



## Laborationshäfte i Digital- och datorteknik

### LEU431

Detta häfte innehåller anvisningar om förberedelser och genomförande av fyra laborationer.

Inför varje laboration skall häftet studeras. Uppgifter i detta häfte och ur Arbetsbok för DigiFlex skall vara utförda och uppvisade inför varje laborationstillfälle. Eventuella lösningar på elektronisk form skall vara tillgängliga på ditt "CTH-konto" vid laborationstillfället. ***Bristfälliga förberedelser kan medföra avvisning från bokad laborationstid.***

Den utrustning som används vid laborationen skall ha studerats ***före*** laborationstillfället. Information om laborationsutrustningen finns på webben via kursens hemsida under länken "Beskrivning av labmoduler".

Laboration	Prov		Anmärkning		Godkännande på laboration	
	Datum	Signatur	1	2	Datum	Laborationshandledares underskrift
1						
2						
3						
4						

Institutionen för data och informationsteknik, Chalmers tekniska högskola, 2012, Version 120703

Laborant: (Ifylls med bläck!)

\_\_\_\_\_

Personnummer                                  Namn (textat)                                  Datum

Godkännande - laborationsserie:

\_\_\_\_\_

Handledare    Datum

Se till att få ovanstående ifyllt..... Det är **ditt kvitto** på att du är godkänd på laborationerna.



## Laboration nr 1 behandlar:

### *Digital teknik, Kombinatoriska nät, Adderare, Väljare och ALU.*

Följande uppgifter ur "Arbetsbok för DigiFlex" skall vara utförda och uppvisade för en handledare innan laborationen utförs.

<b>Uppg</b>	<b>8</b>	<b>11, 13</b>	<b>17</b>	<b>19-24</b>	<b>25-29</b>	<b>31</b>	<b>33-35</b>	<b>45</b>
(Signatur)								

Hemuppgiften i laborationsuppgiften till höger skall vara utförd och uppvisad för en handledare innan laborationen utförs.

<b>Uppg</b>	<b>6</b>
(Signatur)	

Följande laborationsuppgifter visas för en handledare innan de kopplas ner. Uppgift X får du under laborationstillfället.

<b>Uppg</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>X</b>
(Signatur)								

## Laborationsuppgift 1

### *Mätning av logiknivåer.*

Som första praktiska uppgift i laboratoriet skall vi undersöka vilka spänningar relativt jord, som motsvarar logikvärdena "0" och "1" i digitalmaskinen.

Eftersom alla spänningar i digitalmaskinen mäts relativt jord använder vi i fortsättningen endast ordet "spänning" i stället för "spänning relativt jord". Vi väljer att mäta spänningarna på en AND-grunds ingångar och utgång med en voltmeter.

**Läs om modulen:**  
2-ingångars AND

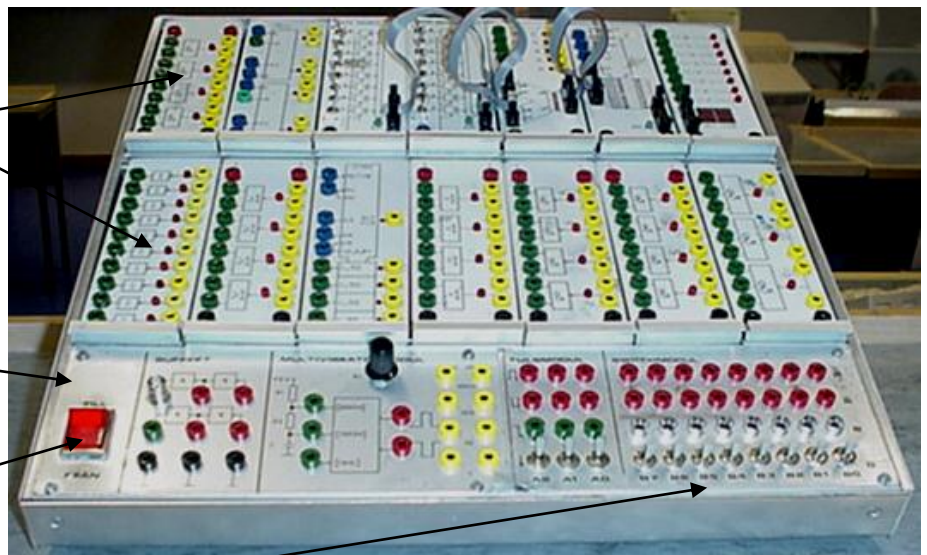
Fråga en assistent om du vill ha hjälp med att använda en voltmeter.

Två rader med plats för utbytbara logikmoduler

Nedersta raden är en manöverpanel och dessa moduler kan inte bytas ut

Nätströmbrytare

Switchmodul

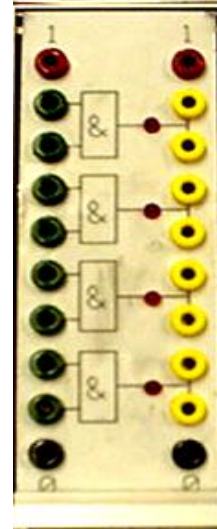


Digitalmaskinen

SNÄLLA NI.....SKRIV INTE PÅ LAB-MODULERNA o "peka ej med pennor" på dem.

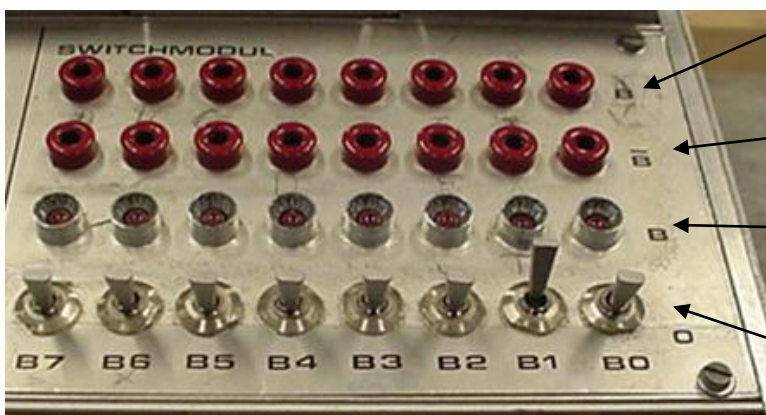
Studera AND-modulen som innehåller fyra 2-ingångars AND-grindar.

Grindarna har gröna bananhylsor på ingångarna. De gula utgångarna är dubblerade för att man enklare skall kunna koppla utsignalen vidare till ingångar på andra grindar. Varje grind har en lysdiod på utgången som indikerar om utgången är en etta (tänd diod) eller nolla (släckt diod). Överst hittas röda bananhylsor för +5V och nederst svarta för 0V. Dessa kan man använda till att koppla en konstant nolla eller etta till ingångar på kretsar.



Slå av nätströmbrytaren på digitalmaskinen.

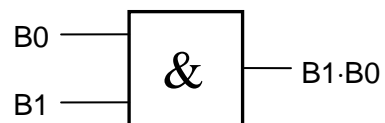
Insignalerna till grinden tas från switchmodulen: Koppla en labsladd från den övre banankontakten på switch B0 (B-raden på switchmodulen) till ena ingången på en av AND-grindarna.



Switchmodul

- Bananhylsor för ut signaler (B) från strömbrytarna
- Bananhylsor för inverterade ut signaler (B') från strömbrytarna
- Lysdioder som visar inställt logikvärde på strömbrytarna
- Strömbrytare för inställning av logiknivåer

Koppla en annan labsladd från den övre banankontakten på switch B1 (B-raden på switchmodulen) till den andra grindingången. Se figuren till höger.



Slå på nätströmbrytaren på digitalmaskinen.

Ställ in logikvärdena enligt tabellen nedan med switcharna B0 och B1. Mät spänningen på den använda grindens ingångar och utgång med en voltmeter och fyll i tabellen. Observera också utgångens logikvärde.

Insignal B1		Insignal B0		Utsignal B1·B0	
Logikvärde	Spänning [V]	Logikvärde	Spänning [V]	Logikvärde	Spänning [V]
0		0			
0		1			
1		0			
1		1			

Från mätvärdena i tabellen drar vi slutsatsen att:

Logikvärdet "0" motsvarar spänningen \_\_\_\_ V. Logikvärdet "1" motsvarar spänningen \_\_\_\_ V.

De uppmätta värdena är typiska för den sorts mikroelektronik, HCMOS, som används i digitalmaskinen.

Vi observerar också att de AND-grindar som inte har någon yttre signal ansluten till sina ingångar via en labsladd har logikvärdet ..... på utgången.

Mätning av spänning på ingångarna till dessa grindar visar spänningen \_\_\_\_ V, dvs. logikvärdet ....., vilket förklarar utsignalvärdet. *Det är samma logikvärde på (nästan) samtliga ingångar, som saknar yttre anslutning via en labsladd.*

Glöm inte att nätströmbrytaren på digitalmaskinen skall vara avstängd vid kopplingsarbetet!

## Laborationsuppgift 2

### Undersökning av XOR-grind.

Slå av nätströmbrytaren på digitalmaskinen.

Koppla upp **Uppgift 8** från arbetsboken.

Slå på nätströmbrytaren på digitalmaskinen.

Använd switch B0 på switchmodulen som styrsignal s och switch B1 som insignal x.

Fyll i funktionstabellen.

s	x	u
0	0	
0	1	
1	0	
1	1	

**Läs om modulerna:**  
XOR

## Laborationsuppgift 3

### NAND-logik.

Slå av nätströmbrytaren på digitalmaskinen.

Koppla upp **Uppgift 17** där du använder NAND/NAND-logik från arbetsboken. Använd switch B0-B3 på switchmodulen som insignaler för x, y, z och w.

**Läs om modulerna:**  
NAND  
INVERTERARE

Slå på nätströmbrytaren på digitalmaskinen och kontrollera utsignalen  $f_a$  för följande värden av  $f_a(xyzw)$ :

$f_a(0000) =$  \_\_\_\_\_,  $f_a(0110) =$  \_\_\_\_\_,

$f_a(0111) =$  \_\_\_\_\_,  $f_a(1011) =$  \_\_\_\_\_

## Laborationsuppgift 4

### Konstruktion av kodomvandlare

Slå av nätströmbrytaren på digitalmaskinen.

Koppla upp **Uppgift 23** från arbetsboken. Använd switcharna B7 till B4 på switchmodulen för insignalerna x, y, z och w enligt figuren nedan.

#### Läs om modulerna:

Vanliga logiska grindar  
(AND, NAND, NOR etc.)

Displaymodul  
(Chokladhjul)

B7	B6	B5	B4	B3	B2	B1	B0
x	y	z	w				

Koppla kodomvandlarens utsignaler till displaymodulen så att det högra sifferfönstret används, dvs utsignal a till insignal nr 3, utsignal b till insignal nr 2, utsignal c till insignal 1 och utsignal d till insignal 0 på displaymodulen.

När du kopplat upp hela kodomvandlaren slår du på nätströmbrytaren på digitalmaskinen och testar kodomvandlaren för hela funktionstabellen i figur 2 i arbetsboken. Demonstrera sedan kopplingen för en handledare.

Diskutera med din handledare om du skall ansluta en modell av ett riktigt chokladhjul. Kanske måste du då använda INVERTERARE på några av insignalerna x, y, z beroende på om du använt B eller B'-utgångarna från switch-modulen. Fråga en handledare om hjälp.

## Laborationsuppgift 5

### Väljare och dess användning.

Studera manualbladen för de moduler som behövs för att koppla upp **Uppgift 34** från arbetsboken.

Slå av nätströmbrytaren på digitalmaskinen.

Koppla upp **Uppgift 34** från arbetsboken.

Använd switcharna B7, B6 och B0 på switchmodulen för signalerna  $X_0$ ,  $X_1$  och  $S_0$  enligt figuren nedan.

B7	B6	B5	B4	B3	B2	B1	B0
$X_0$	$X_1$						$S_0$

Slå på nätströmbrytaren på digitalmaskinen.

Verifiera att kopplingen fungerar som tänkt.

