# LABORATION B1

# LÄRARHANDLEDNING

# BUSSMÄTNINGAR

# STUDIUM AV INBÖRDES

# TIDSBEROENDE

# SIGNALER



LPMB1 v1.1

# LABORATION B1

# LÄRARHANDLEDNING

# BUSSMÄTNINGAR.

# STUDIUM AV INBÖRDES TIDSBEROENDE SIGNALER

Denna laboration behandlar begrepp som läs-, skriv- och bussoberoende cykler. Vidare behandlas "timing" och "tristate". Momenten är grundläggande för den påföljande laborationsserien och bör ägnas stor omsorg.

Observera att laborationen kräver förberedelser (hemuppgifter) gjorda innan laborationstillfället

Följande häften ska Du vid sidan av detta ha tillgängliga vid laborationsplatsen:

Hårdvarubeskrivning för **MD09, ML1, ML2, ML3**
Referensmanual för **db09** MONITOR DEBUGGER
MOTOROLA MC6809 datablad

Du bör också ha bekantat Dig med dessa häften på ett sådant sätt att Du snabbt kan hitta upplysningar i dom.

Text och kommentarer till läraren är här skriven i *kursiv stil*

# INLEDNING

Denna laboration behandlar **MD09**:s data- och adressbussar som skall undersökas när processorn exekverar enkla testprogram. Först kommer olika instruktioners busscykler att undersökas, deras nivåer, antal cykler mm. Sedan kommer bussarnas "tristate-nivåer" att undersökas. Slutligen kommer en busscykel att studeras mer ingående, där tidmätningar kommer at utföras mm.

# BUSSCYKLER

Processorn är i normal drift i tillstånd

1) **FETCH** (hämtar in en ny instruktion)

eller

2) **EXECUTE** (exekverar en inhämtad instruktion)

Båda dessa tillstånd tar en till flera processorcykler (E-cykler). Dessa cykler skall nu undersökas mha nedanstående testprogram.

```
Adr     Data              Beskrivning

c000    b7  loop  sta   $e006
c001    e0
c002    06
c003    20        bra   loop
c004    fb
c005
```





## Hemuppgift 1

Studera databladen och besvara följande frågor

Hur många cykler används för "sta"-instruktionen?_____

Hur många cykler används för "bra"-instruktionen?_____

"sta"-instruktionen består av 3 bytes och "bra"-instruktionen 2. Hur många cykler tar då respektive instruktions FETCH-fas?

_____

Hur lång tid motsvarar det?_____

Hur många cykler tar respektive instruktions EXECUTE-fas?_____

Vad kan man förvänta sig se på bussarna under EXECUTE-faserna av "sta"-instruktionen och "bra"-instruktionen

på adressbussen? _____

på databussen? _____

övriga processorsignaler? _____

*TILL LÄRAREN:*

*"sta" tar 5 E-cykler och "bra" 3. Se "Programming aid" eller datablad.*
*Detta motsvarar 5us respektive 3us (1MHz E-klocka)*

*FETCH-faserna bör ta 3 respektive 2 E-cykler (och gör så) då det behövs 3 respektive 2 minnesacesser för att läsa in instruktionerna.*

*EXECUTE-faserna tar således 2 resp 1 E-cykel.*

*"sta"-instruktionens EXECUTE-fas består av 2 E-cykler och under EN av dessa bör man förvänta sig att adressbussen har värdet $e006, att databussen har register A:s värde och att skrivsignalen går låg (R/W = 0). Detta händer under EXECUTE-fasens andra cykel.*

*Under den första E-cykeln som är bussoberoende, kommer adressbussen att ha värdet $ffff vilket visas i "Cycle-by-cycle Performance" i databladen. En gissning är att processorn "slår samman" hög och låg del av den nyss inlästa adressen ($e006).*

*Observera att när adressbussen har värdet $ffff kommer adressavkodningen på* **MD09** *att generera CS för PROM'et som i sin tur lägger ut $00 på databussen (Low byte av RESET-vektorn).*

*Under "bra"-instruktionens EXECUTE-fas, som är bussoberoende adderas offset till PC. Adressbussen kommer att få värdet $ffff och databussen $00 som beskrivet ovan.*

## Hemuppgift 2

Studera databladen (Cycle-by-cycle flowchart) och skriv i tabellform adress- och databussarnas värden för varje E-cykel när testprogrammet exekveras. Använd tabell 1.





