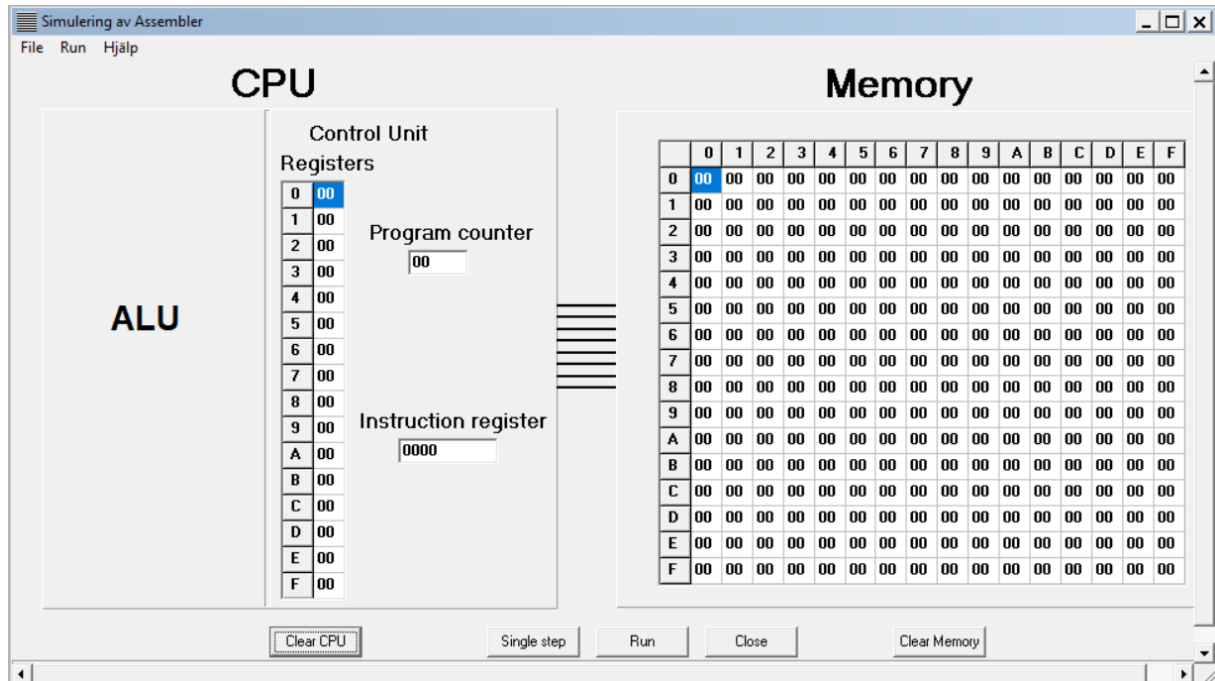


# Laborationsrapport 2

MaskinKod



Av. Khaled Hamza. DT027G, Grupp B1x

2024-09-18

# Redovisning

Koderna som användes redovisas nedan.

00 01: 1010; Ladda in data från minnescellen 01 till registren 0

02 03: 1111; Ladda in data från minnescellen 11 till registren 1

04 05: 5210; Addera talen i register 0 och 1 och spara resultat i register 2

06 07: 3212; Spara värdet från register 2 till minnescellen 12

08 09: C000; Avsluta programmet

## 2.1. Uppgift 1 Addera och subtrahera heltal

Indata och resultaten presenteras i hex form.

a)  $8+5=13$

Nedan visar hur koderna presenteras i simulatorn

00 01 02 03 04 05 06 07 08

:10:10:11:11:54:01:34:12:C0:00:00:00:00:00:00:00

:8:5:0D:00:00:00:00:00:00:00:00:00:00:00:00:00

:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Minnescellen 01 innehåller värdet 0D vilket är rätt eftersom  $0D_{16}$  är  $13_{10}$

b)  $24 - 12 = 12$

Vi måste omvandla -12 till binär form, 2-komplementmetoden.

$12 = 00001100$

$-12 = 11110011$

$18 + F4$

Nedan visar hur koderna presenteras i simulatorn

00 01 02 .....

:10:10:11:11:54:01:C0:00:00:00:00:00:00:00:00:00

:18:F4:00:00:00:00:00:00:00:00:00:00:00:00:00:00

:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Minnescellen 01 innehåller värdet  $18_{16} = 24_{10}$  och cellen 11 innehåller  $F4_{16}$  vilket är  $244_{10}$  men

eftersom op-koden 5 använder 2-komplementmetoden blir talet  $F4_{16} = -12_{10}$  och talet  $F3_{16} = -13$

osv. Resultatet blir  $0C_{16}$  vilket är  $12_{10}$ .

c)  $-125 - 30 = -155$

Nedan visar hur koderna presenteras i simulatorn

00 01 02 03 04 05 06 07

:20:83:21:E2:52:10:C0:00:00:00:00:00:00:00:00:00

00 01: 2083; spara värdet  $83_{16}$  i register 0

02 03: 21E2; spara värdet  $E2_{16}$  i register 1

04 05: 5210; addera värden som finns i register 1 och 0 och spara resultatet i register 2

06 07: C000; avsluta programmet

Svaret blev  $65_{16} = 101_{10}$ . Detta beror på att värdet (8-bit signerad integer) och det minnsta talet som kan sparas i minnet är  $80_{16}$  som är  $-128_{10}$ . Om man subtraherar ett från  $80_{16}$  blir värdet  $7F_{16}$  som är  $127_{10}$ . Värdet  $-125_{10}$  i uppgiften är  $83_{16}$  och värdet  $-30_{10}$  är  $E2_{16}$  så  $83_{16} + E2_{16} = 165_{16} = 256_{10}$  tills värdet blir mindre än  $FF_{16}$  i hex formatet eller  $225_{10}$  i osignerad decimal format.