**Laborationsuppgift 6****Konstruktion av 2-bitars adderare**

Slå av nätströmbrytaren på digitalmaskinen.

Studera **Uppgift 24** i arbetsboken. 2-bitarsadderaren skall bestå av 2 st ihopkopplade heladderare, en för bitposition 0 och en för bitposition 1. Rita av din adderare i figurboxen nedan. Observera att  $c_{in} = c_0$  skall finnas med som insignal.

Koppla upp adderaren där du ansluter insignalerna  $x_1$  och  $x_0$  till B5 och B4 på switch-modulen. Vidare väljer du B1 och B0 för insignalerna  $y_1$  och  $y_0$ . Slutligen väljer du B7 som  $c_0$ .

*När du kopplar upp ett så pass stort nät kan en sladd lätt hamna fel. Det kan då vara ett bra arbetssätt att först koppla upp en del av konstruktionen. Exempelvis kopplar du upp nätet som utgör en 1-bits adderare för bitposition 0 ( $x_0 + y_0 + c_0 = c_1, s_0$ ) och testar detta först. Sedan kan utöka kopplingen till en 2-bitars adderare.*

**Läs om modulerna:**

Vanliga logiska grindar  
(AND, NAND, NOR etc.)

$$\begin{array}{r} c_2 \quad c_1 \quad c_0 \\ \quad x_1 \quad x_0 \\ + \quad y_1 \quad y_0 \\ \hline = \quad s_1 \quad s_0 \end{array}$$

**Hemuppgift 6**  
(Utförs före labtilfället!)

2-bitars adderare

Slå på nätströmbrytaren på digitalmaskinen och testa kopplingen för de värden som är givna överst på sid 35 i arbetsboken.

## Beskrivning av ALU-, DATA SOURCE- och DISPLAY- moduler

Figuren till höger visar frontpanelen för en 8-bitars ALU-modul som ingår i labsystemet.

Modulen har två 8-bitars dataingångar D och E samt en ingång för carry-in. Den har också fyra ingångar,  $f_0$ ,  $f_1$ ,  $f_2$  och  $f_3$  som bestämmer dess funktion.

Förutom datautgången U med 8 bitar har den flaggutgångarna N, Z, V och C.

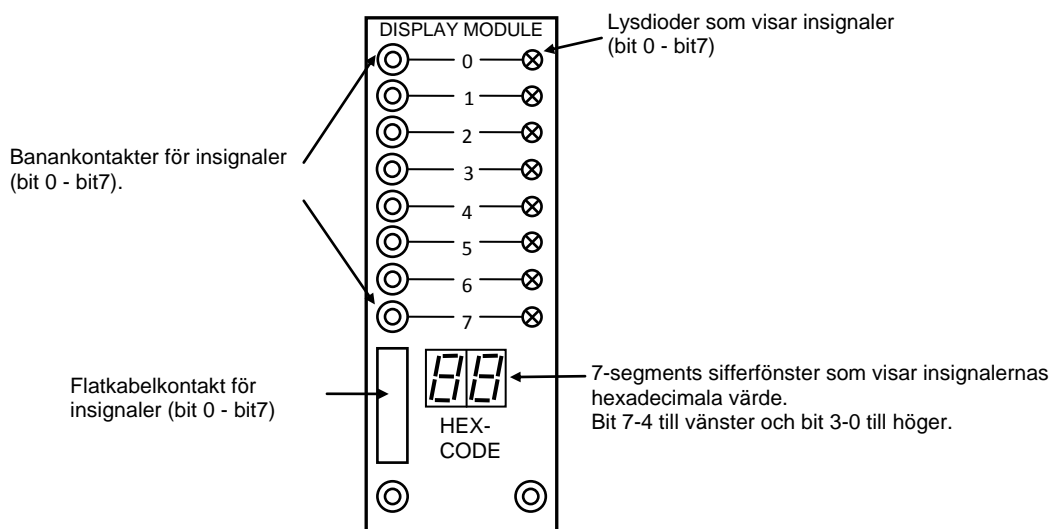
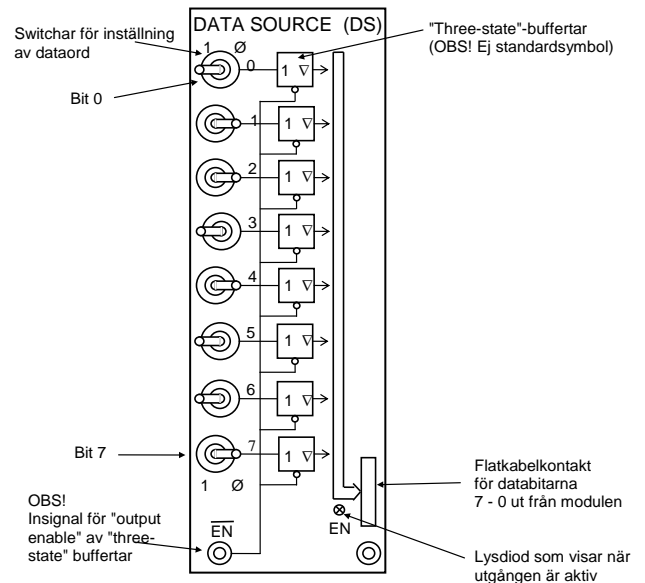
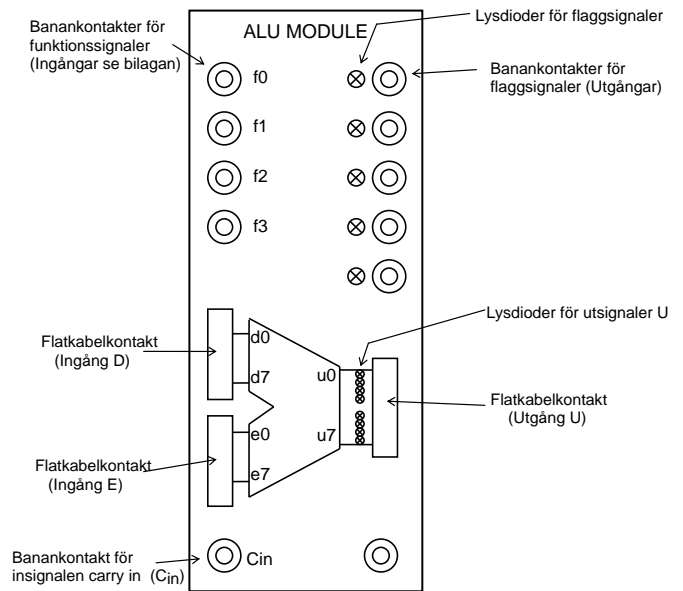
De båda dataingångarna D och E, samt datautgången U finns tillgängliga på frontpanelen i form av flatkabelkontakter. Insignalerna till D- och E-ingångarna och utsignalerna på U-utgången måste därför kopplas med flatkablar via dessa kontakter.

I labsystemet ingår också två moduler med namnet "DATA SOURCE" (DS). Med hjälp av dem kan man koppla in 8-bitars dataord till olika enheter via flatkablar. Dataorden kan ställas in med åtta switchar, som finns på DS-modulens framsida. Varje DS-modul har "three-state"-buffertar på sina åtta datautgångar mot flatkabelkontakten och kan därför användas som en av flera datakällor på samma buss. "Three-state"-bufferterna i en DS-modul kan aktiveras med signalen EN' via en banankontakt på frontpanelen.

Figuren till höger visar "DATA SOURCE"-modulens frontpanel med dataordet 01101001 inställt på switcharna (bit 7 - bit 0). Genom att aktivera ingången EN' nere till vänster på modulen lägger man ut bitmönstret på flatkabelkontakten nere till höger.

För att man enkelt skall kunna se värdet på 8-bitars tal från t ex ALU'n har labsystemet försetts med en displaymodul med 2 st 7-segments sifferfönster.

I sifferfönstren visas det binära 8-bitars värdet på flatkabelkontakten som ett hexadecimalt tal. Se figur nedan.





## Laborationsuppgift 7

### Undersökning av 8-bitars ALU.

Studera manualbladen för de moduler du skall använda.

Slå av nätströmbrytaren på digitalmaskinen.

**Läs om modulerna:**  
 ALU  
 DATA SOURCE (DS)  
 DISPLAY

Koppla sedan flatkablar mellan DS-modulerna och D- resp E-ingången på ALU-modulen.

Koppla också en flatkabel mellan ALU-modulens utgång U och displaymodulen.

Anslut ALU:ns funktionsingångar  $f_3$  - $f_0$  och  $C_{in}$  till switchmodulen nere till höger på digitalmaskinen enligt tabellen nedan.

B7	B6	B5	B4	B3	B2	B1	B0
$f_3$	$f_2$	$f_1$	$f_0$				$C_{in}$

### Glöm inte att båda DS-modulerna måste ha aktiva utgångar!

Slå på nätströmbrytaren på digitalmaskinen.

Bekanta dig med din uppkoppling och verifiera att denna fungerar korrekt. Detta gör du genom att exempelvis addera  $0+0$ . Ställ in enbart nollor på båda DS-modulerna, välj funktionskoden för addition och studera lysdioderna på ALU-modulens utgång U och värdet på DISPLAY-modulen. Testa sedan med  $0+1$ ,  $1+0$  och  $1+1$ . Du borde nu observera 1, 1 och 2 på ALU:ns utgång. Testa även värden där  $C_{in}$  ingår,  $D+E+C_{in}$ . Avsluta testförfarandet genom att utföra en subtraktion, exempelvis  $6-5$  och  $5-6$

Kontrollera att utsignalvärdena U(8) och flaggorna stämmer överens med de värden du har skrivit in i tabellen i **Uppgift 45** i arbetsboken.

Kontrollera åtminstone:  $55 + (-27)$ ,  $75+55$ ,  $27 - 55$ ,  $90 - (-55)$ ,  $90 \text{ AND } 15$ ,  $90 \text{ XOR } 90$ .

**Koppla ner dina lösningar och städa din arbetsplats så det ser snyggt ut för nästa grupp som kommer och skall laborera!**

TACK FÖR HJÄLPEN!

När du är klar med denna laboration har du

- lärt dig använda digitalmaskinen (Laborationsuppgift 1, 2 och 3)
- konstruerat och verifierat en kodomvandlare (Laborationsuppgift 4)
- konstruerat och verifierat en väljare (Laborationsuppgift 5)
- konstruerat en 2-bitars adderare (Laborationsuppgift 6)
- undersökt en 8-bitars ALU (Laborationsuppgift 7)



## Laboration nr 2 behandlar:

### *Räknare, Register med 3-state-utgång, Dataväg och Minne*

Följande uppgifter ur "Arbetsbok för DigiFlex" skall vara utförda och uppvisade för en handledare innan laborationen utförs.

<b>Uppg</b>	<b>58-61</b>	<b>65</b>	<b>68</b>	<b>71-77</b>	<b>89-91</b>	<b>96,97</b>
(Signatur)						

Hemuppgifterna i laborationsuppgiften till höger skall vara utförda och uppvisade för en handledare innan laborationen utförs.

<b>Uppg</b>	<b>3</b>	<b>4</b>
(Signatur)		

Följande laborationsuppgifter visas för en handledare innan de kopplas ner. Uppgift X får du under laborationstillfället.

<b>Uppg</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>X</b>
(Signatur)						

### **Laborationsuppgift 1**

#### **Synkron autonom räknare**

Slå av nätströmbrytaren på digitalmaskinen.

Koppla upp **Uppgift 91** sidan 108 från arbetsboken. Observera att du måste realisera AND-grinden med NAND-grindar. Använd en lämplig switch på switchmodulen som START-signal.

Slå på nätströmbrytaren på digitalmaskinen.

Testa räknaren och verifiera att den fungerar enligt tillståndsgrafan i uppgift 90 i arbetsboken.

**Läs om modulerna:**  
2-ingångars NAND-grindar  
JK-vippor

### **Laborationsuppgift 2**

#### **Räknaren 74HC163**

Slå av nätströmbrytaren på digitalmaskinen.

