

Övningsuppgifter

Övningsuppgifterna i kapitel F avser FLIS-processorn, vars instruktioner och motsvarande koder definieras i "INSTRUKTIONSLISTA FÖR FLISP".

F.2 Ett antal på varandra följande minnesord har följande hexadecimala innehåll:

a) F1, 20, E1, 21, 96, 0F, A9, 25, 06, 97, 10.

Den första byten ($F1_{16}$) är en operationskod för FLIS-processorn.

Översätt sekvensen till assemblerspråk, dvs disassemblera sekvensen.

F.3 Ge en sekvens av LDA- och STA-instruktioner (dvs ett programavsnitt) som flyttar innehållet på adresserna $1A_{16}$ - $1C_{16}$ till adresserna $2A_{16}$ - $2C_{16}$.

F.5 Manuell översättning från assemblerspråk till maskinkod kallas ofta "**handassemblering**".

a) Handassemblera följande instruktionssekvens. Första instruktionen skall placeras på adress 80_{16} . Hur många minnespositioner upptar instruktionssekvensen?

```
LDA    $10
ANDA   #$F
ORA     $11
EORA   #$44
ANDA   #$EE
COMA
STA    $10
```

b) Antag att talet 10110110_2 finns på adress 10_{16} . Ange i hexadecimal form det tal som kommer att finnas i denna minnesposition efter exekvering av instruktionssekvensen ovan, om adress 11_{16} innehåller 01_{16} före exekveringen.

- F.6** a) Skriv en instruktionssekvens som inverterar bitarna b_0 och b_7 i A-registret och ettställer bitarna b_2 och b_5 samt nollställer bit b_4 .
- b) Handassemblera instruktionssekvensen. Första instruktionen skall placeras på adress 20_{16} .

F.7 I minnet finns data enligt figuren till höger.

Skriv en instruktionssekvens som ökar innehållet på adressen 80_{16} med 5 och minskar innehållet på adressen 81_{16} med 7.

Vidare skall instruktionssekvensen addera innehållen på minnesadresserna 82_{16} och 83_{16} samt placera summan på adressen 84_{16} .

Översätt instruktionssekvensen till maskinkod (hexadecimal form) och placera den i minnet med början på adressen 20_{16} .

Adr	
80_{16}	56_{16}
81_{16}	23_{16}
82_{16}	05_{16}
83_{16}	16_{16}
84_{16}	$B7_{16}$

- F.11** a) "Jump"-instruktionen JMP $\$50$ är placerad med början på adress 37_{16} .
Vad är dess maskinkod?
- b) "Branch"-instruktionen BRA $\$50$ är placerad med början på adress 37_{16} .
Vad är dess maskinkod?
- F.12** a) "Jump"-instruktionen JMP $\$05$ är placerad med början på adress 20_{16} .
Vad är dess maskinkod?
- b) "Branch"-instruktionen BRA $\$05$ är placerad med början på adress 20_{16} .
Vad är dess maskinkod?

F.14 I följande instruktionssekvens genereras en puls i position 0 på utport FB_{16} .

Läge	Operation	Operand	Kommentar
START	CLR	\$FB	
	LDA	#25	
SCOUNT	DECA		
	BNE	SCOUNT	
	LDA	##00000001	
	STA	\$FB	
	LDA	#100	
PCOUNT	DECA		
	BNE	PCOUNT	
	CLR	\$FB	

- a)** Rita en maskinberoende flödesplan för sekvensen.
- b)** Antag att FLIS-processorn har klockfrekvensen 1 MHz och att sekvensen startas vid tidpunkten $t = 0$. När i tiden genereras pulsen? Hur lång är den?
- c)** Kommentera instruktionerna på lämpligt sätt. Kommentarererna skall helst vara maskinberoende.
- d)** Översätt instruktionssekvensen till maskinkod. Antag att START skall läggas på adressen 20_{16} . Måste den maskinkod du nu producerat alltid ligga på dessa adresser?

F.15 Vad blir den *effektiva adressen* i följande instruktioner om innehållet i X-registret = 20_{16} . För instruktioner som arbetar med data är *effektiva adressen* adressen till data. Översätt instruktionerna till maskinkod.

