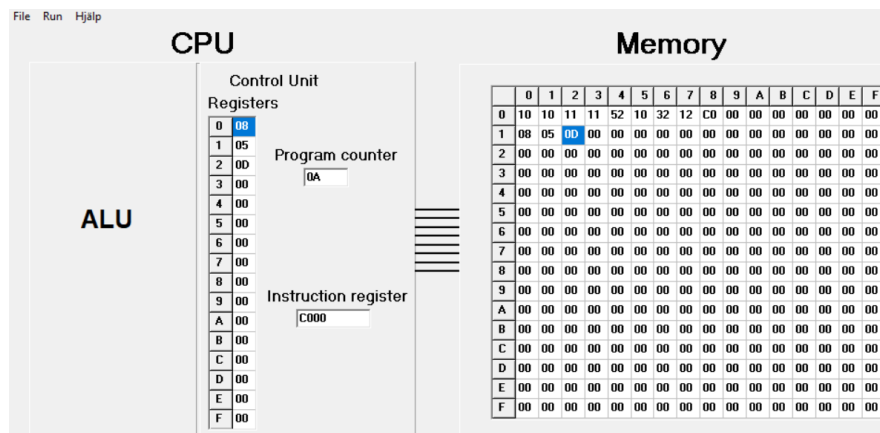


Maskinspråk



Emil Jons, DT155G, Laboration 2 Maskinspråk

Utrustning

Programmet, "simulering av assembler", som användes är en maskinkodsimulator i vilket man kan skriva in ett maskinspråksprogram och sedan simulera hur processorn hade tolkat dem angivna instruktionerna.

För att kunna skriva in rätt koder i minnet, användes denna tabell genom att undersöka vad varje operand kod hade för funktion.

Op-kod	Operand	Beskrivning
1	RXY	Ladda (LOAD) registret R med bitmönstret som finns i minnescellen med adressen XY.
2	RXY	Ladda (LOAD) registret R med bitmönstret XY.
3	RXY	Spara (STORE) bitmönstret som finns i register R i minnescellen med adressen XY.
4	oRS	Flytta (MOVE) bitmönstret i register R till register S.
5	RST	Addera (ADD) bitmönstren i registren S och T. Lägg resultatet i register R. Bitmönstren antas vara kodade enligt 2-komplementmetoden.
6	RST	Addera (ADD) bitmönstren i registren S och T. Lägg resultatet i register R. Bitmönstren antas vara kodade som ett 8 bitars flyttal omvandlat till ett hexadecimalt tal. Det binära talet anges på formatet <code>seeemmmmm</code> där s=signbit (teckenbit), eee = exponent kodad enligt excessmetoden och m = mantissa.
7	RST	Utför bitvis OR på bitmönstren i registren S och T. Lägg resultatet i register R.
8	RST	Utför bitvis AND på bitmönstren i registren S och T. Lägg resultatet i register R.
9	RST	Utför bitvis XOR på bitmönstren i registren S och T. Lägg resultatet i register R.
A	RoX	Rotera cirkulärt (ROTATE) bitmönstret i register R en bit åt höger X gånger.
B	RXY	Hoppa (JUMP) till minnescellen med adress XY om bitmönstret i register R är lika med bitmönstret i register o.
C	ooo	Stoppa (HALT) programkörningen.
D	RoX	Skifta bitmönstret i register R en bit åt vänster X gånger. Bit 1 (MSB) behåller sitt värde. Det hål som uppstår fylls med en 0:a
E	RoX	Skifta bitmönstret i register R en bit åt höger X gånger. Bit 1 (MSB) behåller sitt värde. Den bit som "faller över kanten" ersätts inte.