Du måste byta ut den ena modulen med JK-vippan mot räknarmodulen. Studera först undersidan av räknarmodulen och observera de två bananstift som fungerar som fästordning i digitalmaskinen (och även som strömförsörjning för modulen). Lyft modulen med JK-vippor genom att ta ett stadigt grepp i handtaget närmast dig på modulen. Dra modulen försiktigt *rakt upp*. Placera slutligen bananstiften på räknarmodulen *rakt ovanför* bananhylsorna på digitalmaskinen och tryck modulen på plats..

Koppla upp **Uppgift 97** sidan 116 från arbetsboken. Fundera själv, eller diskutera med en handledare om du behöver fler grindar (moduler) eller ej. Observera att du skall ha möjlighet att återstarta räknaren från tillstånd noll (RESET). Använd därför en lämplig switch på switchmodulen som RESET-signal. Vidare väljer du en switch på pulsmodulen som klocksignal. Tänk på att välja passande puls (positiv / negativ) för klocksignalen.

Slå på nätströmbrytaren på digitalmaskinen och testa sedan räknare. Verifiera att den fungerar enligt den givna räknesekvansen.

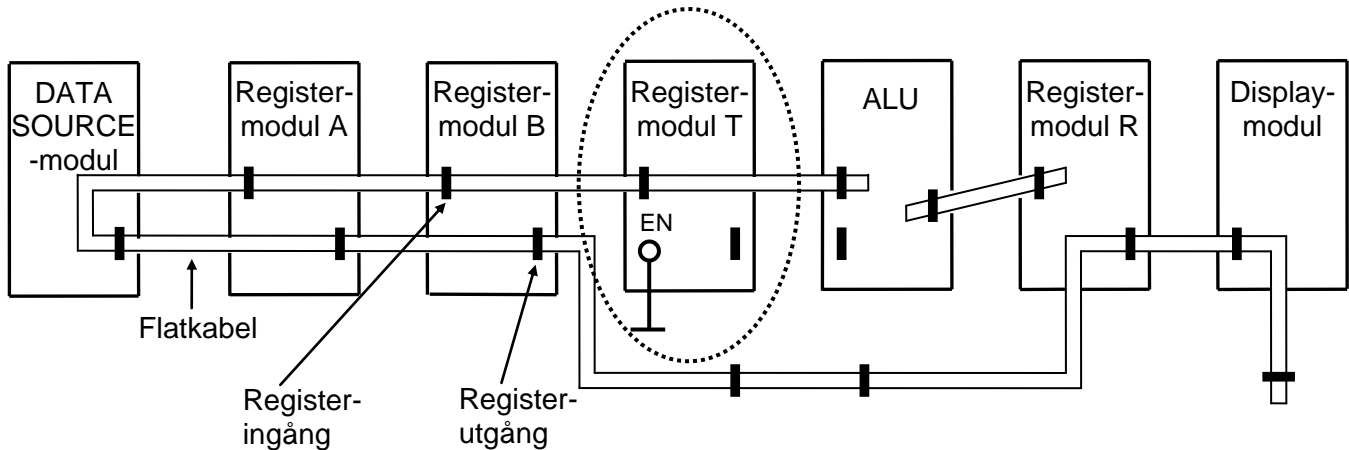
Återställ modulerna.

### Laborationsuppgift 3 (Kontakta handledaren som byter moduler!)

#### Undersökning av registermodulen (T).

Studera figuren nedan och kontrollera att den stämmer överens med uppkopplingen på digitalmaskinen. Kopplingen skall senare användas för att visa hur man kan flytta och behandla data i en enkel dataväg. Först skall vi dock studera hur en registermodul (REG8) fungerar.

**Läs om modulerna:**  
 DATA SOURCE (DS)  
 REG8  
 ALU  
 DISPLAY



Slå på nätströmbrytaren på digitalmaskinen.

Register T:s utgångar är inte anslutna till något. Studera modulen. Lägg märke till att lysdioderna på registermodulens utgångssida (till höger) visar ett odefinierat värde. Detta beror på att registermodulens utgångar inte är aktiverade och därför befinner sig i flytande tillstånd. De får alltså inget logikvärde från registrets Q-utgångar vars värden man ser på lysdioderna i mittraden.

Vidrör utgångskontaktens metallstift lätt med fingerspetsen och iakttag hur lysdioderna på utgångssidan ändrar ljusstyrka. Fenomenet förklaras av att laddningar omfördelas mellan fingret och de olika kontaktstiften. Detta visar också att "three-state"-utgångar som befinner sig i flytande tillstånd har mycket hög impedans (resistans) till matningsspänningen och jord

Om nödvändigt, lyft bort en modul på digitalmodulens mittersta rad. I "hålet" där modulen saknas hittar du två spännings-skenor. Den ena är +5V och den andra är jord (0V). Vidrör utgångskontaktens metallstift på registermodulen lätt med fingerspetsen *samtidigt* som du håller ett finger på en av spännings-skenorna. Testa detta ett antal gånger när du vidrör spännings-skenorna.

Diskutera med din handledare ifall du har frågor.

Läs hela kapitel 12 i arbetsboken och rita en principskiss av hur en "three-state"-utgång på registermodulen ser ut i boxen till höger. Det är tillräckligt att rita endast en bit.

Du skall nu använda T-registret för att undersöka hur registermodulernas styrsignaler fungerar och om de är aktiva som höga (ettor) eller låga (nollor).

**Hemuppgift 3**

Anslut T-registermodulens klockingång, Load Enable och Output Enable-signal till switch-modulen på digitalmaskinen.

Observera registrets innehåll och ge en klockpuls till registermodulen (fundera själv ut om det skall vara positiv eller negativ flank).

Ändrades registerinnehållet (ja/nej – varför/varför inte)? \_\_\_\_\_

\_\_\_\_\_

Ställ in värdet  $37_{16}$  på DS-modulen och ge sedan en klockpuls.

Ändrades registerinnehållet (ja/nej – varför/varför inte)? \_\_\_\_\_

\_\_\_\_\_

Ändra Load Enable-signalen. Ändra värdet på DS-modulen och ge

en klockpuls. Ändrades registerinnehållet (ja/nej – varför/varför inte)? \_\_\_\_\_

\_\_\_\_\_

Aktivera Output Enable-signalen. Vidrör utgångskontaktens metallstift på registermodulen lätt med fingerspetsen. Observera vad som händer på registrets utgång. Upprepa förfarandet när Output Enable inte är aktiverad. Varför ändras utgången ibland och ibland inte?

\_\_\_\_\_

\_\_\_\_\_

Gör en sammanställning i tabellen nedan över registermodulens styrsignaler Load Enable, Output Enable, klockpuls etc. Ange alla ingångar M1, M2, M3, C4 och EN. Ange om signalerna är aktiva som nollor eller ettor. Är du osäker testar du genom att klocka in nya värden i registret och belasta utgången (beröra utgångskontaktens metallstift) på registermodulen.

Ingång	Beteckning	Aktiv	Kommentar
M1			
M2			
M3	Load Enable	Låg (0)	
C4			
EN			

Slå av nätströmbrytaren, låt flatkablarna sitta kvar och avlägsna övriga kablar.

## Laborationsuppgift 4

### Enkel dataväg

Studera figur 34, 35 och speciellt figur 36 på sidan 70-74 i arbetsboken och jämför dessa med nedanstående figur som visar uppkopplingen för denna laborationsuppgift. En lång flatkabel med många kontakter är ansluten till de fyra registermodulerna, Data-Source-modulen (DS-modulen) och displaymodulen. Flatkabeln fungerar då som en databuss.

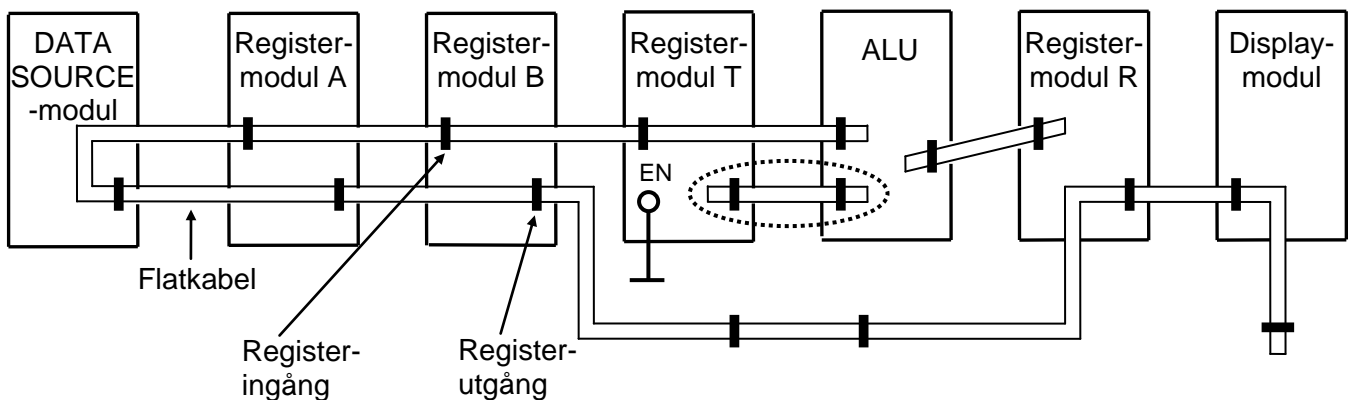
#### Läs om modulerna:

Reg8  
Displaymodul  
DATA SOURCE  
ALU

Studera hylsdonen (hon-kontakterna) på flatkabeln och observera att de kan placeras på endast ett sätt på stifttagen (hankontakterna) på modulerna. När du skall avlägsna en kabel, **dra då inte i själva kabeln**, utan använd utkastar-armarna som finns på stifttagen på modulerna. Studera dessa!

ALU-modulen använde du i laboration 1 tillsammans med DS-modulerna. Register T skall nu användas för lagring av den ena ALU-operanden E. Koppla därför en flatkabel mellan register T:s utgång och ALU:ns E-ingång enligt figuren nedan.

Den andra ALU-operanden D är inkopplad direkt från bussen. Register R används för lagring av ALU:ns resultat U.



**Styrsignalen  $OE_T$  skall nu ständigt vara aktiv då register T:s utgång hela tiden skall vara kopplad till ALU:ns E-ingång. Koppla därför  $OE_T$ -signalen till lämplig nivå (+5V eller 0V)**

På digitalmodulens mittersta rad finns en manuell styrenhet. Koppla följande signaler:

- Anslut styrsignalerna  $OE'$ ,  $LD'$  och  $CP$  för samtliga registermoduler till respektive kontakt på den manuella styrenheten. ( $OE_T$  skall inte kopplas från den manuella styrenheten!)
- Anslut  $OE'$ -signalen för DS-modulen ( $EN'$ ) till kontakten  $OEDS'$  på styrenheten.
- Koppla även ALU:ns styr signaler (Funktion och  $Cin$ )

Slå på nätströmbrytaren på digitalmaskinen.

Lägg först in värdena  $10_{16}$ ,  $3C_{16}$ , resp  $22_{16}$  i register **A**, **B**, respektive **R** m h a DS-modulen.

Skriv in RTN-beskrivning och styrsignalerna nedan för att byta innehållen i register A och B. Register A skall alltså ha värdet  $3C_{16}$  och register B  $10_{16}$  efter bytet. Register R skall fortfarande vara  $22_{16}$ . Under detta moment får du inte utnyttja DS-modulen. Testa förloppet i kopplingen.

RTN	Styrsignaler	<b>Hemuppgift 4</b>

Utför nu **Uppgift 71** sidan 80 ur arbetsboken. I uppgiften anges att register B skall ha värdet  $40_{16}$  från start. Här skall register B vara  $60_{16}$  från start. A har fortfarande värdet  $27_{16}$ . Vi tolkar nu bitmönstren som tal utan tecken  $[0,255]$ .

Fyll i tabellen nedan med hexadecimala värden. Du skall ange de nya registerinnehållen efter det att du gett en klockpuls. Ange även flaggbitarnas värden.

