,0
ADD HL,SP
EX DE,HL;;
LD HL,10
LD A,L
LD (DE),A

MOV R7,0x0A

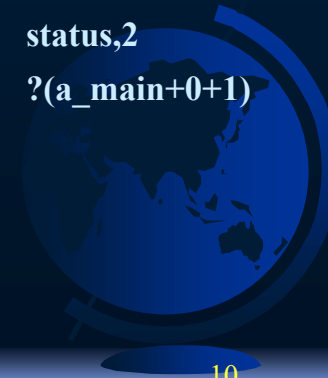
MOVLW 0Ah
MOVWF ?(a_main+4)

a++ ;

LD HL,3
ADD HL,SP
LD D,H
LD E,L
CALL CCGINT##
INC HL
CALL CCPINT##
DEC HL

INC R5
CJNE R5,#0x00,L20
INC R4
L20: NOP

INCF ?(a_main+0)
BTFS status,2
INCF ?(a_main+0+1)



Z80

I8051

PIC16c54

a = a + b; /* a i b su tipa int*/

| | | | | | |
|------|----------|------|--------|-------|-----------------|
| LD | HL,3 | MOV | A,R7 | MOVF | ?(a_main+2+1),W |
| ADD | HL,SP | RLC | A | MOVF | btemp+1 |
| PUSH | HL | SUBB | A,0xE0 | MOVF | ?(a_main+2),W |
| LD | HL,5 | MOV | R6,A | MOVWF | btemp |
| ADD | HL,SP | MOV | A,0x09 | MOVWF | btemp,w |
| CALL | CCGINT## | ADD | A,R7 | ADDWF | ?(a_main+0) |
| EX | DE,HL;; | MOV | R5,A | BTFSC | status,0 |
| LD | HL,3 | MOV | A,0x08 | INCF | ?(a_main+0+1) |
| ADD | HL,SP | ADDC | A,R6 | MOVWF | btemp+1,W |
| CALL | CCGINT## | MOV | R4,A | ADDWF | ?(a_main+0+1) |
| ADD | HL,DE | | | | |
| POP | DE | | | | |
| CALL | CCPINT## | | | | |

Z80

I8051

PIC16c54

a = b + c; / a i b su tipa int, c je tipa char */*

| | | | | | |
|------|-----------|------|--------|-------|---------------|
| LD | HL,3 | MOV | A,R7 | MOVF | ?(a_main+4),W |
| ADD | HL,SP | RLC | A | MOVF | btemp |
| PUSH | HL | SUBB | A,0xE0 | CLRF | btemp+1 |
| LD | HL,3 | MOV | R6,A | MOVF | ?(a_main+2),W |
| ADD | HL,SP | MOV | A,0x09 | MOVWF | btemp |
| CALL | CCGINT## | ADD | A,R7 | BTFSC | status,0 |
| EX | DE,HL;; | MOV | R5,A | INCF | btemp+1 |
| LD | HL,2 | MOV | A,0x08 | MOVWF | ?(a_main+2+1) |
| ADD | HL,SP | ADDC | A,R6 | ADDWF | btemp+1 |
| CALL | CCGCHAR## | MOV | R4,A | MOVWF | btemp,W |
| ADD | HL,DE | | | MOVWF | ?(a_main+0) |
| POP | DE | | | MOVWF | btemp+1,W |
| CALL | CCPINT## | | | MOVWF | ?(a_main+0+1) |

Z80

I8051

PIC16c54

```
modul (a);      /* poziv funkcije sa parametrom */
```

| | | | | | |
|------|----------|-------|---------|-------|----------------------|
| LD | HL,3 | MOV | R6,A | MOVF | ?(a_main+0+1),W |
| ADD | HL,SP | MOV | R7,0x05 | MOVWF | btemp+1 |
| CALL | CCGINT## | LCALL | _modul | MOVF | ?(a_main+0),w |
| PUSH | HL | | | MOVWF | btemp |
| CALL | modul | | | MOVWF | btemp,W |
| POP | BC | | | MOVWF | ?(0+a_modul+1) |
| | | | | MOVWF | btemp+1,W |
| | | | | MOVWF | ?(1+a_modul+1) |
| | | | | MOVWF | u28&0ffH //pov. adr. |
| | | | | MOVWF | ?(a_modul) |
| | | | | LJMP | (_modul) |

Z80

I8051

PIC16c54

while (b < 1000) {

```
LD    HL,1
ADD   HL,SP
CALL  CCGINT##
EX    DE,HL;;
LD    HL,1000
CALL  CCLT##
LD    A,H
OR    L
JP    Z,CC7
```

```
l24:
CLR   C
MOV   A,0x09
SUBB  A,#0xE8
MOV   A,0x08
XRL   A,#0x80
SUBB  A,#0x83
JNC   0x5C // while gotov
```

```
l5:
MOVWF ?(a_main+2+1),W
XORLW 80h
MOVWF btemp
MOVLW (03)^80h
SUBWF  btemp,W
MOVLW 0E8h
BTFSC 3,2
SUBWF  ?(a_main+2),W
BTFSS  status,0
GOTO   16 // izvrši while
GOTO   17 // while je gotov
```