Redovisning av uppgifter

1.a

Programmet är placerat i minnescellerna

00 01 02 03 04 05 06 07 08 09

Indata och resultat är placerat i minnescellerna

10 11 12

Programmet redovisat radvis

1010 : ladda register 0 med data från minnescell 10

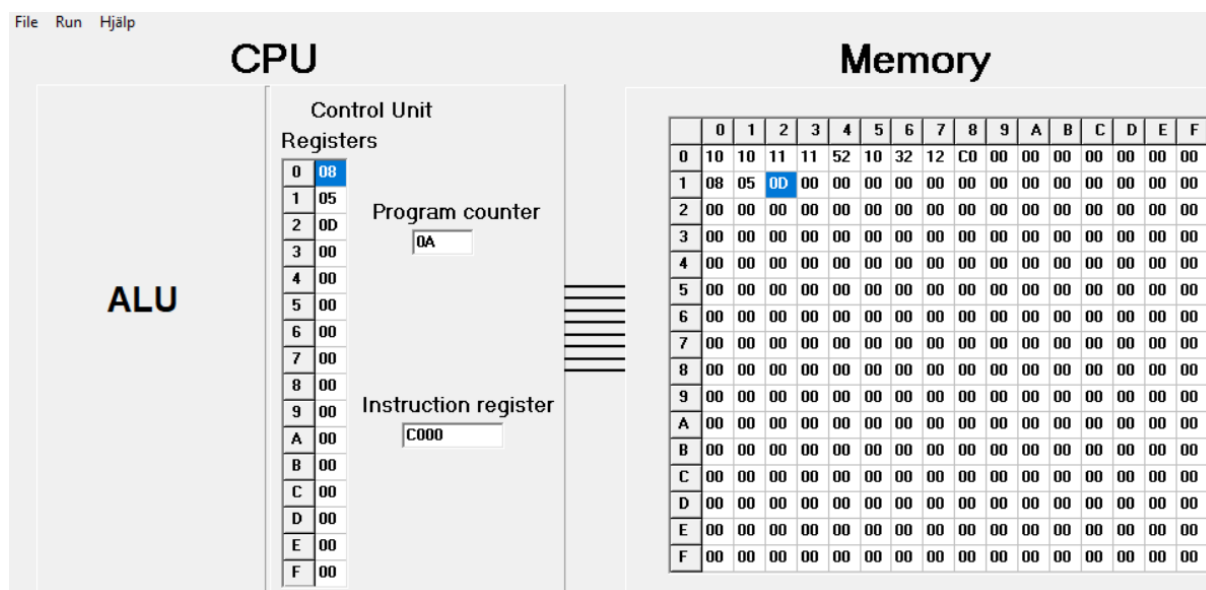
1111: ladda register 1 med data från minnescell 11

5210: addera registerna 1 och 0 sedan lägg resultatet i register 2

3212: spara register 2 i minnescell 12

C000: avsluta programmet

Programmet presenterat i form av en skärmbild samt i form av text från den sparade filen



:10:10:11:11:52:10:32:12:C0:00:00:00:00:00:00:00

:08:05:0D: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

:00:00:00:00:00:00:00 ...

Övriga kommentarer

Programmet utför operationen $8 + 5$ vilket resulterar till 13_{10} eller $0D_{16}$

1.b

Programmet är placerat i minnescellerna

00 01 02 03 04 05 06 07 08 09

Indata och resultat är placerat i minnescellerna

10 11 12

Programmet redovisat radvis

1010 : ladda register 0 med data från minnescell 10

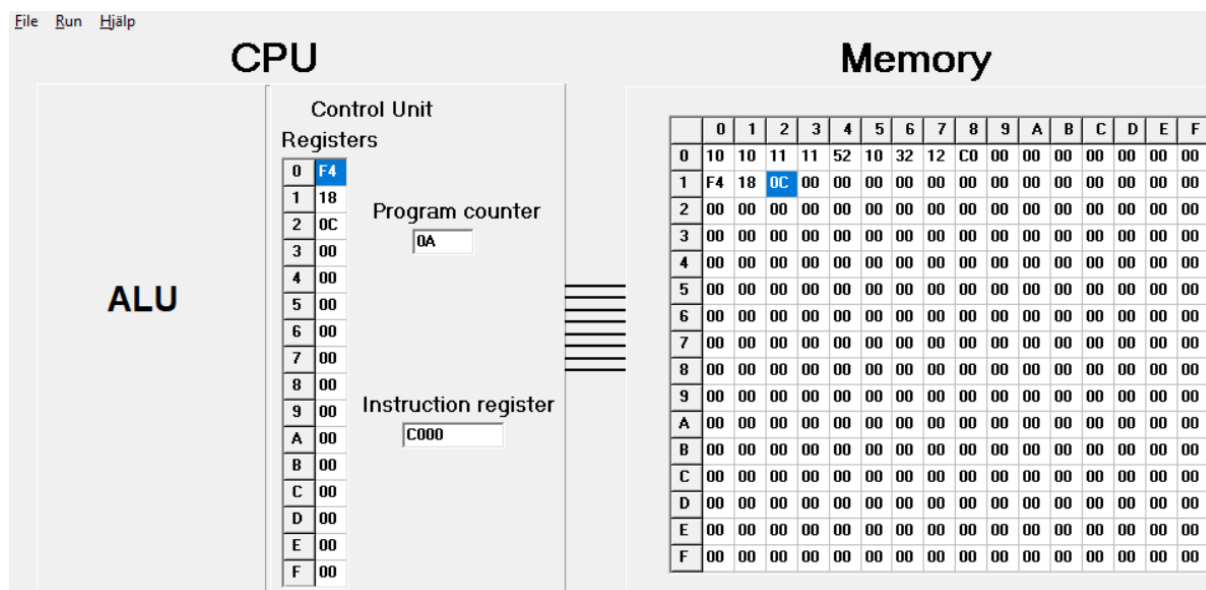
1111: ladda register 1 med data från minnescell 11