CP	Register A	Register B	Register T	Register R	Lysdioder på ALU-modulen			
					N	Z	V	C
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								

Troligen har du fått korrekt svar (för tal utan tecken). Resultatet borde få plats ty  $60_{16} = 96_{10}$ ;  $27_{16} = 39_{10}$  och  $(3B - 2A) = 3 \cdot 96 - 2 \cdot 39 = 210_{10}$ . Då 210 ligger i talområdet  $[0,255]$  borde det vara korrekt. (Testa med en fickräknare.)

Vad borde rätt resultat vara på hexadecimal form?

Jämför med vad du fått ovan. Använd penna och papper.  $3B - 2A =$  \_\_\_\_\_

Kommentera ditt resultat! Blev det fel? \_\_\_\_\_

Trots att ditt slutresultat är korrekt så kan uträkningen innehålla felaktiga delresultat. Detta beroende på hur du löser uppgiften. Som vi såg tidigare så får resultatet plats eftersom  $60_{16} = 96_{10}$ ;  $27_{16} = 39_{10}$  och  $(3B - 2A) = 3 \cdot 96 - 2 \cdot 39 = 288_{10} - 78_{10} = 210_{10}$ .

Problemet är om vi först gör additionen  $(B+B+B = 3B) = 3 \cdot 96 = 288_{10}$  som ligger utanför talområdet  $[0,255]$  och därefter utför subtraktionen. Man tycker att vi då borde få ett felaktigt resultat av subtraktionen. Blir resultatet av subtraktionen felaktigt?.

Svar: \_\_\_\_\_

Förklara vad som händer vid subtraktionen ovan.

---

---

Diskutera din lösning med en handledare om du är osäker.

Ge en RTN-beskrivning som löser problemet utan att något delresultat blir felaktigt.

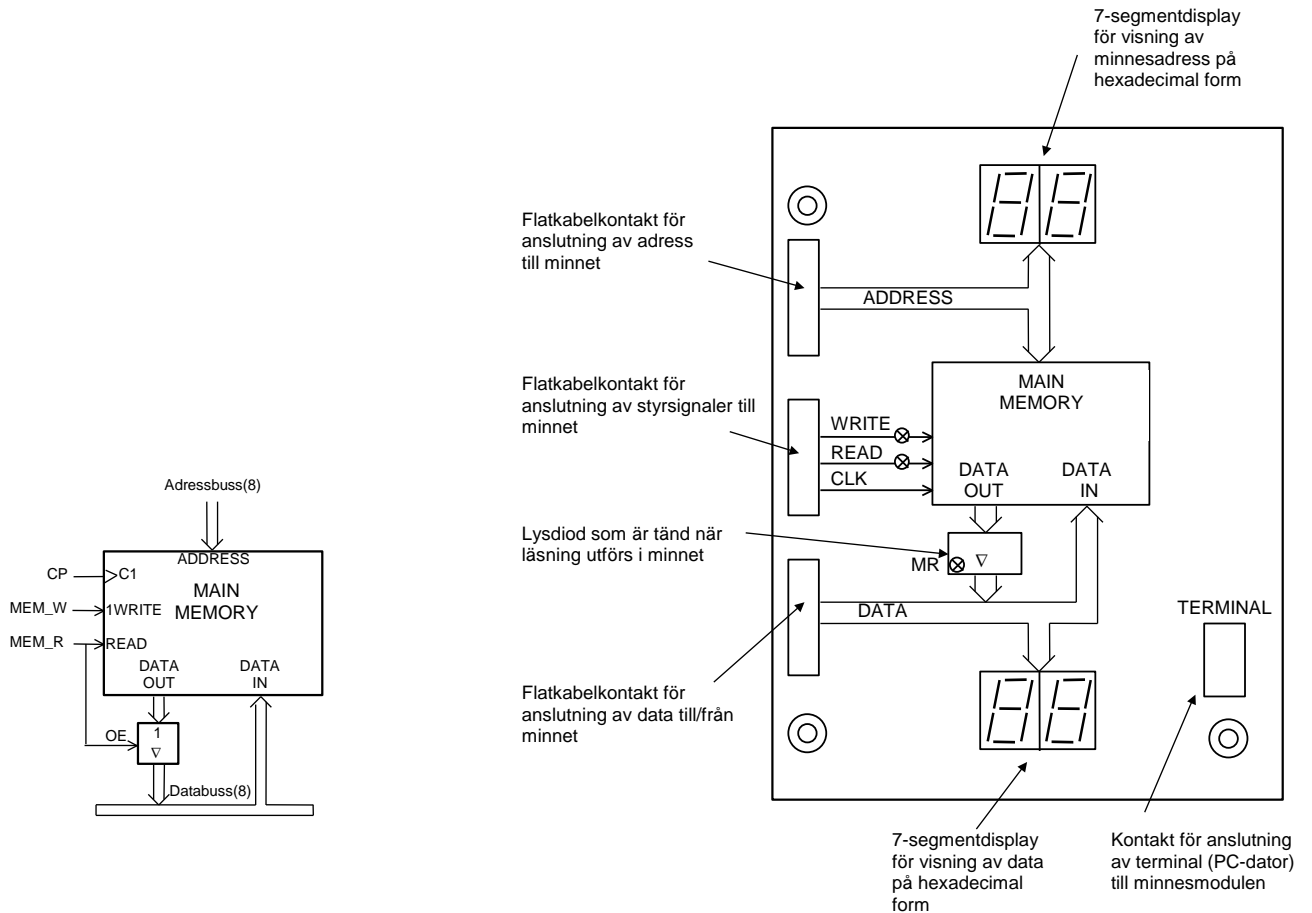
1:	_____
2:	_____
3:	_____
4:	_____
5:	_____
6:	_____
7:	_____
8:	_____
9:	_____
10:	_____
11:	_____
12:	_____
13:	_____

**Behåll din koppling till nästa uppgift!**

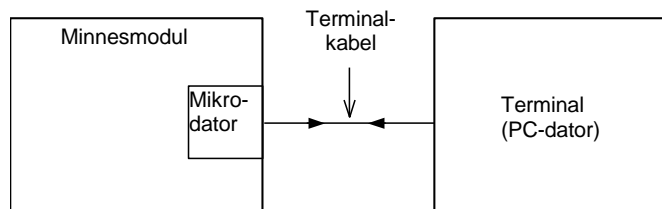


## Beskrivning av minnesmodul och terminalanslutning

I labsystemet ingår en modul med läs- och skrivbart minne, RWM, med kapaciteten  $2^8 = 256$  st 8-bitars minnesord. Blockschemat för RWM-modulen och dess frontpanel visas nedan.



För att man lättare skall kunna studera och ändra minnesinnehållet har en mikrodator byggts in i minnesmodulen som finns på labplatsen. Mikrodatorn kan kommunicera med en terminal (PC-dator) såsom visas i figuren nedan. Kommunikationen sker via terminalkontakten nere till höger på frontpanelen.



Ett program i minnesmodulens mikrodator visar innehållet på hexadecimal form på alla 256 adresser på terminalens skärm.

Man kan enkelt ändra ett dataord på en minnesadress genom att först flytta markören på skärmen till den önskade adressen med hjälp av tangenterna på terminalens tangentbord. Därefter skriver man in det nya dataordet på hexadecimal form med siffer- och bokstavstangenterna.

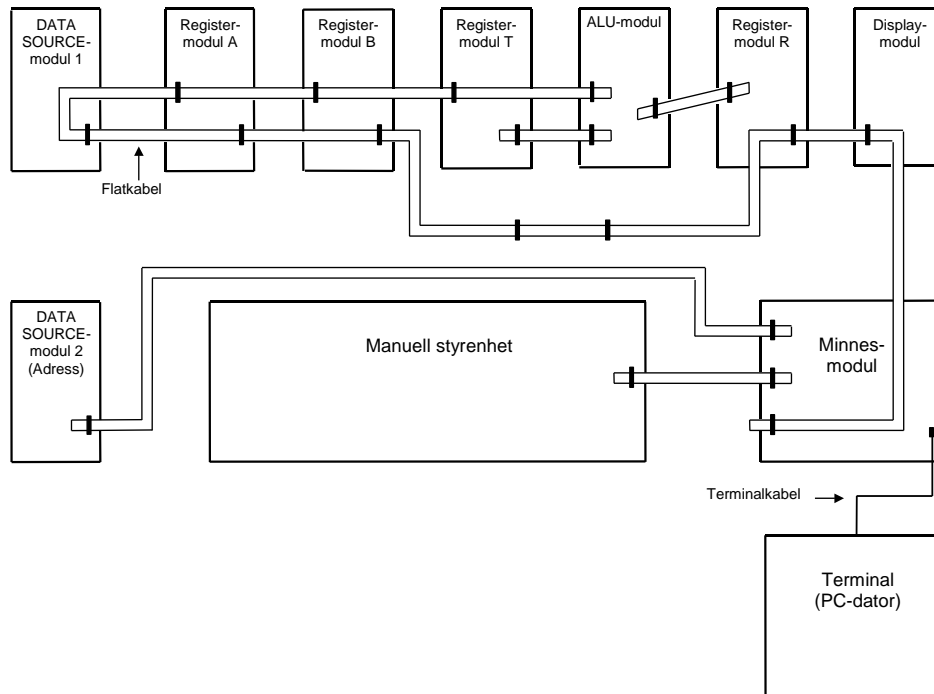
## Laborationsuppgift 5

### Anslutning av en minnesmodul till datavägen

**Läs om modulerna:**  
Enkel manuell styrenhet  
Minnesmodul  
Terminal

Slå av nätströmbrytaren på digitalmaskinen.

Studera figur 40 i arbetsboken. Jämför med figuren nedan och motsvarande uppkoppling på digitalmaskinen. Observera speciellt kablarna för överföring av styrsignaler, adress, data, etc. mellan modulerna



Koppla in minnesmodulen enligt figuren ovan. **Se till att  $OE_{DS2}$  är aktiv.**

Starta programmet "Flexminne" på din PC. Fråga en handledare vid problem.

Studera hur data skrivs till terminalen när du nu slår på nätströmbrytaren på digitalmaskinen. Det är innehållet i FLEX-datorns minnesmodul som skrivs ut på skärmen. För att uppdatera Flexminnet tryck CtrlA upprepade gånger. Testa detta.

Ange data som är lagrade på följande minnesadresser:

$20_{16}$ : \_\_\_\_\_,  $3B_{16}$ : \_\_\_\_\_,  $AA_{16}$ : \_\_\_\_\_,  $CF_{16}$ : \_\_\_\_\_,

$70_{16}$ : \_\_\_\_\_,  $AB_{16}$ : \_\_\_\_\_,  $EA_{16}$ : \_\_\_\_\_,  $FF_{16}$ : \_\_\_\_\_

Slå av nätströmbrytaren på digitalmaskinen, vänta några sekunder och slår på nätströmbrytaren igen. *Det är aldrig bra att **snabbt** slå av och på nätströmbrytaren på elektroniska apparater!!!*

Fundera över följande och diskutera med din handledare vid lämplig tidpunkt.

Hittas något "mönster" i utskriften på skärmen? Jämför ovanstående med någon annan laborationsgrupp. Vem har skrivit in det data du läser på skärmen? Läste du samma minnesinnehåll efter att slagit av och på strömmen?

Nu skall minnesmodulens funktion undersökas. Följ instruktionerna:

1. Tryck på terminalens ENTER-knapp och iakttag bildskärmen. Fyll hela minnet med nollor genom att hålla siffertangenten 0 nedtryckt tills minnet är fullt.
2. Ställ sedan in adressen  $30_{16}$  på DS-modul 2, som är ansluten till adresskontakten, samt dataordet  $5A_{16}$  på DS-modul 1, som är ansluten till datakontakten.
3. Aktivera DS-modul 1 genom att sätta switchen OEDS = 1 på styrenheten. Kontrollera att dataordet  $5A_{16}$  finns på databussen.
4. Ställ switchen MEM\_W i läget 1 på den manuella styrenheten medan du ser på minnesmodulen.
5. Ge en klockpuls medan du ser på bildskärmen.
6. Fortsätt att skriva och läsa några olika dataord med hjälp av DS-modulerna och styrenheten. Kontrollera att resultatet överensstämmer med informationen på bildskärmen.
7. Prova sedan med att ändra minnesinnehållet med hjälp av terminalens pil- och siffertangenter.
8. Ställ in en viss minnesadress på DS-modul 2 och ställ switchen MEM\_R = 1 (läsning) och ändra minnesinnehållet på den valda adressen med hjälp av terminalen. Observera ändringen på minnesmodulens datadisply.