Z80

```
LD    HL,1
ADD   HL,SP
PUSH  HL
LD    HL,3
ADD   HL,SP
CALL  CCGINT##
EX    DE,HL;;
LD    HL,5
ADD   HL,SP
CALL  CCGINT##
ADD   HL,DE
POP   DE
CALL  CCPINT##
```

I8051

b = b + a;

```
MOV   A,R5
ADD   A,0x09
MOV   0x09,A
MOV   A,R4
ADDC  A,0x08
MOV   0x08,A
SJMP  124
```

PIC16c54

I6:

```
MOVF ?(a_main+0+1),W
MOVWF btemp+1
MOVF ?(a_main+0),W
MOVWF btemp
MOVWF btemp,W
ADDWF?(a_main+2)
BTFSC status,0
INCF ?(a_mian+2+1)
MOVWF btemp+1,W
ADDWF?(a_main+2+1)
GOTO 15
```

Z80

I8051

PIC16c54

```
init modul (int x){  
    if (x > 0){
```

```
POP    BC  
POP    HL  
PUSH   HL  
PUSH   BC  
XOR    A  
OR     H  
JP     M,CC3  
OR     L  
JP     Z,CC3
```

```
SETB   C  
MOV    A,R7  
SUBB   A,#0x00  
MOV    A,R6  
XRL    A,#0x80  
SUBB   A,#0x80  
JC     _else(#0x0F)
```

```
MOVF   ?(a_modul+1+1),W  
XORLW  080h  
MOVWF  btemp  
MOVLW  (0)^80h  
SUBWF  btemp,W  
MOVLW  01h  
BTFSC  3,2  
SUBWF  ?(a_modul+1),W  
BTFSS  status,0  
GOTO   u11  
GOTO   u10
```


Z80

I8051

PIC16c54

return (x);

| | | | | |
|------|----|-----|-------|---------------------|
| POP | BC | RET | MOV | ?(a_modul+1+1),W |
| POP | HL | | MOVWF | btemp+1 |
| PUSH | HL | | MOVF | ?(a_modul+1) |
| PUSH | BC | | MOVWF | btemp |
| RET | | | GOTO | 11 //koda za povra. |



Z80

I8051

PIC16c54

```
} else {  
    return ( -x );}}
```

```
POP    BC  
POP    HL  
PUSH   HL  
PUSH   BC  
CALL   CCNEG##  
RET
```

```
CLR    C  
CLR    A  
SUBB   A,R7  
MOV    R7,A  
CLR    A  
SUBB   A,R6  
MOV    R6,A  
RET
```

```
MOV    ?(a_modul+1+1),W  
MOVWF  btemp+1  
MOVF   ?(a_modul+1)  
MOVWF  btemp  
COMP   btemp  
COMP   btemp+1  
INCF   btemp  
BTFSC  3,2  
INCF   btemp+1  
GOTO   11 //koda za povra.
```

C-- - Z80

☞ C--

☞ telnet



Nije svejedno kako se izražavamo!

- istu aktivnost možemo izraziti na različite načine
- pri obradi plaće o tome baš ne razmišljamo



Usporedba (I)

| C— naredba | prevedeni kod za μ P Z80A(4MHz) | zauzeće memorije | broj taktova | vrijeme izv. (μ s) |
|-------------------|--|---------------------|-----------------|----------------------------|
| a = a + 1; | LD HL,(a) LD DE,1 ADD HL,DE LD (a),HL | 12 x 8 (100%) | 61 | 15,25 (100%) |
| a++; | LD HL,(a) INC HL LD (a),HL DEC HL | 10 x 8 (84%) | 52 | 13 (85%) |
| ++a; | LD HL,(a) INC HL LD (a),HL | 9 x 8 (75%) | 46 | 11,5 (75%) |

Usporedba (II)

| C— naredba | prevedeni kod za μP Z80A(4MHz) | zauzeće memorije | broj taktova | vrijeme izv. (μs) |
|---|--|---------------------|-----------------|-------------------------|
| a = a + 2; | LD HL,(a) LD DE,2 ADD HL,DE LD (a),HL | 12 x 8 (100%) | 61 | 15,25 (100%) |
| ++a; ++a; | LD HL,(a) INC HL LD (a),HL LD HL,(a) INC HL LD (a),HL | 18 x 8 (150%) | 92 | 23 (150%) |
| a; #asm INC HL INC HL LD (a), HL #endasm | LD HL,(a) INC HL INC HL LD (a),HL | 10 x 8 (84%) | 52 | 13 (84%) |