5210: addera registerna 1 och 0 sedan lägg resultatet i register 2

3212: spara register 2 i minnescell 12

C000: avsluta programmet

Programmet presenterat i form av en skärmbild samt i form av text från den sparade filen



:10:10:11:11:52:10:32:12:C0:00:00:00:00:00:00:00

:F4:18:0C:00:00:00:00:00:00:00:00:00:00:00:00

:00:00:00:00:00 ...

Övriga kommentarer

Programmet utför operationen 24 - 12 vilket resulterar till 12_{10} eller $0C_{16}$

1.c

Programmet är placerat i minnescellerna

00 01 02 03 04 05 06 07 08 09

Indata och resultat är placerat i minnescellerna

10 11 12

Programmet redovisat radvis

1010 : ladda register 0 med data från minnescell 10

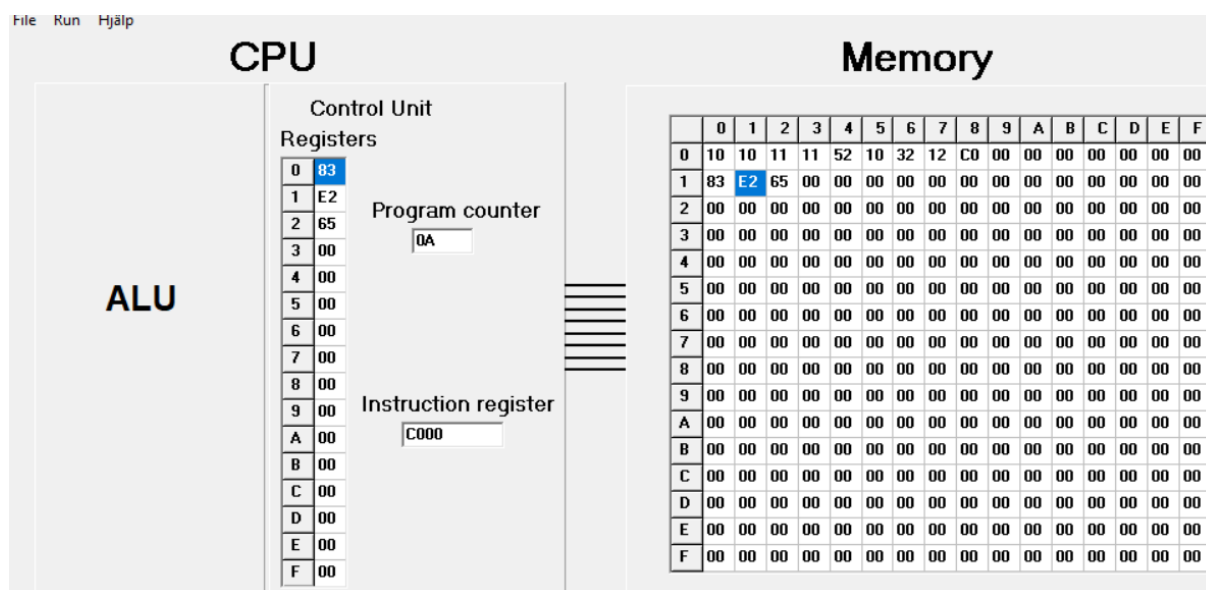
1111: ladda register 1 med data från minnescell 11