Lägg in datavärdet  $24_{16}$  på adress 4 och datavärdet  $13_{16}$  på adress 5 i minnet med hjälp av terminalen.

Genomför sedan tabelloperationerna i uppgift 76 i arbetsboken. Kontrollera att resultatet i adress 6 i minnet blir korrekt.

**Låt den långa flatkabeln sitta kvar, koppla ner dina lösningar och städa din arbetsplats!**

---

När du är klar med denna laboration har du

- kopplat upp och testat en synkron räknare. (Labuppgift 1)
- konstruerat, kopplat upp och testat en räknare som kommer att användas i styrenheten för FLEX-processorn. (Labuppgift 2)
- undersökt begreppet 3-state. (Labuppgift 3)
- visat att du kan hantera ett register med laddfunktion och 3-stateutgång. (Labuppgift 3)
- visat att du behärskar dataöverföring mellan register via en buss. (Labuppgift 4)
- utfört aritmetiska operationer i en enkel dataväg. (Labuppgift 4)
- kopplat in ett minne till datavägen och utfört en aritmetisk operation på minnesdata. (Labuppgift 5)



**Laboration nr 3 behandlar:*****Styrenheter för FLEX-processorn***

Följande uppgifter ur "Arbetsbok för DigiFlex" skall vara utförda och uppvisade för en handledare innan laborationen utförs.

<b>Uppg</b>	<b>64</b>	<b>67</b>	<b>81</b>	<b>100-104</b>	<b>115</b>
(Signatur)					

Hemuppgifterna i laborationsuppgiften till höger skall vara utförda och uppvisade för en handledare innan laborationen utförs.

<b>Uppg</b>	<b>1</b>	<b>2</b>
(Signatur)		

Följande laborationsuppgifter demonstreras för en handledare innan dessa kopplas ner. Uppgift X får du under laborationstillfället.

<b>Uppg</b>	<b>1</b>	<b>2</b>	<b>X</b>
(Signatur)			

*SNÄLLA NI.....SKRIV INTE PÅ LAB-MODULERNA och "peka inte med pennor" på dem.*

**Laborationsuppgift 1*****Manuell styrning av FLEX-datorn***

Slå av nätströmbrytaren på digitalmaskinen.

Studera uppkopplingen av datavägen, styrenheten, minnesmodulen och terminalen. Observera speciellt kablarna för överföring av styrsignaler, adress, data, etc. mellan modulerna

Starta programmet "Flexminne" på din PC.

Skriv noll till alla minnespositioner genom att trycka på siffran noll på tangentbordet och håll denna nertryckt. Studera utskriften till skärmen.

Välj nu att nollställa registren i datavägen genom att ettställa CLEAR och ge sedan en klockpuls på styrenheten. Nollställ sedan CLEAR.

Du har nu värdet 0 i register MA. Läs nu innehållet på adress 0 i minnet genom att ettställa styrsignalen MR på styrenheten. Observera hur indikatorerna tänds på minnesmodulen. Nollställ sedan MR.

Använd terminalen och skriv in  $3A_{16}$  på adress 0 i minnet. Du kan manövrera dig till olika minnespositioner genom att använda piltangenterna på tangentbordet. Ettställ på nytt MR på styrenheten och verifiera att  $3A_{16}$  överförs på databussen till datavägen. Nollställ sedan MR.

**Läs om modulerna:**

FLEX Dataväg  
Manuell styrenhet  
Minnesmodul  
Terminal

**RESET – FETCH - EXECUTE**

Du skall nu för hand genomföra RESET, FETCH och EXECUTE-faserna för ett litet program som visas nedan.

Lägg först in maskinprogrammet i minnet med start på adress  $20_{16}$ . Använd terminalen. Placera även  $20_{16}$  på adress  $FF_{16}$  i minnet för att initiera RESET-vektorn.

Studera **Uppgift 64** och **67** sidan 77 i arbetsboken där du angett styrsignalsekvenser för INCB och COMB. Instruktionerna INCA och COMA löses på liknande sätt. Observera att dessa instruktioner också påverkar flaggbitarna. Studera även dessa instruktioner i instruktionslistan!

Adress  
(Hex)

1F		
20	0F	LDAA   #\$7F
21	7F	
22	41	INCA
23	23	COMA
24		

Studera figur 44, 46 och 47 i arbetsboken. Delar av informationen är ifylld i tabellen på nästa sida. Kolumnen R/F/E anger om RESET, FETCH eller EXECUTE-fasen utförs. De högra kolumnerna anger aktuellt värde i register och på bussar i varje tillstånd (*före* klockpuls).

Fyll i tabellen på nästa sida *med blyertspenna*. (Hemuppgift 1)

Databussens värde läser du av på datavägen och adressbussens värde är samma som innehållet i register MA.

När du nu genererar styrsignaler till datavägen med den manuella styrenheten, tänk på att om du gör fel, så måste du börja om från början. Så var noggrann och metodisk och använd arbetsgången nedan.

1. Ettställ önskade styrsignaler
2. Studera *aktiverade styrsignaler i datavägen* och jämför med din RTN-beskrivning
3. Ställ dig frågan: **Var det detta jag ville? Ändra eventuellt styrsignaler.**
4. Ge en klockpuls
5. Fick du rätt värden i dina register, enligt RTN-beskrivningen? **Var det detta jag ville?**
6. Nollställ alla styrsignaler
7. Börja om från 1

Genomför RESET, FETCH och EXECUTE-faserna för instruktionssekvensen ovan och kontrollera och ändra eventuellt i tabellen nedan. Kolumn 2 i tabellen anger faserna R: RESET, F: FETCH och E: EXECUTE.

**Hemuppgift 1**

State	R/ F/ E	RTN	Styrsignaler (=1)	Register					Bussar	
				A	R	MA	PC	I	Data	Adr
0	R	FF <sub>16</sub> →R	f <sub>3</sub> f <sub>2</sub> f <sub>1</sub> f <sub>0</sub> , LD <sub>R</sub>	0	0	0	0	0	3-state	0
1	R	R→MA	OE <sub>R</sub> , LD <sub>MA</sub>	0	FF	0	0	0	FF	
2	R	M→PC	MR, LD <sub>PC</sub>	0	FF	FF				
3	F	PC→MA, PC+1→PC	OE <sub>PC</sub> , LD <sub>MA</sub> , IncPC							
4	F	M→I	MR, LD <sub>I</sub>							
5	E	PC→MA, PC+1→PC	OE <sub>PC</sub> , LD <sub>MA</sub> , IncPC							
6	E	M→A	MR, LD <sub>A</sub>							
7	F	PC→MA, PC+1→PC								
8	F	M→I								
9	E									
10										
11										
12										
13										
14										
15										

Diskutera tabellen med din handledare. Koppla ner dina lösningar.

**Moduler**  
Räknare CRT4  
INVERTERARE  
Avkodarenhet  
Styrsignalmodul  
Flex dataväg  
Minnesmodul

## Laborationsuppgift 2

**(Kontakta handledaren som byter moduler och kopplar in deras matningsspänning!)**

**Implementering av ett antal maskininstruktioner i en fast kopplad styrenhet.**

### Hemuppgift 2

Läs igenom kapitel 22 tom avsnitt 22.4 i arbetsboken. Studera speciellt figurerna 60 och 61

Det kombinatoriska nätet i figur 60 utgörs i laboratoriet av två specialgjorda moduler, en avkodarenhet och en styrsignalmodul. Den sistnämnda modulen innehåller även en klockgenerator. Studera modulerna i databladet där du även hittar uppkopplingen för denna laborationsuppgift under rubriken "Konfiguration för Laboration 3 (del 1)".

Studera figur 61 i arbetsboken. För att bilda styrsignalernas summertermer används OR-grindar med många ingångar. Styrsignalmodulen saknar OR-grindar för vissa styrsignaler, exempelvis LD<sub>CC</sub> överst till vänster. Detta beror på att modulen är optimerad för de instruktioner du skall implementera. Då LD<sub>CC</sub>-signalen enbart skall aktiveras från "ett ställe" kan OR-grinden utelämnas.

Då styrsignalmodulen är optimerad kommer varje ingång på OR-grindarna att utnyttjas för att bilda summertermerna för styrsignalerna till datavägen. Observera att en och endast en signal får anslutas till var och en av ingångarna på OR-grindarna. Om du upptäcker att du har för få ingångar på OR-grindarna, har du tänkt fel eller kopplat fel.

Styrsignalmodulen skall placeras på arbetsbänken framför digitalmaskinen.

**Undersidan av modulen får inte lutas mot digitalmaskinen då detta kan orsaka kortslutning i uppkopplingen.**

Slå av nätströmbrytaren på digitalmaskinen.

### Kontrollera följande:

Matningsspänning är kopplad till styrsignalmodulen. +5V och jord hämtas från röda och svarta bananhylsor på datavägen.

En flatkabel för OP-koden och flaggorna är kopplad mellan "CTRL OUT" på datavägen och "CTRL IN" på avkodarmodulen.

En flatkabel för styrsignaler är kopplad mellan "CTRL OUT" på styrsignalmodulen och "CTRL IN" på datavägen.



**Uppkoppling och test av styrenhetens räknare (Detta kopplar du själv!)**

Använd i första hand korta kopplingskablar.

Koppla upp signalerna till räknaren enligt figur 60 i arbetsboken.

Se till att värdet 0011 = 3 kopplas till dataingångarna på räknaren. Värdet hämtas lämpligen från röda och svarta bananhylsor.

NF-signalen från avkodarenheten inverteras och används som LOAD-signal för räknaren.

Räknarens klocksignal hämtas från en bananhylsa på styrsignalmodulen.

Använd switch B7 på switchmodulen som Reset-signal.

Räknarens utsignaler  $q_3$ -  $q_0$  kopplas till "STATE DECODER" på avkodarmodulen.

Ställ switchen på klockgeneratoren på styrsignalmodulen i läget "SINGLE CLOCK".

Slå på nätströmbrytaren på digitalmaskinen och behåll den på i fortsättningen.

Studera räknarens utgång och tillståndsindikatorerna på avkodarenheten när du ger ett par klockpulser med klockgeneratoren på styrsignalmodulen. Händer inget kan detta bero på att räknarens Reset-signal är aktiv, switch B7.

Verifiera att du genomlöper alla tillstånd och slutligen hamnar i ERR-tillståndet (ERROR). Aktivera Reset och verifiera att du kan starta om från tillstånd 0.

Studera figur 59 i arbetsboken. Verifiera sedan att du startar om från tillstånd 3 när NF aktiveras. Detta gör du genom att temporärt koppla en labsladd mellan bananhylsan för Q4 och ingången på OR-grunden för NF. Du borde nu ha en räknare som efter Reset genomlöper tillstånden 0-1-2-3-4-3-4-3-4 etc. Ta bort den temporära labsladden när räknaren fungerar korrekt.

**Implementering av RESET**

RESET-fasen är beskriven i avsnitten 22.1-3 i arbetsboken (och i kapitel 18) och återges nedan.

Tillstånd	Summaterm	RTN-beskrivning	Styrsignaler (=1)
$Q_0$	$Q_0$	$FF_{16} \rightarrow R$	$f_3, f_2, f_1, f_0, LD_R$
$Q_1$	$Q_1$	$R \rightarrow MA$	$OE_R, LD_{MA}$
$Q_2$	$Q_2$	$M \rightarrow PC$	$MR, LD_{PC}$

Mata in instruktionssekvensen från **uppgift 103** i arbetsboken med hjälp av terminalen. Mata också in värdet  $20_{16}$  (startadressen) på adressen  $FF_{16}$  i minnet (RESET-vektorn).

Starta "körning" av instruktionssekvensen genom att aktivera Reset-switchen, switch B7, och ge en klockpuls genom att trycka på knappen CLOCK. Nollställ sedan Reset-switchen igen.