*TILL LÄRAREN*

| Cykel | Adress buss(hex) | Data buss(hex) | Beskrivning |
|---|---|---|---|
| 1 | c000 | b7 | OP-Fetch |
| 2 | c001 | e0 | Adr Hi Fetch |
| 3 | c002 | 06 | Adr Lo Fetch |
| 4 | ffff | 00 | Bussoberoende |
| 5 | e006 | [Reg A] | R/W = 0 |
| 6 | c003 | 20 | OP-Fetch |
| 7 | c004 | fb | Offset Fetch |
| 8 | ffff | 00 | Bussoberoende |
| 9 | c000 | b7 | |

**Tabell 1**

Tabell 2 nedan visar E-klockan, R/$\overline{W}$ och D7-D0 under ett antal klockcykler. E-klockan och R/$\overline{W}$-signalen är inritad för ett antal cykler. Tabellen skall visa databussens och R/$\overline{W}$-signalens värde under exekveringen av ovanstående testprogram.

## Hemuppgift 3

Rita in resterande förlopp av R/$\overline{W}$-signalens värde i tabell 2 så att två skrivpulser uppträder (drygt ett helt förlopp av testprogrammet visas). Utgå från hemuppgift 2 och skriv in på nedresta raden in de hex-siffror som överförs på databussen under respektive cykel. Ta de värden du är säker på. Utgå från att register A har värdet $2a när programmet exekveras. Rita sedan in hur varje bit av databussen kommer att se ut.

## Laborationsuppgift 1

Mät databussen med ocilloskop och jämför med dina antaganden i tabell 2. Lämpligt mätställe är anslutning K7 på **MD09**. Studera **MD09**:s hårdvarubeskrivning för pinplacering och läs avsnittet "Externbuss". "Trigga" ocilloskopet på R/W-signalen.
Stämmer dina antaganden? _____

Vad överförs på databussen i cykel 4 och 8? _____

_____

_____

*TILL LÄRAREN:*

*Under cykel 4 och 8 har adressbussen värdet $ffff vilket ger "Chip Select" till PROM och låg byte av RESET-vektorn läggs därför ut på databussen.*





| | CYKEL | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Signal | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| E | /‾\ | /‾\ | /‾\ | /‾\ | /‾\ | /‾\ | /‾\ | /‾\ | /‾\ |
| R/$\overline{W}$ | _/‾ | ‾ | ‾ | ‾ | ‾ | ‾ | ‾ | _/‾ | ‾ |
| D7 | | | /‾\ | _/ | /‾\ | ‾ | _/ | ‾ | ‾ |
| D6 | | | /‾\ | _/ | ‾ | /‾\ | ‾ | ‾ | ‾ |
| D5 | | | ‾ | ‾ | /‾\ | ‾ | ‾ | ‾ | /‾\ |
| D4 | | | /‾\ | _/ | /‾\ | _/ | ‾ | ‾ | ‾ |
| D3 | | _/ | /‾\ | _/ | ‾ | ‾ | ‾ | ‾ | /‾\ |
| D2 | | | ‾ | ‾ | /‾\ | _/ | /‾\ | _/ | ‾ |
| D1 | | _/ | /‾\ | _/ | /‾\ | _/ | /‾\ | _/ | /‾\ |
| D0 | | | /‾\ | _/ | /‾\ | ‾ | ‾ | ‾ | ‾ |
| Hex | 2a | 20 | fb | 00 | b7 | e0 | 06 | 00 | 2a |

**Tabell 2.**

## Hemuppgift 4

Tabell 3 visar E-klockan, R/$\overline{W}$ och A15-A0. Rita in resterande förlopp av R/$\overline{W}$-signalens värde så att två skrivpulser uppträder. Utgående från hemuppgift 2 skriv in på nedersta raden in de hex-siffror som adressbussen har under respektive cykel. Ta de värden du är säker på. Rita sedan in hur varje bit av adressbussen kommer att se ut.

## Laborationsuppgift 2

Mät adressbussens nivåer med oscilloskop och kontrollera dina antaganden. Lämpligt mätställe är anslutning K7 på **MD09**. Studera **MD09**:s hårdvarubeskrivning för pinnplacering.