5210: addera registerna 1 och 0 sedan lägg resultatet i register 2

3212: spara register 2 i minnescell 12

C000: avsluta programmet

Programmet presenterat i form av en skärmbild samt i form av text från den sparade filen



```
:10:10:11:11:52:10:32:12:C0:00:00:00:00:00:00:00
:83:E2:65:00:00:00:00:00:00:00:00:00:00:00:00
:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
:00:00:00:00 ...
```

Övriga kommentarer

Programmet utför operationen $-125 - 30$ vilket resulterar till -155 eller $FF65_{16}$. På grund av att simulatören bara kan hantera tal som kan lagras i en byte så skrivs i stället talet 65_{16} ut. Problemet kallas ett overflow error, vilket resulterar att svaret blir fel. I detta fall blev svaret 101_{10} istället för -155_{10} .

2.a

Programmet är placerat i minnescellerna

00 01 02 03 04 05 06 07

Indata och resultat är placerat i minnescellerna

10 11

Programmet redovisat radvis

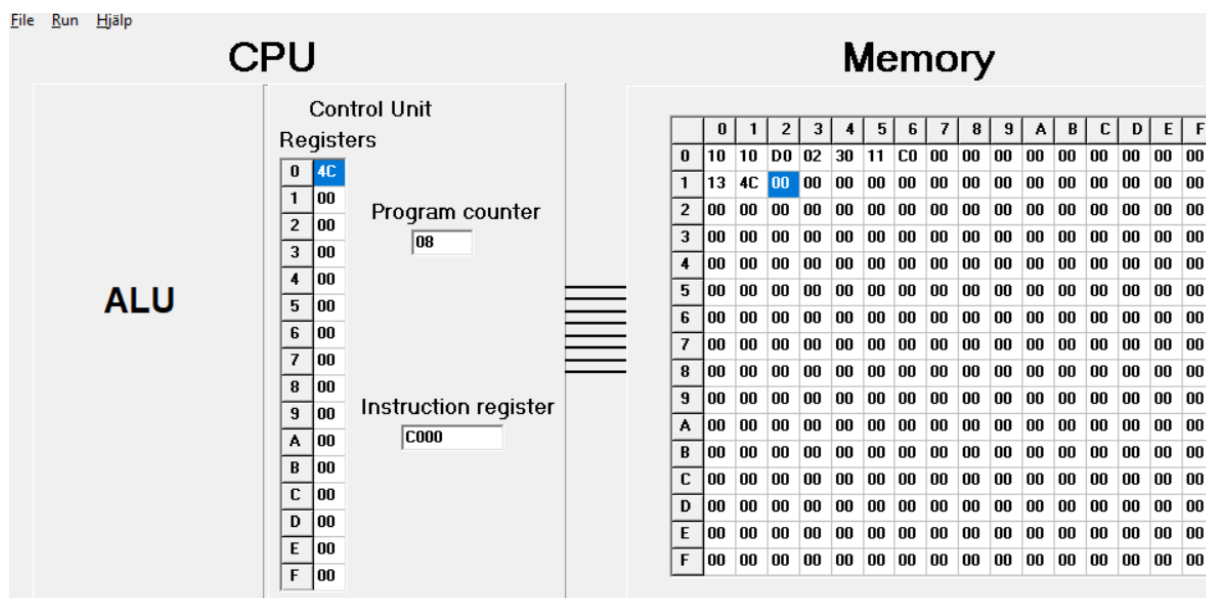
1010 : ladda register 0 med data från minnescell 10

D002: skiftar register 0 två steg åt vänster

3011: spara register 0 i minnescell 11

C000: avsluta programmet

Programmet presenterat i form av en skärmbild samt i form av text från den sparade filen



:10:10:D0:02:30:11:C0:00:00:00:00:00:00:00:00:00

:13:4C: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

:00:00:00:00:00:00:....

Övriga kommentarer

Programmet utför operationen 4×19 vilket resulterar till 76_{10} eller $4C_{16}$. Den utför detta med op-koden D som skiftar bitmönstret i registret en bit åt vänster vilket resulterar i att den multiplicerar datan med två. Den skiftar två gånger, alltså multiplicerar med 4.