Signalen  $Q_0$  på tillståndsavkodaren skall nu ha blivit aktiv.

Observera att nätströmbrytaren på digitalmaskinen skall vara påslagen under hela kopplingsarbetet.

När du implementerar RESET, FETCH och utförandefaser av instruktioner blir det oftast fel då det är många kablar som skall kopplas. Använd därför arbetsgången nedan, som visar implementering av tillståndet  $Q_0$ , där summatermen är  $Q_0$  och styrsignalerna  $f_3$ ,  $f_2$ ,  $f_1$ ,  $f_0$  och  $LD_R$  skall aktiveras. Se tabellen för RESET-fasen ovan.

- Verifiera att önskat tillstånd,  $Q_0$  i detta fall, är aktiverat och att dess lysdiod är tänd.
- Ta en *lång* labsladd och koppla den från  $Q_0$  på avkodarenheten till  $f_3$  på styrsignalmodulen. *Behåll ditt pekfinger på labsladden vid  $f_3$ !!!*
- Fortsättningsvis används *korta* labsladdar. Koppla en labsladd från ditt pekfinger ( $f_3$ ) till  $f_2$ . *Behåll ditt pekfinger på labsladden vid  $f_2$ !!!*
- Koppla vidare på samma sätt till  $f_1$ ,  $f_0$  och  $LD_R$ . *Först nu kan du vila ditt pekfinger.*
- Verifiera att dina angivna styrsignaler är aktiva i datavägen.
- Ge en klockpuls med klockgeneratoren på styrsignalmodulen.
- Observera att du nu är i nästa tillstånd,  $Q_1$  i detta fall.
- Kontrollera slutligen att utförd operation ( $FF_{16} \rightarrow R$ ) enligt RTN-beskrivningen är utförd.

<p><b>Var det detta jag ville?</b></p>
--

*Om förloppet i datavägen inte stämmer överens med din RTN-beskrivning, be då om hjälp efter att ha felsökt i 5 minuter.*

Då ni jobbar i grupp, läser lämpligen den ena personen styrsignaler i tabellen och levererar korta respektive långa kablar till sin partner som använder pekfingeret och kopplar. Verifieringar efter koplandet gör ni gemensamt.

*Vi påminner om att endast en signal, får anslutas till var och en av ingångarna på OR-grindarna på styrsignalmodulen. Se inledningen på denna laborationsuppgift. Samma signal kan givetvis kopplas vidare till flera ingångar enligt arbetsgången ovan.*

Fortsätt med att implementera RESET- och FETCH-fasen enligt metoden ovan. Följ förloppet i datavägen kontrollera och att OP-koden för LDAA-instruktionen överförs till instruktionsregistret.

I EXECUTE-tillstånden från tillstånd  $Q_5$  och högre, krävs en grindning med aktuell OP-kod. Studera hur den gröna indikatorn för  $Q_5 \bullet I_{OF}$  på avkodarenheten tänds upp och implementera nu utförandefasen av instruktionen med arbetsmetoden beskriven tidigare. LDAA #Data är beskriven på sidan 129 i arbetsboken.

De övriga instruktionerna hittar du i **uppgift 100, 101** och **102** i arbetsboken.

Fortsätt på samma sätt tills alla fyra instruktionerna är implementerade.

Klocka igenom programsnurren en gång till och observera att minnesinnehållet på adress 05 ökas.

Välj kontinuerlig klocka på styrsignalmodulen genom att ändra switchen SINGLE CLOCK / CONT CLOCK och observera förloppet. Du bör se att register A hela tiden ändras och att innehållet i minnesadress 5 hela tiden ökas. Kanske uppfattar du också att det blinkar till i Z-flaggan när register A inkrementeras (ökas) från  $FF_{16}$  till  $00_{16}$ .

**Du har nu konstruerat styrenheten till en processor!**

Skriv nu ett eget program till din processor. Programmet får endast innehålla de fyra instruktionerna LDAA #Data, STAA Adr, INCA och JMP Adr som du implementerat. Programmet skall vara utformat som en evighets slinga. Samma instruktion får förekomma på flera ställen i programmet. Översätt programmet till maskinkod med hjälp av instruktionslistan för FLEX-processor eller utgå från din lösning av uppgift 103 i arbetsboken.

Observera att du måste ändra RESET-vektorn till den nya startadressen. **Fyll i nedanstående hemma.**

Adress (Hex)	Assemblerprogram	<b>Hemuppgift</b>
2F		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
3A		
3B		
3C		

Demonstrera för- och diskutera med en handledare.

Slå av nätströmbrytaren på digitalmaskinen och koppla ner din styrenhet.

### Koppla ner dina lösningar och städa din arbetsplats!

---

När du är klar med denna laboration har du:

- manuellt gett styrsignalsekvenser för att starta och utföra ett program för FLEX-datorn (Laborationsuppgift 1)
- konstruerat en styrenhet för FLEX-datorn och därmed fullföljt bygget av en riktig (visserligen enkel, men en *riktig*) dator. (Laborationsuppgift 2)



**Laboration nr 4 behandlar:*****Programmering av FLEX-datorn i maskin- och assemblerspråk***

Följande uppgifter ur Arbetsbok för DigiFlex skall vara utförda och uppvisade för en handledare innan laborationen utförs

<b>Uppg</b>	<b>124</b>	<b>125</b>	<b>127</b>	<b>128</b>
(Signatur)				

Hemuppgifterna till höger skall vara utförda och uppvisade för en handledare innan laborationen utförs.

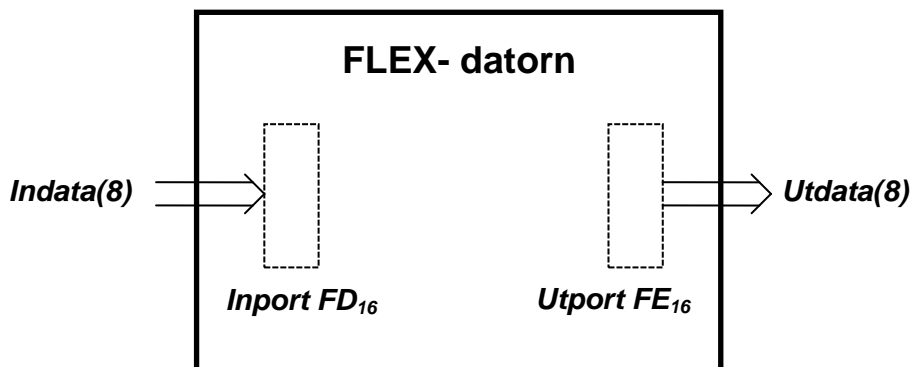
<b>Uppg</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
(Signatur)						

Laborationsuppgifterna till höger demonstreras för en handledare innan nästa påbörjas. Uppgift X får du under laborationstillfället.

<b>Uppg</b>	<b>2</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>X</b>
(Signatur)					

*SNÄLLA NI.....SKRIV INTE PÅ LAB-MODULERNA och "peka inte med pennor" på dem.*

Under hela denna laboration är FLEX-datorn färdiguppkopplad och försedd med en inport på adressen  $FD_{16}$  och en utport på adressen  $FE_{16}$  enligt figuren nedan.

***Uppgift 1(endast hemuppgift)******Studium av programmet "ETERM for FLEX"***

Läs igenom arbetsboken kapitel 23-27 och arbeta igenom samtliga uppgifter.

**Hemuppgift 1**

***Slut på uppgift 1***

## Uppgift 2

### Branch-instruktioner

**Läs om modulerna:**  
FLEX Dataväg  
Minnesmodul  
Terminal

### Hemuppgift 2

Översätt instruktionssekvensen nedan till maskinkod med hjälp av instruktionslistan för FLEX-processorn. Instruktionssekvensen är tänkt att placeras i minnet med början på adressen  $20_{16}$ .

Adress (Hex)			
1F			
20	0F	LDA	#\$FC
21			
22		INCA	
23		BNE	\$22
24			
25		BRA	\$20
26			
27			

Diagram description: A table with 4 columns: Address (Hex), a blank column, instruction name, and operand. Rows 20-25 contain instructions: LDA #\$FC, INCA, BNE \$22, and BRA \$20. Brackets on the right group rows 20-25 as 'Inre program-snurra' and rows 20-27 as 'Yttre program-snurra'.

Skapa en mapp, *Lab4*, där du kan spara dina egna filer.

Starta *ETERM 6 for FLEX* och öppna fliken *Debug/Simulator* (FLEX-datorsimulatoren).

Lägg maskinkoden från *hemuppgift 2* direkt i minnet. Det enklaste sättet är att markera ett minnesord med musen, skriva in data och sedan flytta till nästa minnesord med piltangenterna.

Lägg också in värdet  $20_{16}$  på adressen  $FF_{16}$  i minnet ( $20_{16}$  = Startadressen).

Hämta startadressen till programräknaren genom att klicka på knappen "*Reset av FLEX-processorn*" (= "*Reset CPU*"), uppe till vänster i simulatorfönstret. Lägg märke till att instruktionssekvensen nu visas i klartext (= disassemblerad form) i subfönstret "*Program*" i simulatorfönstret.

Stega sekvensen, instruktion för instruktion, genom att upprepade gånger klicka på knappen "*Step*" högst upp till höger i subfönstret "*Program*". Iakttag vad som händer medan du stegar programslingan några varv. I subfönstret "*Control*" finns räknare för antal minnesord (= bytes) och antal klockpulser för varje instruktion som utförs. Lägg märke till att endast en av instruktionerna påverkar flaggregistret.

Hur många gånger exekveras den inre program snurran innan du utför BRA-instruktionen? \_\_\_\_\_

Ändra nu BNE-instruktionen till en BCC-instruktion och undersök hur många varv den inre programsnurran utförs nu: \_\_\_\_\_

Välj nu en BMI-instruktion i stället.  
Hur många varv blir det nu? \_\_\_\_\_

Prova att köra programslingan med knappen "*Kör program långsamt*" och stoppa sedan körningen med knappen "*Stoppa programkörning*". Prova sedan med knappen "*Kör program snabbt*" och stoppa sedan. Iakttag vad som händer.

Du skall nu spara simulatorns tillstånd, dvs registerinnehåll och minnesinnehåll, i en fil så att du kan hämta tillbaka det och köra det vid ett senare tillfälle. Högerklicka därför någonstans i simulatorfönstret, men undvik minnesfönstret och programfönstret! Välj "*Save state*" i den meny som visas och ge filnamnet *hem2*. Spara i din filmapp!

## **Laborationsuppgift:**

### **Programexekvering i FLEX-datorn**

Studera uppkopplingen av datavägen, styrenheten, minnesmodulen och terminalen. Studera också FLEX styrenhet i manualbladen.

Starta programmet "Flexminne" på din PC.

Slå på nätströmbrytaren på digitalmaskinen som skall innehålla hela FLEX-datorn.

Skriv noll till alla minnespositioner genom att trycka på siffran noll på tangentbordet och håll denna nertryckt. Du kan manövrera dig till olika minnespositioner genom att använda piltangenterna på tangentbordet.

Ställ switchen SINGLE CLOCK/CONT CLOCK i läget SINGLE CLOCK och switchen INTERNAL CLOCK/EXTERNAL CLOCK i läget INTERNAL CLOCK.

Mata in kodsekvensen från *hemuppgift 2* i minnet med hjälp av terminalen. Ge RESET-vektorn värdet  $20_{16}$  då detta är programmets startadress.

Starta körningen genom att aktivera Reset-switchen på styrenheten och ge en klockpuls genom att trycka på knappen CLOCK (STEP CYCLE). Nollställ sedan Reset-switchen igen.

Stega igenom RESET- och FETCH-fasen klockcykel för klockcykel genom att använda knappen CLOCK (STEP CYCLE) tills du får första OP-koden, 0F, i indikatorfönstret på styrenheten. Du kan här bortse från vilket tillstånd, state, som visas på styrenheten.

Använd sedan knappen STEP INSTRUCTION för att exekvera en instruktion i taget och kör programmet instruktionsvis. Lägg märke till vilka värden som laddas i instruktionsregistret (IR), A-registret och programräknaren (PC).

