
Laboration nr 4 behandlar

Assemblerprogrammering

Följande uppgifter ur "Arbetsbok för DigiFlisp" ska vara utförda som förberedelse för laborationen. Du ska på begäran av laborationshandledare redogöra för dessa.

Uppg.	16.9	16.10 16.11	16.16	16.20
Sign.				

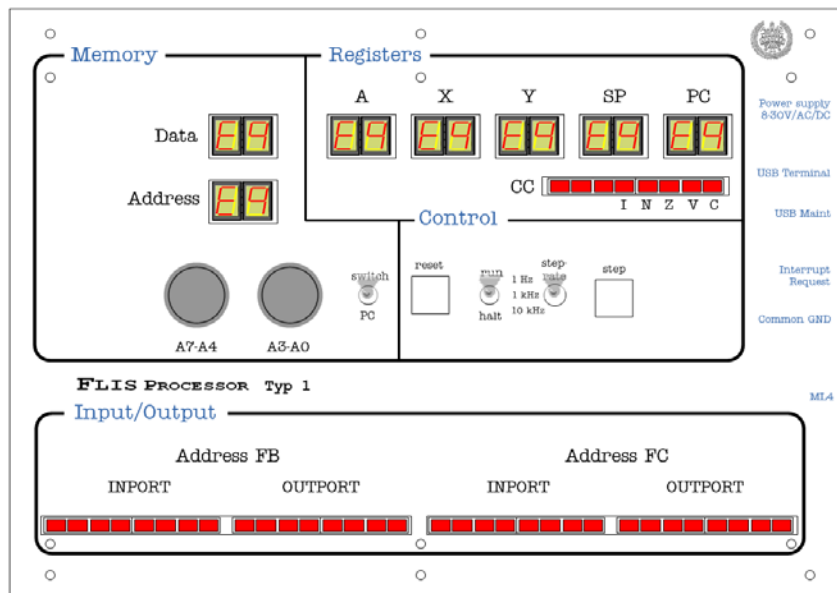
Följande laborationsuppgifter skall redovisas för en handledare för godkännande under laborationen.

Laborations- uppgift	4.1	4.2	4.3	4.4
Sign.				

Beskrivning av laborationssystemet

Laborationssystemets panel är indelad i fyra sektioner: *Memory*, *Registers*, *Control* och *Input/Output*.

- **Memory:** Indikatorerna för Data visar minnesinnehållet på den adress, visningsadressen, som anges i Address-indikatorerna. Visningsadressen kan väljas med omkopplaren switch/PC.
 - I läge switch, väljs visningsadressen med de båda vridomkopplarna A7-A4 respektive A3-A0.
 - I läge PC kommer visningsadressen automatiskt att uppdateras med programräknarens värde, datafönstret kommer då vanligtvis att visa aktuell operationskod.
- **Registers**, visar innehållet i FLIS-processorns register.
- **Control**, här manövreras FLIS-processorn.
 - Tryckomkopplare reset, processorn utför ett återstartsförlopp.
 - Omkopplare run/halt, i läge run exekverar processorn kontinuerligt och exekveringshastigheten kan väljas med omkopplaren *steprate* (1, 1000 eller 10 000 instruktioner per sekund).
 - I läge halt utförs en instruktion då du trycker ned omkopplaren step.
- **Input/Output** visar de värden som för tillfället finns i FLIS-processorns portar.



Nedladdning av program

ETERM har inbyggd funktion för att underlätta nedladdning av program och data till laborationsdatorn. Funktionen filtrerar en fil av typen `.hwflisp` (skapas då du assemblerar din källtext) och skickar `setMemory-` och `setRegister-` kommandon till laborationsdatorn.