## 2.2. Uppgift 2 Multiplicera och dividera

a)  $4 * 19 = 76$

Nedan visar hur koderna presenteras i simulatorn

00 01 02 03 04 05

:20:13:D0:02:C0:00

00 01: 2013; spara värdet  $13_{16} = 19_{16}$  i registern 0

02 03: D002; skifta bitmönstret i registern 0 2 gånger åt vänster

04 05: C000; avsluta programmet

Genom att skifta bitmönstret åt vänster multipliceras värdet med två. Värdet  $19_{10}$  skrivs som  $00010011_2$  i binär formatet. Skifter man bitmönstret en gång till vänster blir värdet  $00100110_2$  vilket är  $38_{10}$  som är dubbelt så mycket som  $19_{10}$ . För att kunna multiplicera ett värde med 4 skifter man bitmönstret 2 gånger åt vänster. För värdet  $19_{10}$  blir det  $01001100_2$  vilket är  $78_{10}$ .

b)  $24/4 = 6$

Nedan visar hur koderna presenteras i simulatorn

00 01 02 03 04 05

:20:18:E0:02:C0:00

00 01: 2018; sparavärdet  $18_{16} = 24_{10}$  i registern 0

02 03: E002; skifta bitmönstret i registern 0 2 gånger åt höger

04 05: C000; avsluta programmet

Att skifta bitmönster till vänster multipliceras värdet med 2 däremot skiftning till höger gör precis tvärtom, alltså värdet divideras med 2 i stället. Värdet  $18_{16}$  skrivs  $00011000_2$  i binär formatet. Resultatet blir  $00000110_2$  efter 2 skiftningar åt höger.

c)  $19/2=9$

Nedan visar hur koderna presenteras i simulatören

00 01 02 03 04 05

:20:13:E0:01:C0:00

00 01: 2013; spara värdet  $13_{16} = 19_{10}$  i registern 0

02 03: E001; skifta bitmönstret i registern 0 en gång åt höger

04 05: C000; avsluta programmet.

Värdet blir  $9_{10}$  eftersom  $19_{10}$  är ett udda tal och sista biten i bitmönstret är 1. När det utförs en skiftning till höger försvinner sista biten helt vilket är 1 i detta fall.

### 2.3. Uppgift 3 Använd AND, OR och XOR

a)  $1010\ 1100\ \text{AND}\ 0000\ 1111 = 0000\ 1100$

Nedan visar hur koderna presenteras i simulatören

00 01 02 03 04 05 06 07

:20:AC:21:0F:82:10:C0:00

00 01: 20AC; spara värdet  $AC_{16} = 1010\ 1100_2$  i registern 0

02 03: 210F; spara värdet  $0F_{16} = 0000\ 1111_2$  i registern 1

04 05: 8210; utför en AND operation på värden i registern 0 och 1 och spara resultatet i 2

06 07: C000; avsluta programmet

Resultatet som returnerades i registern 2 är 0C16 som är korrekt svar.

b)  $1010\ 1100\ \text{OR}\ 0000\ 1111 = 1010\ 1111$

Nedan visar hur koderna presenteras i simulatören

00 01 02 03 04 05 06 07

:20:AC:21:0F:72:10:C0:00

00 01: 20AC; spara värdet  $AC_{16} = 1010\ 1100_2$  i registern 0

02 03: 210F; spara värdet  $0F_{16} = 0000\ 1111_2$  i registern 1

04 05: 7210; utför en OR operation på värden i registern 0 och 1 och spara resultatet i 2

06 07: C000; avsluta programmet.

Det användes samma kod som på föregående uppgift men en liten ändring på operationen ändrades. Koden i minnescellen 04 ändrades från 8X till 7X. Koden retunerade rätt svar ( $AF_{16}$ ).

c)  $1010\ 1100\ \text{XOR}\ 0000\ 1111 = 1010\ 0011$

Nedan visar hur koderna presenteras i simulatorn

00 01 02 03 04 05 06 07

:20:AC:21:0F:92:10:C0:00

00 01: 20AC; spara värdet  $AC_{16} = 1010\ 1100_2$  i registern 0

02 03: 210F; spara värdet  $0F_{16} = 0000\ 1111_2$  i registern 1

04 05: 9210; utför en OR operation på värden i registern 0 och 1 och spara resultatet i 2

06 07: C000; avsluta programmet.

Programmet retunerade rätt svar ( $A3_{16}$ ).

### Slutsatser

För att förstå datorn och kunna tänka bättre som den är det nödvändigt att använda ett sådant språk på lägre nivå. Varje gång har då lägre språknivå så du kommer ha mer kontroll på datorminne och annan hårdvara. Det är möjligt att utveckla förbättrade algoritmer för att utföra till exempel aritmetiska operationer när man använder ett lågnivåspråk.