Hur många gånger exekveras den inre programsnurran innan du utför BRA-instruktionen? \_\_\_\_\_

Ändra nu BNE-instruktionen till en BCC-instruktion och undersök hur många varv den inre programsnurran utförs nu: \_\_\_\_\_

Välj nu en BMI-instruktion i stället.

Hur många varv blir det nu? \_\_\_\_\_

Jämför svaren ovan med motsvarande svar i hemuppgiften!

Diskutera med en handledare.

**Slut på uppgift 2**

## **Uppgift 3 (endast hemuppgift)**

### **Anslutning av laborationskortet ML4 via portar till FLEX-datorn.**

Läs om ML4 devices under IO Simulator i hjälpsystemet för ETERM. Studera även Teknisk Beskrivningen av ML4 som du hittar via hemsidan. Studera speciellt de första åtta sidorna av beskrivningen

**Hemuppgift 3-1**

Då ML4 är bestyckad med bl.a. strömbrytare och lysdioder som vi önskar ansluta måste FLEX-datorn ha en inport och en utport. Som inport för anslutning av strömbrytarna används en "Data Source"-modul (DS2) och som utport för anslutning av lysdioderna en registermodul (Reg8).

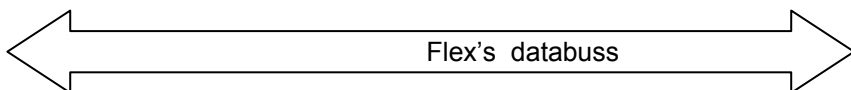
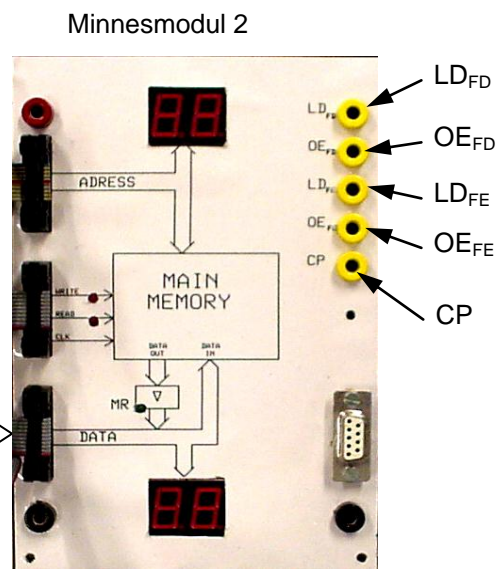
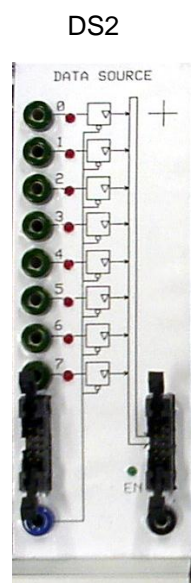
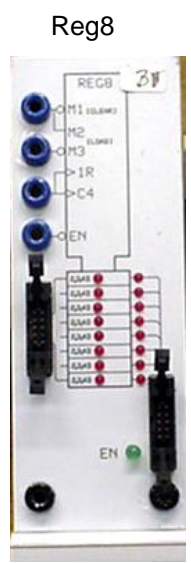
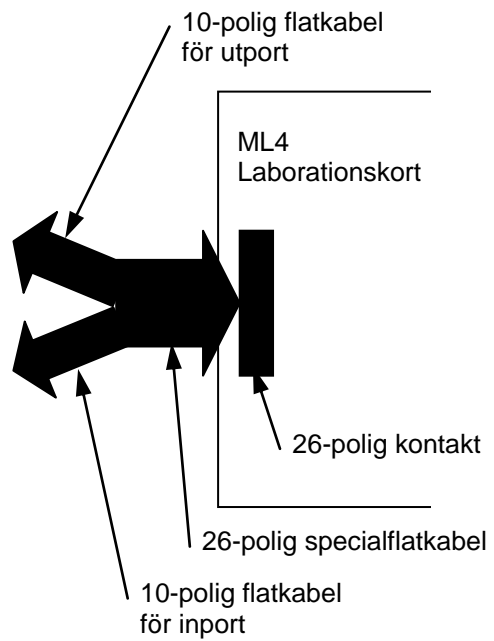
För att ansluta portarna till FLEX-datorn krävs dessutom adressavkodningslogik som bildar Chip Select-signalerna. Minnesmodul 2 innehåller nödvändig adressavkodningslogik chip select-signalerna för anslutning av in- och utportar (I/O-portar). Studera kapitel 25 i arbetsboken och manualbladen för modulerna som används.

- Moduler**  
 FLEX dataväg  
 Minnesmodul 2  
 FLEX styrenhet  
 Reg8  
 DS2

En specialflatkabel används för att ansluta ML4 till in- och utporten. Kabeln har en 26-polig kontakt som ansluts till ML4 och två 10-poliga kontakter som ansluts till in- respektive utporten på FLEX-datorn (ansluts till modulerna DS2 och Reg8).

**Hemuppgift 3-2**

I bilden nedan skall du rita in hur inporten och utporten är kopplade till databussen. Du skall också rita in hur klocksignalen (CP), "Chip select"-signalerna ( $OE_{xx}$  och  $LD_{xx}$ ) samt specialflatkabeln till ML4 är kopplade. Observera att inporten skall placeras på adress  $FD_{16}$  och utporten på adress  $FE_{16}$ .



Inkoppling av portar

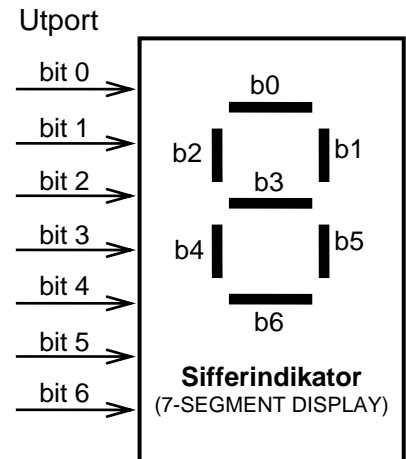
**Slut på uppgift 3**



## Uppgift 4

### Sju-segment sifferindikator

I Ext-1 beskrivs en sju-segments sifferindikator, som används för att visa decimala siffror. En sådan sifferindikator finns på laborationskortet ML4 och sådana indikatorer kan även anslutas till den simulerade FLEX-datorn i ETERM via utportarna. Hur bitarna på utportarna är kopplade till segmenten framgår av figuren till höger. Bit 7 är ansluten till en "decimalpunkt" nere till höger om de övriga segmenten, men den är inte inritad eftersom vi inte kommer att använda den. För att ett segment skall vara synligt krävs att motsvarande bit har värdet 1.



### Hemuppgift 4-1

Komplettera tabellen nedan med binär och hexadecimal sju-segmentkod för alla decimala siffror.

Decimalsiffr	Sjusegmentkod				
	Binär kod	Utseende	Binär kod	Hex-kod	Förklaring
0	0000	0	01110111	77	Nolla
1	0001	:			Etta
2	0010	2			Tvåa
3	0011	3			Trea
4	0100	4			Fyra
5	0101	5			Fem
6	0110	6			Sex
7	0111	7			Sju
8	1000	8			Åtta
9	1001	9			Nia

Skriv ett program i assemblerspråk för FLEX-processorn enligt flödesplanen till höger som läser inport  $FD_{16}$ , maskar bort bit 4-7 och sedan visar den decimala siffran som motsvarar det binära talet i bit 0-3 på sifferindikatorn på utport  $FE_{16}$ . Placera tabellen med sjusegmentkod direkt efter programmet. Startadressen för programmet skall vara 0. Översätt programmet till maskinkod nedan till höger.

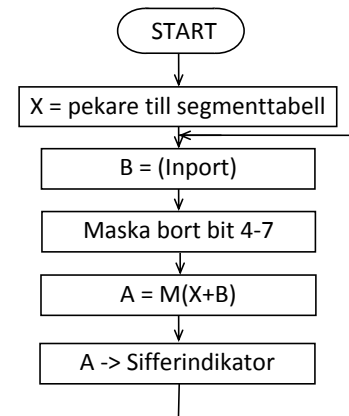
Program:

Assemblerspråk

Adress(Hex)  
00

Maskinkod

### Hemuppgift 4-2



- Starta ETERM och skriv in programtexten ovan i filen *hem4-2*.
- Assemblera filen och kontrollera i listfilen att maskinkoden ovan är korrekt.
- Ladda maskinkoden i simulatorns minne.
- Anslut DIPSWITCH och sifferindikatorn (7-SEGMENT) till sina portar.
- Tänk igenom hur programmet fungerar och börja med att stega igenom det. Ge olika värden  $[0, 1001_2]$  på DIPSWITCH och kontrollera att korrekta siffror visas. Rätta vid behov!
- Kör sedan programmet i läget "*Run Slow*". Vad händer om man ger värden  $> 1001_2$ .

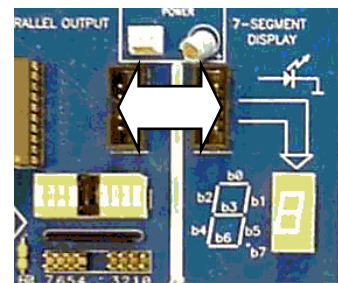
### Laborationsuppgift 4a:

Programmet skall nu testas i den verkliga FLEX-datorn med ML4. Anslut därför en 10-polig flatkabel till sifferindikatorn enligt figuren om detta inte är gjort tidigare.

Mata in programmet från *hemuppgift 4-2* i minnet och ge RESET-vektorn värdet 0.

Stega sedan igenom programmet instruktionsvis så att du ser att det fungerar.

Starta därefter programmet (RESET + CONT CLOCK) och ställ in olika värden  $[0, 9]$  på strömbrytarna och kontrollera att korrekt siffra visas på sifferindikatorn.



**Hemuppgift 4-3**

Spara filen *hem4-2* som *hem4-3* och ändra programmet så att det kontinuerligt skriver ut siffrorna [0,9] enligt flödesplanen till höger. Testa detta i simulatorn och förvissa dig om att det fungerar korrekt. Kör först "Run Slow" och sedan "Run Fast" i simulatorn. Vilken skillnad upplever du och varför?

---



---

**Hemuppgift 4-4**

Spara filen *hem4-3* som *hem4-4* och lägg till en fördröjning i programmet efter visningen av en siffra på sifferindikatorn, enligt flödesplanen nedan till höger. Varje siffra kommer då att synas längre tid på sifferindikatorn.

Du måste alltså ha någon form av fördröjningsslinga som utförs efter det att en siffra har visats på sifferindikatorn. Placera denna fördröjningsslinga i en subrutin så att den enkelt kan modifieras senare.

Exempel på ett par fördröjningsslingor:

```

DELAY1  NOP          DELAY2  PSHA
        RTS          LOOP    LDAA  #Värde (testa ut ett lämpligt värde)
        PULA          BNE    LOOP
        RTS          PULA
    
```

Den enklaste fördröjningsslingan ovan är bra att använda när du testar och kör STEP i simulatorn. Den andra är lämplig när du kör "RUN FAST".

Placera din fördröjningsslinga med start på adress 80<sub>16</sub> i minnet (ett nytt ORG). Detta medför att det blir enklare att modifiera programmet i FLEX-datorn i labbet.

Eftersom stacken används vid subrutinanrop måste du också sätta stackpekaren SP till ett lämpligt värde t ex BOS = FC<sub>16</sub> i början av programmet.

För att anropa (= hoppa till) fördröjningssubrutinen bör du använda exempelvis instruktionen JSR DELAY i huvudprogrammet. Tänk på att din DELAY-rutin inte får förstöra registerinnehåll för huvudprogrammet.