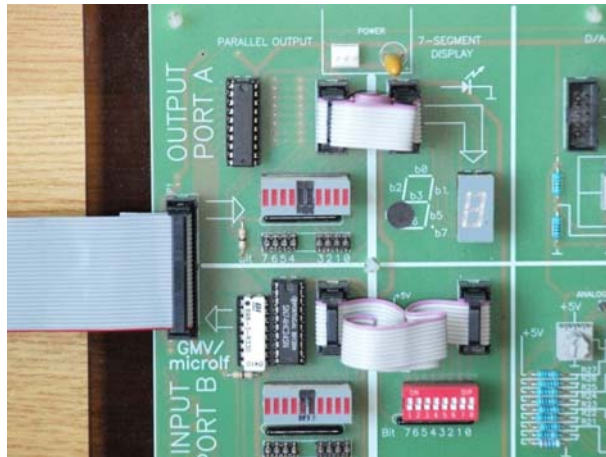
- Välj Debug | Terminal och sedan den COM-port som anvisats av laborationshandledare. Ett terminalfönster (blå färg) öppnas nu.
- Kontrollera att laborationsdatorn står i läge halt.
- Gör reset på laborationsdatorn.
- Placera markören i terminalfönstret och högerklicka, välj Load New och sedan fil för nedladdning.

Laborationsuppgift 4.1.**Sjusegmentsindikator**

I denna uppgift ska du testa din lösning på uppgift 16.9 (DisplaySegE), i arbetsboken. Du ska tidigare ha provat den i simulator så du vet att ditt program beter sig som det ska.

Du måste dock göra ett litet tillägg till din tidigare lösning: för att programräknaren (PC) i laborationsdatorn ska få rätt värde (20_{16}) från början ska du lägga till assemblerdirektiv som placerar programmets startadress i RESET-vektorn.

- Laborationsdatorn ska vara ansluten till laborationskort ML4 via en 26-polig flatkabel.
- På ML4 ska sektionen DIPSWITCH input vara ansluten (10-polig flatkabel) till INPUT-sektionen.
- Sektionen 7-SEGMENT DISPLAY ansluts till OUTPUT-sektionen.



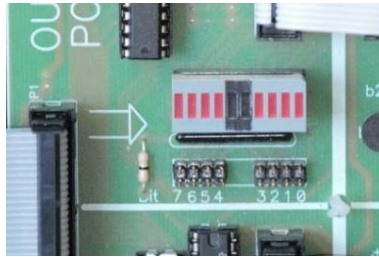
- Assemblera DisplaySegE.sflisp, det ska inte finnas några felmeddelanden.
- Kontrollera att laborationsdatorn står i läge halt.
- Gör reset på laborationsdatorn.
- Placera markören i terminalfönstret och högerklicka, välj Load New och sedan filen DisplaySegE.hwflisp för nedladdning.
- Öppna listfilen (DisplaySegE.lst) i texteditorn.
- Sätt visningsadressen på laborationsdatorn i läge PC.
- Tryck en gång på step-omkopplaren och observera hur PC uppdateras med programmets startadress.
- Läs av Data och se vad som finns i minnet på denna adress.
- Fortsätt med att utföra programmet instruktionsvis (step), följ med i listfilen, gör detta ett helt varv i programmet, dvs. tills PC på nytt får värdet 22_{16} .

Laborationsuppgift 4.2.**Realtidsegenskaper, rinnande ljus.**

Inför denna uppgift ska du ha utfört och testat uppgift 16.11 (RunDiodeDelay), i arbetsboken. Du ska alltså tidigare ha provat den i simulator så du vet att ditt program beter sig som det ska.

Du måste även här lägga till assemblerdirektiv som placerar programmets startadress i RESET-vektorn för att laborationsdatorn ska starta korrekt.

- Laborationsdatorn ska vara ansluten till laborationskort ML4 via en 26-polig flatkabel på samma sätt som i föregående uppgift.
- I denna uppgift använde vi ljusdiodrampen på ML4's OUTPUT-sektion. Vi använder inte 7-SEGMENT-DISPLAY denna gång men du kan ändå låta den 10-poliga flatkabeln sitta kvar (den gör ingen skada).