2.b

Programmet är placerat i minnescellerna

00 01 02 03 04 05 06 07

Indata och resultat är placerat i minnescellerna

10 11

Programmet redovisat radvis

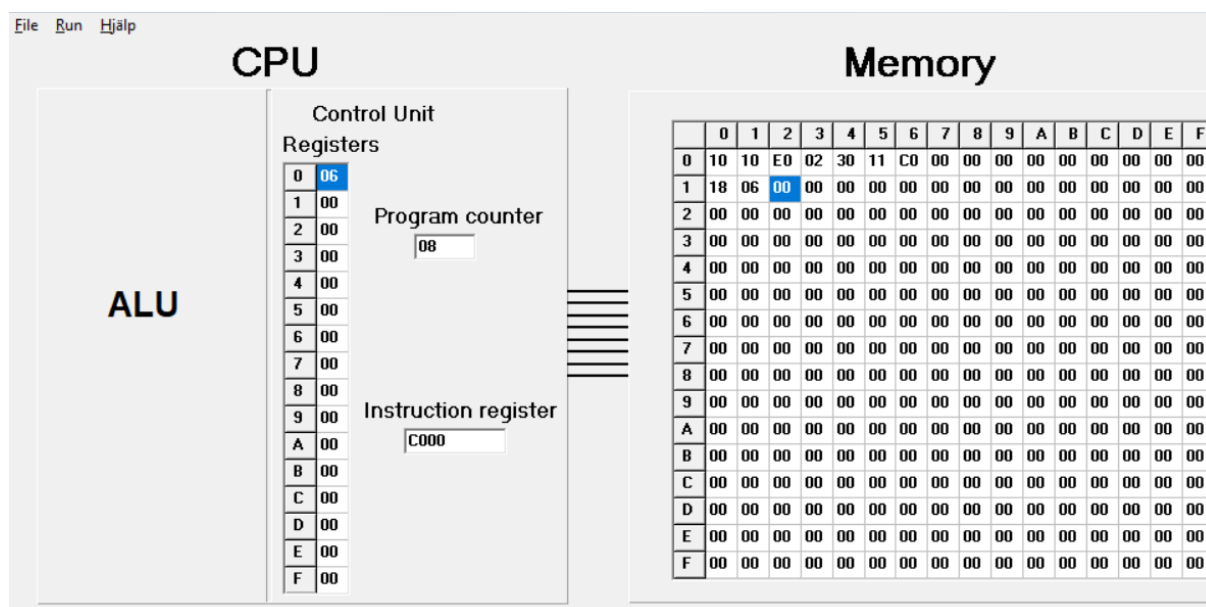
1010 : ladda register 0 med data från minnescell 10

E002: skiftar register 0 två steg åt höger

3011: spara register 0 i minnescell 11

C000: avsluta programmet

Programmet presenterat i form av en skärmbild samt i form av text från den sparade filen



:10:10:E0:02:30:11:C0:00:00:00:00:00:00:00:00:00

:18:06:00:00:00:00:00:00:00:00:00:00:00:00:00:00

:00:00:00:00:00:00:00 ...

Övriga kommentarer

Programmet utför operationen 24/4 vilket resulterar till 6_{10} och 6_{16} . Den utför detta med op-koden E som skiftar bitmönstret i registret en bit åt höger vilket resulterar i att den dividerar datan med två. Den skiftar två gånger, alltså dividerar med 4.