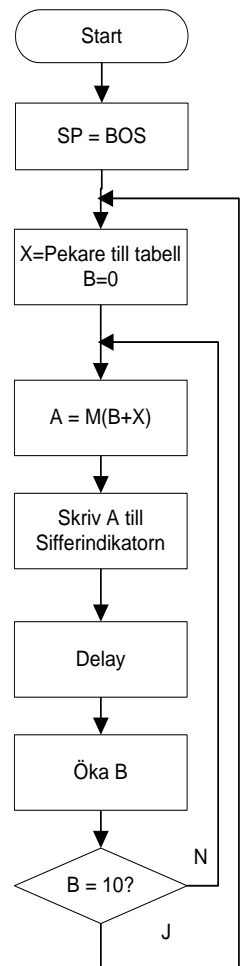
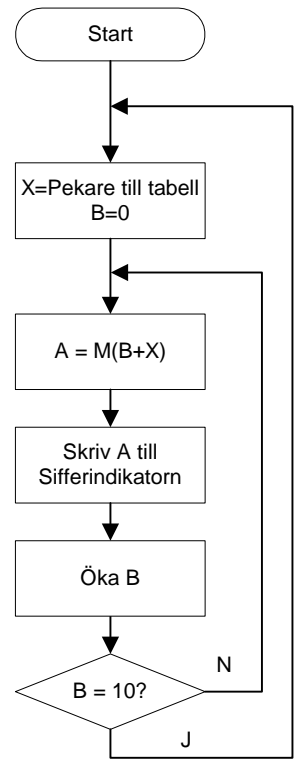
Testa programmet i simulatorn och förvissa dig om att det fungerar korrekt. Ändra din delayrutin så att varje siffra visas i ungefär 500 ms när du kör "Run Fast". Det tar då ca 5 s att visa alla siffrorna [0,9]. Förmodligen upptäcker du att fördröjningen i DELAY2 inte kan göras tillräckligt lång när du kör "RUN FAST". Hur löser du detta?

**Laborationsuppgift 4b:**

Skriv in programmet från *hemuppgift 4-4* i FLEX-datorns minne och stega det först. Använd DELAY1 ovan.

Starta det sedan så att siffrorna [0,9] kontinuerligt visas på ML4. Använd nu den ursprungliga DELAY2 som är given ovan.

Justera DELAY2 så att varje siffra lyser ungefär i 300 ms. Det tar då ca 3 s att visa alla siffrorna [0,9].

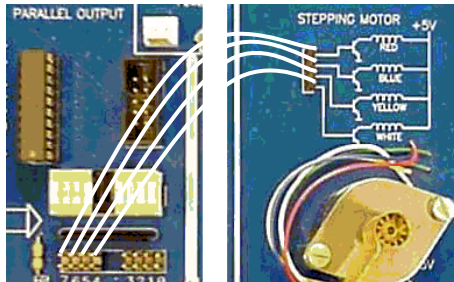


**Slut på uppgift 4**

## Uppgift 5

### Styrning av stegmotor

ML4 innehåller en stegmotor som du skall styra. Bilden och tabellen nedan visar anslutningarna mellan utporten och stegmotorn på ML4.



Anslutning av faser:	
RÖD	bit 7
BLÅ	bit 6
GUL	bit 5
VIT	bit 4

Stegmotorns axel ställer in sig i ett bestämt läge om ström leds (styrbit = 1) genom två av de fyra spolarna (RED, BLUE, YELLOW och WHITE) som visas i bilden "STEPPING MOTOR" ovan. Om sedan strömmen stängs av (styrbit = 0) i den ena spolen och istället leds genom en av de andra så kommer axeln att vrida sig en liten vinkel åt ena hållet, beroende på vilken av de andra spolarna som aktiveras. Genom att aktivera och stänga av strömmen i spolarna i tillståndsordningen 0, 1, 2, 3, 0, ... enligt tabellen nedan kan man få axeln att rotera åt ena hållet. Byter man till motsatt ordning byter man också rotationsriktning.

	b7	b6	b5	b4	HEX
Tillstånd 0	1	0	0	1	
Tillstånd 1	1	0	1	0	
Tillstånd 2	0	1	1	0	
Tillstånd 3	0	1	0	1	

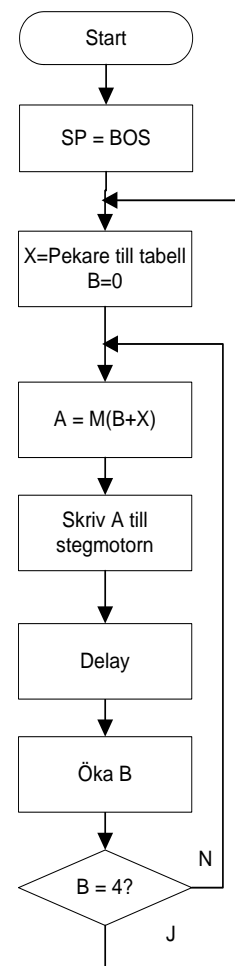
### Hemuppgift 5-1

Skriv ett program enligt flödesplanen till höger som får motorn att rotera med en jämn hastighet. Utgå från programmet i *hemuppgift 4-4*. Spara som ny fil (*hem5-1*) och ändra i denna fil. Använd startadress 0.

Editera klart, assemblera, ladda ner och testa ditt program i simulatort. Eftersom stegmotorn inte finns i simulatort ansluter du istället "HEXDISPLAY" till utporten. Kör programmet och justera DELAY så att du kan se varje tillstånd från tabellen ovan på HEXDISPLAY. Kontrollera att tillstånden visas i rätt ordning.

### Laborationsuppgift 5a:

Se till att stegmotorn är kopplad till utporten enligt figuren ovan. Använd sedan listfilen från *hemuppgift 5-1* och skriv in maskinkoden i minnet. Använd DELAY2 och justera den till ca 200 ms. Kör programmet och studera hur stegmotorn uppför sig.



**Hemuppgift 5-2**

För att kunna köra stegmotorn med variabel rotationshastighet ändrar du programmet från *hemuppgift 5-1* enligt flödesplanen till höger. Spara med nytt filnamn *hem5-2*.

I flödesplanen kompletteras det förra programmet med en läsning av strömbrytare anslutna till inporten. Läses 3 från strömbrytarna skall delay-rutinen utföras tre gånger. Se flödesplanen.

Det är meningslöst att fortlöpande skriva nya tillstånd till stegmotorn utan en tillräcklig fördröjning mellan tillstånden. Det skulle innebära att stegmotorn står still då den inte hinner reagera innan nästa stillstånd skrivs. Av denna anledning innehåller flödesplanen en kontroll att fördröjningsvärdet som ställs in på strömbrytarna, dvs. antalet gånger du anropar Delay är  $\neq 0$ .

Editera klart, assemblera, ladda ner och testa ditt program i simulatorn. Anslut "HEXDISPLAY" till utporten och "DIPSWITCH" till inporten. Kör programmet medan du ändrar strömbrytarna och observera hur ofta tillstånden byts på displayet.

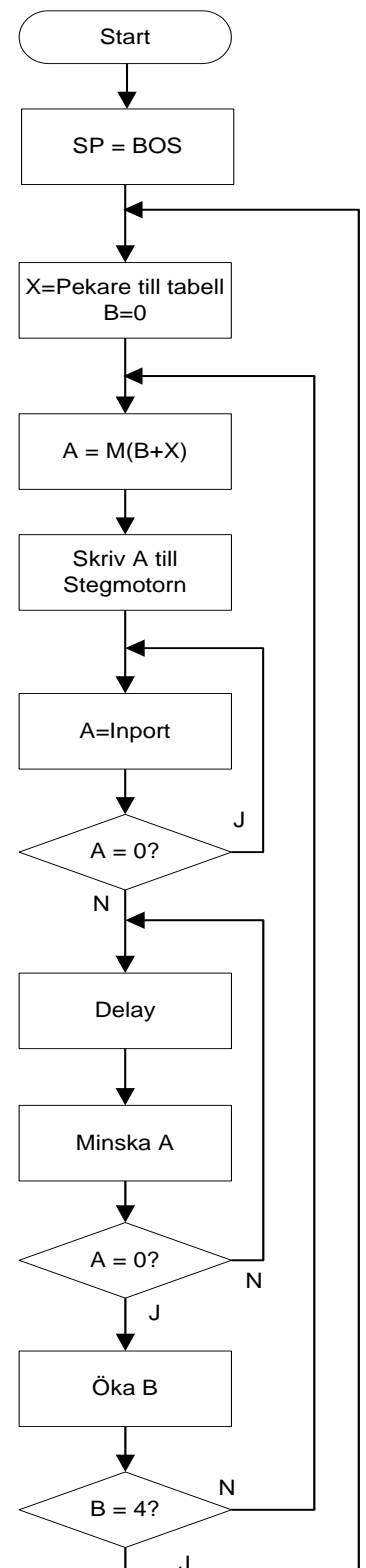
**Laborationsuppgift 5b:**

Använd listfilen från *hemuppgift 5-2* och skriv in maskinkoden i minnet.

Kör sedan programmet och studera hur stegmotorn uppför sig när du ställer in olika fördröjningsvärden på strömbrytarna.

Justera fördröjningen så att den blir mycket kort. Du har då större möjlighet att variera rotationshastigheten med strömbrytarna.

Ändra tabellen så att tillstånden kommer i omvänd ordning och kör programmet på nytt medan du studerar stegmotorn.



**Slut på uppgift 5**

## Uppgift 6

### Rinnande ljus

**Hemuppgift 6**

Skriv ett program som gör rinnande ljus på utportens ( $FE_{16}$ ) lysdioder på ML4. Du skall styra ljuset med strömbrytarna på ML4 som skall vara anslutna till inporten ( $FD_{16}$ ).

Med rinnande ljus menas att lysdioderna på utporten skall tändas i följande ordning: Först tänds  $b_0$  för att sedan släckas när  $b_1$  tänds och när  $b_2$  tänds släcks  $b_1$  etc. På så sätt kommer ljuset att "rinna fram" från  $b_0$  till  $b_7$ . När  $b_7$  släcks tänds  $b_0$  på nytt.

Följande gäller:

- När bit 7 på strömbrytarna är ett (ON) skall lysdioderna visa rinnande ljus om och om igen.
- När bit 7 på strömbrytarna är nollställd (OFF) skall den lysdiod som är tänd lysa färdigt och sedan skall lysdioderna släckas.
- När bit 7 på strömbrytarna på nytt ettställs (ON) skall rinnande ljus starta genom att börja om från början med bit 0.

Programmet skall placeras med start på adress  $40_{16}$  i minnet.

När en lysdiod tänds så skall den lysa ca 125 ms. Det innebär att det tar ca en sekund att göra rinnande ljus *en gång*. Du måste därför fördröja programmet i 125 ms varje gång du tänt *en* lysdiod via utporten. Använd subrutinen DELAY2 som fördröjning.

### Rita först en flödesplan till vänster på nästa sida!

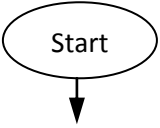
Skriv sedan in programmet, assemblera, ladda och testa det i simulatorn med DIPSWITCH ansluten till inport  $FD_{16}$  och LED ansluten till utport  $FE_{16}$ . Spara programmet i filen *hem6*.

Då programmet laddas i minnet i simulatorn får inte RESET-vektorn något nytt värde. Följande programrader visar hur man kan placera startadressen i RESET-vektorn när programmet laddas.

Lägg till följande programrader i slutet av källfilen:

```
ORG    $FF
FCB    $40      (Om programmet har startadressen =  $40_{16}$ )
```

**Skriv programmet och maskinkoden till höger om flödesplanen på nästa sida. Ange även vilka adresser de olika instruktionerna är placerade på.**

FLÖDESPLAN	Adress	Maskinkod	Assemblerprogram
	40		

**Laborationsuppgift 6:**

Mata in programmet i FLEX-datorn i labbet och verifiera att det fungerar korrekt. Observera att du inte vet hur snabb FLEX-datorn är, så du måste troligen justera fördröjningsslingan.

**Slut på uppgift 6**

---

Lämna din plats med samma uppkopplingar som när du började.

Nu när du är klar med denna laboration har du

- skrivit assemblerprogram som *du* översatt till maskinprogram och laddat i datorns minne.
- styrt ett antal olika applikationer med *dina* program som körs på *din* dator.