Lösningsförslag/Facit till Dugga 3 i Mikrodatorteknik 4p

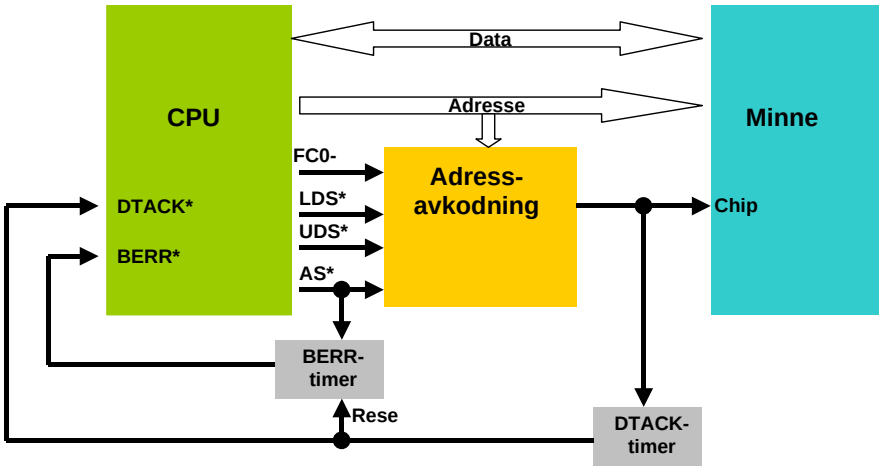
Moment C, I/O- och undantagshantering

1. ***Beskriv den principiella skillnaden mellan synkron och asynkron busscykel.***

   Vid en synkron busscykel sker dataöverföringen vid de tidpunkter som anges av en gemensam klocka. Därmed förväntas alla enheter också att ha den snabbhet som krävs för att hinna med överföringen. Det finns inget utrymme för långsammare enheter än vad som anges av den gemensamma klockan.

   Vid asynkron överföring får den enhet som begär en dataöverföring vänta till dess den svarande enheten är klar och kan leverera svaret. Detta sker med ett handskakningsförfarande där den svarande enheten kvitterar (DTACK) när data finns tillgängligt. Detta förfaringssätt tillåter en bred variation av olika svarstider.

2. ***Visa med en principskiss hur en BERR*-timer (bus error) kopplas till ett mikrodatorsystem baserat på en Motorola 68000-processor. Förklara funktionen.***



BERR-timern hanterar timeout för den asynkrona busscykeln. Dvs om inget minne finns anslutet till en adress som ett program försöker läsa eller skriva till, då får processorn ingen kvittens (DTACK) och skulle behöva vänta i evigheter (genom att infoga WAIT-states i busscykeln) om inte BERR-timern fanns.
Funktion: När en giltig adress indikeras med adress-stroben AS*, startar en timer, BERR. Denna timer stoppas av den kvitterande signalen DTACK. I annat fall kommer timern efter en förutbestämd tid att signalera timeout till BERR* ingången på processorn och därmed generera undantaget BUSS ERROR.

3. ***Vad sparas på stacken när ett avbrott inträffar? Motivera varför det sparas på stacken.***
   När ett avbrott inträffar sparas den adress (programräknare) i huvudprogrammet vid vilket avbrottet inträffade samt hela statusregistret. Adressen behövs för att processorn skall kunna återgå till normal exekvering av huvudprogrammet efter att avbrottsrutinen exekverats. Statusregistret behöver sparas för att processorn skall kunna återgå till samma förutsättningar som rådde innan avbrottet skedde. Dvs återgå till samma värden på statusflaggor, avbrottsmask och mode.

4. ***Förklara begreppet bussfördelning. Det räcker med huvudprincipen och du behöver ej redogöra för olika fördelningsmetoder eller olika hårdvarumässiga inkopplingar.***
   När fler än en aktör behöver ges möglighet att ta initiativ till dataöverföring (fler än en bussmästare), då behöver tillgången till bussen fördelas mellan dessa aktörer. De aktörer som kan ta initiativ till en överföring kallas bussmästare och de andra slavar. Bussfördelning behövs därför att endast en bussmästare åt gången kan ges tillgång till bussen.

5. ***Vilka tre huvudmetoder (protokoll) kan användas för att synkronisera mot en händelse i en periferikomponent (extern händelse) samt förklara skillnaden mellan dessa. Beskriv för- och nackdelar med respektive metod.***

   Polling (Programmerad I/O):
   Programmet i processorn har självt ansvar för att kontinuerligt kontrollera (polla) i fall en periferienhet behöver service. Fördelen är enkelheten, dvs att ingen extra hårdvara behövs. Nackdelen är att metoden kräver mycket resurser från processorn.

   Avbrott:
   En periferienhet får signalera till processorn genom avbrottsbegäran att nu behöver den service. Processorn avbryter pågående aktivitet för att serva (föra över data) den enhet som inkom med avbrottsbegäran. Fördelen är att processorn avlastas eftersom den ej behöver kontinuerligt fråga om service behövs. Nackdelen är den något komplicerade hårdvaran.

   Direct Memory Access (DMA):
   När en periferienhet behöver service skickas en signal till en DMA-controller som fungerar som en bussmästare och begär tillgång till systembussen (DMA request). När processorn ger DMA-controllern tillgång till bussen överförs data direkt från periferienheten till minnet utan att gå vägen via processorn. DMA-controllern sköter överföringen och lämnar tillbaka systembussen till processorn när överföringen är klar (blocköverföring). DMA-överföring kan alternativt blandas med processorns busscykler (interlieving) så att en DMA-överföring sker i bakgrunden. Fördelen är den mycket snabba blocköverföringen och nackdelen är den än mer komplicerade hårdvaran.

   Som alternativ till någon av de tre ovan nämnda metoderna kan anges Timed IO som ingår i gruppen programmerad IO. Periferienheten får service vid jämna förutbestämda tidpunkter då enheten alltid förutsätts vara redo för datatransaktion. Nackdelen är att metoden kräver ett realtidsoperativssystem eller åtminstone en realtidsklocka.

6. ***Skriv en avbrottsrutin som räknar antalet händelser som inträffar. Rutinen ska skriva antalet till en ut port (OUT_EVENTS). Avbrottsrutinen placeras på adress $005000. Skriv den sekvens av ett huvudprogram som initierar avbrottet för autovektoravbrott nivå 5. Redovisa kommenterad källkod.***

Den lilla del av huvudprogrammet som startar avbrottet.

```
    MOVE.L      #count_event,$74  * Kopiera avbrottsrutinens startadress
*                                   till avbrottsvektor 29 med adress 74 Hex
    CLR.B       OUT_EVENT         * Nollställ räknaren för händelser
    MOVE        #$2400,SP         * Möjliggör avbrott på nivå 5 och högre
```

Avbrottsrutinen som endast räknar upp värdet på utporten

```
    ORG $5000                     * Avbrottsrutinen börjar på adress 5000 H

count_event:

    ADDQ.B      #1,OUT_EVENT      * Räkna upp antalet händelser på utporten
    RTE
```