2.c

Programmet är placerat i minnescellerna

00 01 02 03 04 05 06 07

Indata och resultat är placerat i minnescellerna

10 11

Programmet redovisat radvis

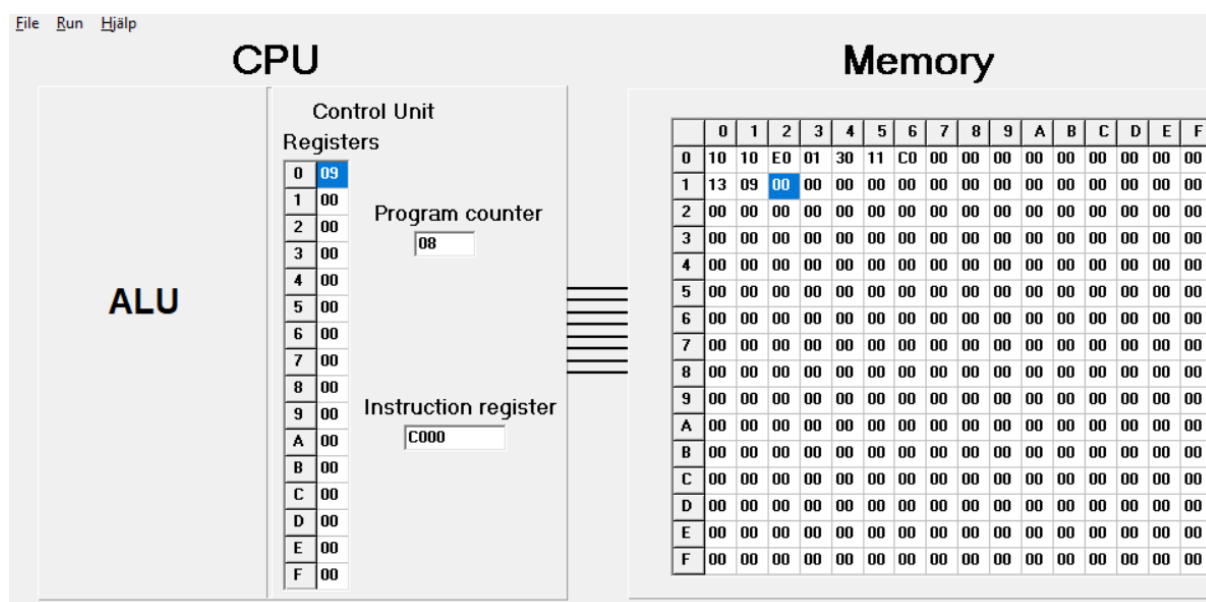
1010 : ladda register 0 med data från minnescell 10

E001: skiftar register 0 två steg åt höger

3011: spara register 0 i minnescell 11

C000: avsluta programmet

Programmet presenterat i form av en skärmbild samt i form av text från den sparade filen



:10:10:E0:01:30:11:C0:00:00:00:00:00:00:00:00:00

:13:09:00:00:00:00:00:00:00:00:00:00:00:00:00:00

:00:00:00:00:00:00:00 ...

Övriga kommentarer

Programmet utför operationen 19/2 vilket resulterar till 9_{10} och 9_{16} när man bortser från resten. Operationen hanterar endast heltalsdivision, så 19/2 blir 9 i stället för 9,5 då resten ignoreras. Den utför detta med op-koden E som skiftar bitmönstret i registret en bit åt höger vilket resulterar i att den dividerar datan med två.

3.a

Programmet är placerat i minnescellerna

00 01 02 03 04 05 06 07 08 09

Indata och resultat är placerat i minnescellerna

10 11 12

Programmet redovisat radvis

1010 : ladda register 0 med data från minnescell 10

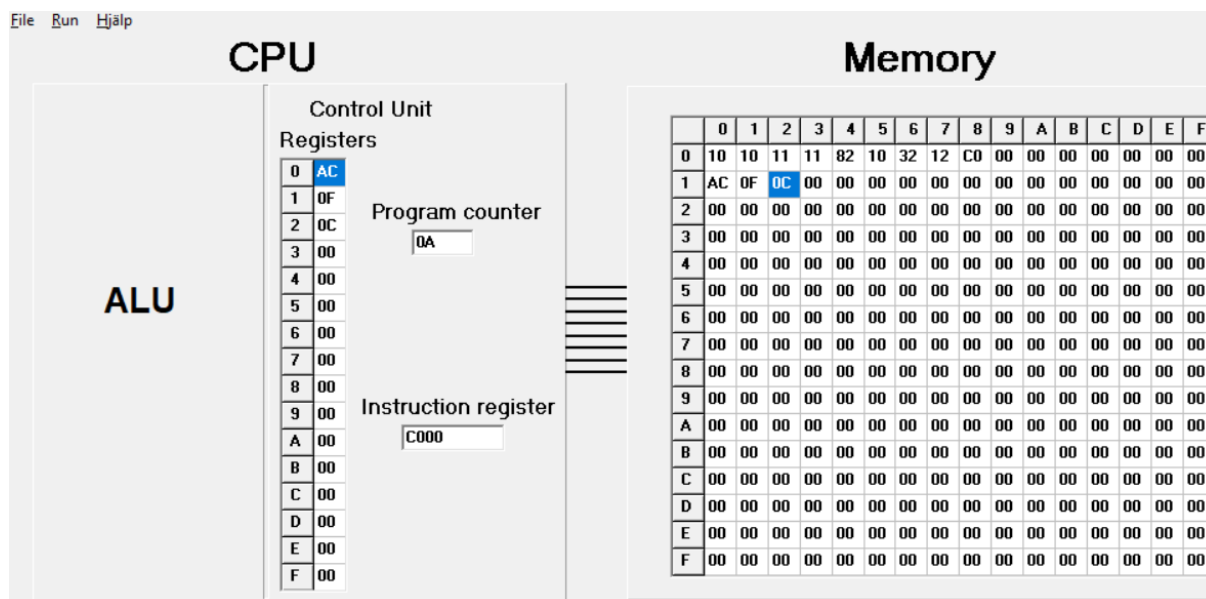
1111: ladda register 1 med data från minnescell 11