Stämmer dina antaganden? _____

Hur förklaras adressbussens värde under cykel 4 och 7? _____

_____

*TILL LÄRAREN*

*Då cykel 4 och 7 är bussoberoende kommer adressbussen att få värdet $ffff (se cycle-by-cycle flowchart i databladen)*

# "TRISTATE"-NIVÅER

Nu skall databussens "tri state"-nivåer studeras. Databussen används för att överföra data mellan processor och - RAM, ROM, periferikretsar osv. Bara EN av dessa enheter får lägga ut data på databussen under en klockcykel, annars kommer det att uppstå "busskrock". Bussbufferterna internt i processor, RAM, ROM och periferikretsarna får därför ej "driva" eller belasta bussen då enheten inte skall lägga ut data på databussen. Man säger då att bussbufferterna ligger i tristate.

För att kunna mäta "tristate"-nivån på tex databussen är det lämpligt att belasta bussen något, annars kommer "tristate"-nivån att se ut som en hög nivå.

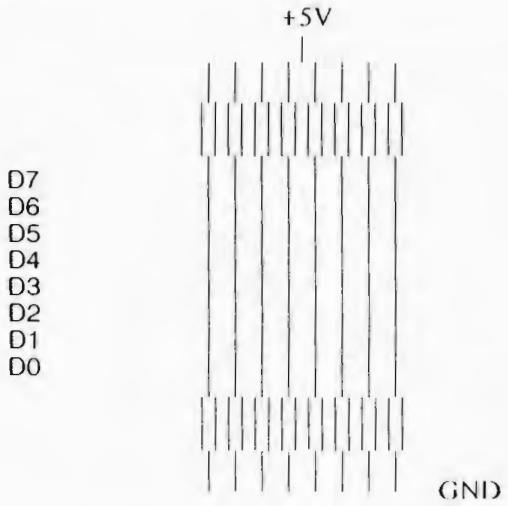




| | CYKEL | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Signal | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| E | / | \ / | \ / | \ / | \ / | \ / | \ / | \ / | \ / |
| R/$\overline{W}$ | \ | / | | | | | | | \ / |
| A15 | | | | | | | | | |
| A14 | | | | | | | | | |
| A13 | | \ | | / | \ | | | / | |
| A12 | | | | / | \ | | | / | \ |
| A11 | | | | / | \ | | | / | \ |
| A10 | | | | / | \ | | | / | \ |
| A9 | | | | / | \ | | | / | \ |
| A8 | | | | / | \ | | | / | \ |
| A7 | | | | / | \ | | | / | \ |
| A6 | | | | / | \ | | | / | \ |
| A5 | | | | / | \ | | | / | \ |
| A4 | | | | / | \ | | | / | \ |
| A3 | | | | / | \ | | | / | \ |
| A2 | | \ | / | | \ | | | / | |
| A1 | | | \ | / | \ | | / | | |
| A0 | | / | \ | / | \ | / | \ | / | \ |
| HEX | e006 | c003 | c004 | ffff | c000 | c001 | c002 | ffff | e006 |

**Tabell 3**

Jfr tidigare mätningar. En bussbelastare vars schema visas nedan kommer att andvändas.




**Figur 1 Bussbelastare.**

## Laborationsuppgift 3

Koppla bort spänningen på **MD09**, byt ut databussbufferten (IC14) för expansionsporten och sätt bussbelastaren i IC14. Studera hårdvarumanualen för att lokalicera IC14. Spänningssätt **MD09** och lägg in följande testprogram på adress $c000

| Adr | Data | Beskrivning | | |
|---|---|---|---|---|
| c000 | b6 | loop | lda | $e006 |
| c001 | e0 | | | |
| c002 | 06 | | | |
| c003 | 20 | | bra | loop |
| c004 | fb | | | |



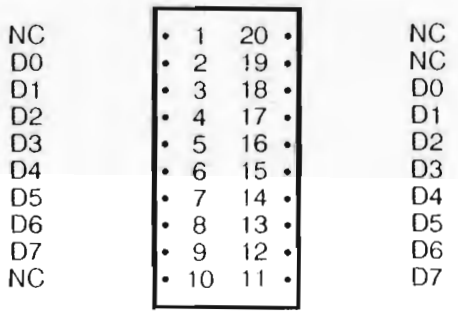


Obsevera att "sta"-instruktionen är utbytt mot en "lda"-instruktion i det tidigare använd exemplet. Detta innebär att man inte kan "trigga" oscilloskopet på R/$\overline{W}$-signalen utan man får andvända $\overline{IO0}$ som finns tillgänglig på K7. Denna signal är aktiv låg då adressbussen har värdet $e006. Se hårdvarumanualen för **MD09**.