- Assemblera `RunDiodeDelay.sflisp`, det ska inte finnas några felmeddelanden.
 - Kontrollera att laborationsdatorn står i läge halt.
 - Gör reset på laborationsdatorn.
 - Placera markören i terminalfönstret och högerklicka, välj **Load New** och sedan filen `RunDiodeDelay.hwflisp` för nedladdning.
 - Starta programmet genom att ställa omkopplare i run-läge.
 - Variera laborationsdatorns exekveringshastighet genom att ändra `steprate` och observera skillnader hos ljusdiodrampen.
-

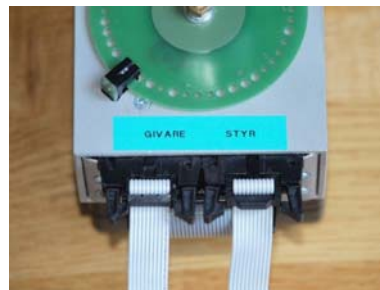
Laborationsuppgift 4.3.**Mekatronik, borra ett hål**

Inför denna uppgift ska du ha utfört och testat uppgift 16.16 (DrillHole), i arbetsboken. Du ska ha provat programmet i simulator så du vet att det beter sig som det ska. Lägg också till assemblerdirektiv som placerar programmets startadress i RESET-vektorn så att laborationsdatorn ska starta korrekt.



- Laborationsdatorn ska vara ansluten till laborationskort ML4 via en 26-polig flatkabel på samma sätt som i föregående uppgift.
 - Sektionen INPUT ska ansluten till sektionen DIPSWITCH. Omkopplaren används för att starta och stoppa bormaskinprogrammet.
 - Sektion OUTPUT används inte i denna uppgift.

- Bormaskinen ansluts till laborationsdatorn via två stycken 10-poliga flatkabler:
 - Laborationssystemets port OUTPORT FC ansluts till bormaskinens port STYR.
 - Laborationssystemets port INPORT FB ansluts till bormaskinens port GIVARE.



- Utför programmet instruktionsvis (step), kontrollera att bormaskinen utför de enskilda styrkommandona korrekt.

Nivåerna hos bormaskinens styr- och status- signaler kan avläsas av dioderna på bormaskinens ena sida:



- Ställ steprate-omkopplaren i mellanläget och låt laborationsdatorn utföra programmet (run).
- Modifiera eventuellt Delay-rutinen så att rätt beteende uppnås.

Laborationsuppgift 4.4.

Nivåtrigget avbrott från ett tangentbord

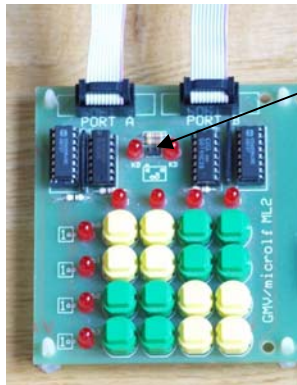
Uppgift 16.20 (sidan 185 i arbetsboken) i upplagan har figurerna fallit bort. En ersättningssida finns på kursens hemsida.

Observera att denna uppgift ej är identiskt densamma som i arbetsboken. Subrutinen `GetKey` ska här göras i en betydligt enklare variant.

Subrutinen `GetKey` behöver nu inte returnera korrekt tangentkod. Det räcker med att avbrottet detekteras och avbrottsrutinen utförs. I avbrottsrutinen ska variabeln `kbdBuffer` ökas med ett för varje tangentnedtryckning.

- Tangentbordet ansluts till laborationsdatoren via två stycken 10-poliga flatkablar:
 - Laborationssystemets port `OUTPORT FC` ansluts till tangentbordets `PORT A`.
 - Laborationssystemets port `INPORT FB` ansluts till tangentbordets `PORT B`.
 - Laborationssystemets `Interrupt Request` (banankontakt på lådans kortsida) ska också anslutas med en enpolig kabel till tangentbordets `KEY DOWN`.

Uppkoppling:



Stift KEY DOWN ansluts till laborationssystemets `Interrupt Request`

- Ställ visningsadress (läge `switch`) på adress 0, dvs. variabeln `kbdBuffer`.
- Starta programmet, kontrollera att den globala variabeln uppdateras vid tangentnedtryckning.