8210: utför den logiska operationen AND på register 0 och 1, för att sedan lägga resultatet i register 2

3212: spara register 2 i minnescell 12

C000: avsluta programmet

Programmet presenterat i form av en skärmbild samt i form av text från den sparade filen



:10:10:11:11:82:10:32:12:C0:00:00:00:00:00:00:00

:AC:0F:0C:00:00:00:00:00:00:00:00:00:00:00:00

:00:00:00:00:00:00 ...

Övriga kommentarer

Programmet utför den logiska operationen AND på talen AC_{16} och $0F_{16}$ eller 10101100_2 och 00001111_2 . Resultatet blir $0C_{16}$ eller 00001100_2

3.b

Programmet är placerat i minnescellerna

00 01 02 03 04 05 06 07 08 09

Indata och resultat är placerat i minnescellerna

10 11 12

Programmet redovisat radvis

1010 : ladda register 0 med data från minnescell 10

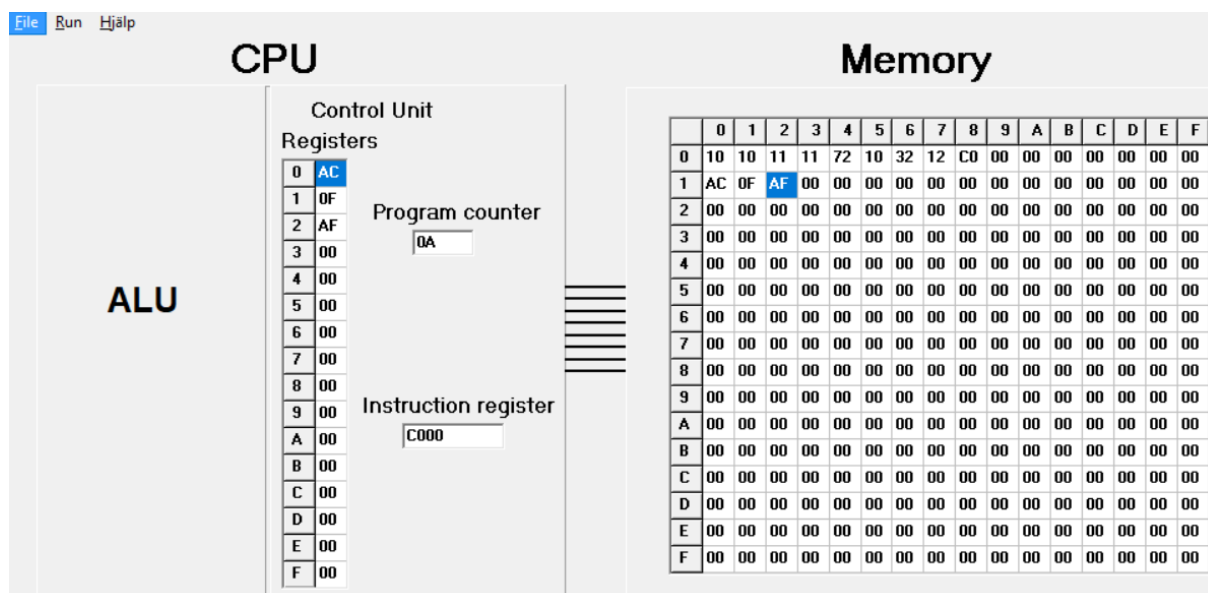
1111: ladda register 1 med data från minnescell 11

7210: utför den logiska operationen OR på register 0 och 1, för att sedan lägga resultatet i register 2

3212: spara register 2 i minnescell 12

C000: avsluta programmet

Programmet presenterat i form av en skärmbild samt i form av text från den sparade filen



:10:10:11:11:72:10:32:12:C0:00:00:00:00:00:00:00

:AC:0F:AF:00:00:00:00:00:00:00:00:00:00:00:00

:00:00:00:00:00:00 ...