**OBS! MÄTNINGARNA GÖRS NU DIREKT PÅ PROCESSORNS DATABUSS !**

**V A R F Ö R S I K T I G ! !**

Starta programmet och mät med oscilloskopet, använd $\overline{IO0}$ som "trigg"-signal. Relatera alla mätningar till E-klockan. Studera varje enskilld bit av databussen och jfr med tabell 2 du tidigare ifyllt. Lämpliga mätpunkter hittas på bussbelastaren. Se figur 2 för layout och pinplacering för bussbelastaren.




**Figur 2 Layout av bussbelastare.**

Bortsett från att databussen överför en annan operationskod i cykel 5 ($b6 i stället för $b7) vad mer skiljer?

_____

_____

Förklara databussens värde under första delen av varje E-cykel.

_____

_____

Förklara databussens värde under cykel 1.

_____

_____

*TILL LÄRAREN:*

*Databussen ligger i "tristate" under första fjärdedelen av en E-cykel. Detta är på grund av att alla* **MD09***:s CS-signaler är grindade med E + Q. Alltså: ingen krets får CS under denna period och håller därför sina bussbuffertar i "tristate"-läge.*

*Under cykel 1 görs en minnesacess på adress $e006 där det inte finns någon enhet som lägger ut något på adressbussen, alltså: "tristate" under en hel cykel.*

*Under cykel 4 och 8 har adressbussen värdet $ffff som i sin tur genererar $\overline{CS}$ för PROM och "low byte" av RESET-vektorn läggs ut på databussen.*





# TIDMÄTNINGAR

När man skall koppla tex. en minneskapsel till en processor måste man veta hur snabbt minne man måste ha. Detta är för att processorn förväntar sig att det finns giltig data (OP-koder och dyl.) på databussen vid en läsning av minnet.

Man säger att man har minnen med en viss "acesstid" (snabbhet). "Acesstiden" för ett minne är hur lång tid minnet tar på sig för att leverera giltig (stabil) data på databussen, räknat från det att minnet får en giltig (stabil) adressbuss. Uppgiften nedan går ut på och undersöka dessa tider.

## Laborationsuppgift 4.

Nu skall adress- och databussen examineras under en E-cykel för att slutligen se vilken "accesstid" minnet har. För att utföra mätningarna används samma testprogram som tidigare, nämligen:

| Adress | Data | Beskrivning | | |
|---|---|---|---|---|
| c000 | b6 | loop | lda | $e006 |
| c001 | e0 | | | |
| c002 | 06 | | | |
| c003 | 20 | | bra | loop |
| c004 | fb | | | |

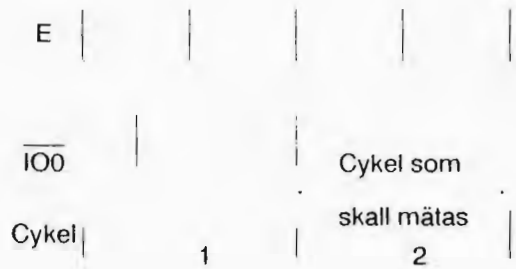
Lägg inn $f0 i register A **innan** programmet körs igång.
Ocilloskopet skall "triggas" på IO0 som tidigare. Den cykel som skall undersökas är andra cykeln som visas på ocilloskopet. Det är här som OP-koden

för "bra"-instruktionen överförs och adressbussen har värdet $c003. Jämför med tidigare mätningar du gjort.

Har du tillgång till tre probar och ett ocilloskop med "external trigg" kopplas $\overline{IO0}$ till "external trigg" kanal 1 till E-kolckan och kanal 2 används för att mäta på bussarna. Ställ in ocilloskopet för external trigg och tidbasen på 1us per ruta.

Har du bara tillgång till två probar kopplas kanal 1 till $\overline{IO0}$ och kanal två till E-klockan. Ställ in ocilloskopet för att trigga på kanal 1 och tidbasen på 1us per ruta.