- a) LDA 0,X
- b) LDA 16,X
- c) LDA -16,X
- d) LDA ,X+
- e) LDA ,-X

F.16 Skriv en instruktionssekvens som omvandlar ett 4-bitars binärt tal till Graykod. Det binära talet skall kontinuerligt läsas från bit b_0 - b_3 på inport FB_{16} . Bit b_4 - b_7 :s värden är ej kända och kan variera mellan läsningarna. Graykoden skall matas ut på bit b_0 - b_3 på utport FB_{16} med bit b_4 - b_7 nollställda. Graykoden för siffrorna 0_{16} - F_{16} finns lagrade i bit b_0 - b_3 i adress 30_{16} - $3F_{16}$, med bit b_4 - b_7 nollställda.

F.19 En tabell med 5 olika 8-bitars tal utan tecken lagras med början på adress 18_{16} i minnet. Skriv en instruktionssekvens som letar upp det största talet. Talets värde och läge skall anges i minnespositionerna VALUE resp PLACE, vilka anses fördefinierade.

F.20 Ge en sekvens av instruktioner som definierar en stack med början på adress $F0_{16}$ och placerar operanderna 06_{16} , 35_{16} och 28_{16} på stacken. Vad är stackpekarens innehåll när sekvensen genomlöpts av processorn?

F.23 Skriv ett programavsnitt som läser ett tal från inport FB_{16} .

Om talet är 00_{16} skall en subrutin FALL1 anropas. Talet skall då ligga i A-registret för att subrutinen skall kunna använda det för bearbetning.

Om talet är 18_{16} skall på motsvarande sätt en subrutin FALL2 anropas.

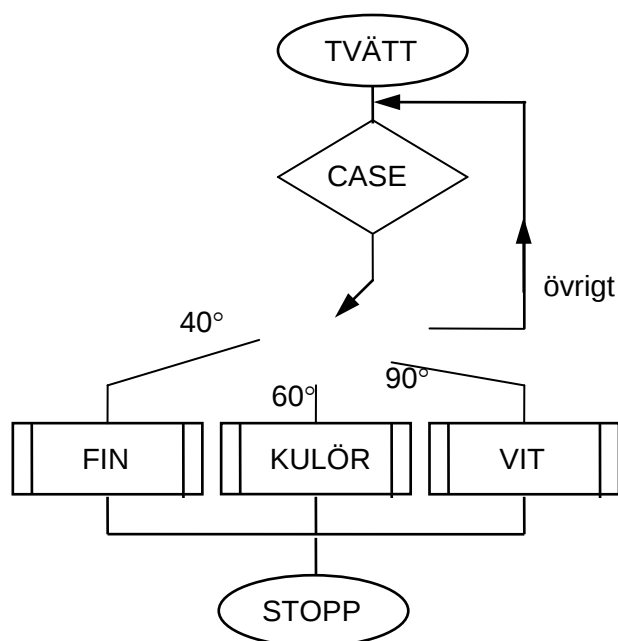
Om talet inte är lika med något dessa värden skall en subrutin FALL3 anropas.

Rita också en flödesplan.

F.25 En enkel datorstyrd tvättmaskin har tre olika tvättprogram

- 40°C Fintvätt / 60°C Kulör tvätt / 90°C Vittvätt.

När tvättmaskinen startats agerar den enligt flödesplanen nedan.



a) Rita om flödesplanen så att fyrvalssituationen realiseras som en följd av tvåval (**if-then-else**)!

b) Användaren kan trycka på tre olika knappar som svarar mot de olika tvättprogrammen. Den valda tvättemperaturen är åtkomlig för processorn via inporten FB_{16} enligt figuren nedan.

Observera att vissa värden på $x_2x_1x_0$ representerar ogiltiga val (t.ex. om både x_1 och x_0 är ett). Dessa fall motsvaras av alternativet "övrigt" i flödesplanen.

De olika tvättprogrammen är subrutiner med början på de symboliska adresserna "Fin", "Kulör" respektive "Vit". Omsätt den nya flödesplanen i ett program skrivet med FLISP instruktioner.

Ledning:

Varje tvättemperatur motsvaras av ett unikt bitmönster i de tre minst signifikanta positionerna i data från inport FB_{16} , 60°C Kulör tvätt motsvaras t.ex. av:

$x_2x_1x_0 = 010$.