Övriga kommentarer

Programmet utför den logiska operationen OR på talen AC_{16} och $0F_{16}$ eller 10101100_2 och 00001111_2 . Resultatet blir AF_{16} eller 10101111_2

3.c

Programmet är placerat i minnescellerna

00 01 02 03 04 05 06 07 08 09

Indata och resultat är placerat i minnescellerna

10 11 12

Programmet redovisat radvis

1010 : ladda register 0 med data från minnescell 10

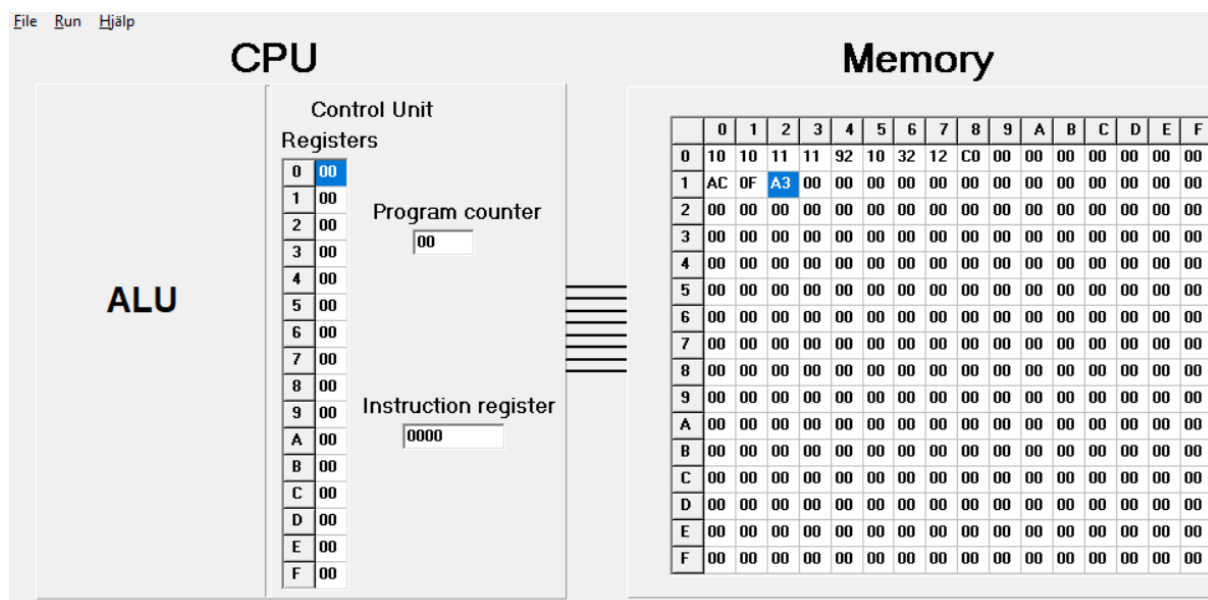
1111: ladda register 1 med data från minnescell 11

9210: utför den logiska operationen XOR på register 0 och 1, för att sedan lägga resultatet i register 2

3212: spara register 2 i minnescell 12

C000: avsluta programmet

Programmet presenterat i form av en skärmbild samt i form av text från den sparade filen



:10:10:11:11:92:10:32:12:C0:00:00:00:00:00:00:00

:AC:0F:A3:00:00:00:00:00:00:00:00:00:00:00:00

:00:00:00:00:00:00 ...

Övriga kommentarer

Programmet utför den logiska operationen XOR på talen AC_{16} och $0F_{16}$ eller 10101100_2 och 00001111_2 . Resultatet blir $A3_{16}$ eller 10100011_2

Slutsatser

Alla operationer fick de väntade svaren förutom uppgift 1.c. På grund av att minnescellerna är så pass små, begränsar det ganska mycket hur stora uträkningar man kan utföra i datorn. Man kan bara skriva tal mellan 127 och -128, när man endast har 8 bitar i 2 komplement att arbeta med. I uppgiften 1.c visas det hur talet blir för stort för att räknas ut korrekt, vilket resulterar i fel svar eller även kallat ett överflödes fel. Utöver det felet, gick det bra att genomföra att uppgiften och resultaten togs fram utan övriga problem.