Sätt nu tidbasen på 10X (gånger 10) och justera horisontal position för att visa endast den andra E-cykeln. Se figur 2. Du bör nu ha en horisontell upplösning på 100 ns.




**Figur 2 E-cykel som skall mätas**

Använd kanal två för de fortsatta mätningarna nedan. Kontrollera med jämna mellanrum att hela E-cykeln visas riktigt.





## Tidmätningar på adressbussen.

Hur snabbt levererar processorn giltig (stabil) adressbuss? Mätningarna görs på pinne A0 och A2. Mät tiden från giltig adressbuss till positiv flank av E-klockan.

Tid för A0: _____ ns

Tid för A2: _____ ns

Vilken tid anges i databladen för processorn? _____ ns

Skiljer tiderna sig åt? _____ och i så fall hur mycket? _____ ns

Betyder detta att processorn du mäter på är sämmre eller bättre än vad som specificeras i databladen?

_____

*TILL LÄRAREN:*

*Dessa mätningar är gjorda på ett system bestyckad med MC68A09, som kan köras på 1,5 MHz klockfrekvens. Tiderna nedan kan därför vara något olika de du mäter beroende på vilken processortyp du har.*

*Hur snabbt levererar processorn giltig adressbuss? Mätningarna görs på pinne A0 och A2. Mät tiden från giltig adressbuss till positiv flank av E-klockan.*

*Tid för A0: ca 420 ns*

*Tid för A2: ca 420 ns*
*Vilken tid anges i databladen för processorn?*

*Tiderna (Ident. Number +10 -4 -7 +2 +4) = 280 ns*
*(Ident Number är tagna från BUS TIMING CHARACTERISTICS i processorns datablad.)*

*Skiljer tiderna sig åt? Ja, tydligen.*
*och i så fall hur mycket? 140 ns*

*Betyder detta att processorn du mäter på är sämmre eller bättre än vad som specificeras i databladen? Mycket bättre. Detta beror delvis på att mätningarna är gjorda på MC68A09.*

## Tidmätningar på databussen.

Hur snabbt levererar minnet (RWM) giltig (stabil) databuss? Mätningarna görs på pinne D4 och D5. Mät tiden från giltig databuss till negativ flank av E-klockan.

Tid för D4: _____ ns

Tid för D5: _____ ns

Hur lång är denna tiden angiven (för en läscykel) i databladen för processorn?

_____ ns

Skiljer tiderna sig åt? _____ och i så fall hur mycket? _____ ns

Vad kan du säga om denna tidsskillnad? _____

_____

_____





Hur lång är "Usable access time" som är specad i datablad för processorn?

_____ ns

Utgående från de mätningar du gjort kan du nu beräkna minnets "access"-tid. Hur lång är denna?

_____ ns

*TILL LÄRAREN:*

*Hur snabbt levererar minnet (RWM) giltig databuss? Mätningarna görs på pinne D4 och D5. Mät tiden från giltig databuss till negativ flank av E-klockan.*

*Tid för D4: 700 ns*

*Tid för D5: 650 ns*

*Hur lång är denna tiden angiven (för en läscykel) i databladen för processorn? (Ident. Number 17) 80 ns (Processorn kräver stabil databuss minst 80 ns före negativ flank av E.)*

*Skiljer tiderna sig åt? Ja*
*och i så fall hur mycket? 650-80 = 570 ns*

*Vad kan du säga om denna tidsskillnad?* **MD09** *är bestyckad med ett RWM som är mycket snabbare än nödvändigt.*

*Hur lång är "Usable access time" som är specad i datablad för processorn? (Ident. Number 29) 695 ns*

*Utgående från de mätningar du gjort kan du nu beräkna minnets "access"-tid. Hur lång är denna?*
*Stabil adressbuss blev uppmätt till 420 + 500 = 920 ns före negativ flank av E.*

*Stabil databuss blev upmätt till 150 + 500 = 650 ns före negativ flank av E.*
*Detta innebär att minnets accesstid är 920-650 = 270 ns.*
*Detta kan verka förvirrande då man studerar kapsen (IC2) och upptäcker att det sitter (vanligen) ett RWM med en accesstid på 150 ns.*

*Orsaken är att RWM:et får Chip Select förhållandevis sent, så det är alltså denna signal som ger fördröjningen.*