TSEA28 Datorteknik Y, lösningar till tentamen 170815 reviderad 180524

1. a) Kopiera GRx till AR, minska AR med 1 (från uM), spara nya värdet i GRx och hoppa till address 0 om inte noll, annars flytta IR till PC för att genomföra hoppet. Placer mikrokoden på 1:a lediga adress, dvs adress 11.

```
Adress Mikrokod                              Kommentar
  11   buss:=GRx, R:=0, S:=0, AR:=buss      ; Kopiera GRx till AR
       uPC:=uPC+1                           ;
  12   AR:=AR-buss, buss=uM(värde 1)        ; Minska AR med 1, påverkar Z
       uPC:=uPC+1                           ;
  13   Grx:=buss, R:=0, S:=0, buss:=AR,     ; GRx = AR, Nästa instruktion om Z=1
       uPC:=0 om Z=1                        ;
  14   buss:=IR, PC:=buss, uPC:=0           ; Hoppa till A-fältets adress, nästa
                                            ; instruktion
```

b) Två möjliga tider beroende på om hoppet tas eller inte. Utan hopp utförs adresserna 0,1,2,3,11,12,13, dvs 7 klockcykler. Om hopp tas (GRx = 1 vid instruktionen start) utförs förutom 0-13 även adress 14. dvs 8 klockcykler.

c) K1 styrs av ett 4-bitars värde, därför har det 16 adresser. uM styrs av uPC vilket har 7 bitar. Därför har uM 128 olika adresser.

2. a) Fördel med länkregister: behöver inte läsa/skriva minnet vid subrutinanrop. Det kan då gå fortare att utföra subrutinanrop. Förutsätter i så fall att subrutinen inte själv gör subrutinanrop.

Nackdel med länkregister: Behöver ytterligare instruktion om återhoppsaddress ska sparas på stacken ifall subrutinen själv ska utföra ett subrutinanrop.

b) När strömmen slås på måste det finnas ett program som datorn ska utföra. Detta måste ligga någonstans i primärminnet. Det behövs därför någon form av icke-volatilt minne med ett program (t ex BIOS, bootstrap eller liknande ruinter).

c) Big-endian betyder att om processorn lagrar ett ord i minnet så placeras mest signifikant byte på lägsta address. Detta är vad 68000 gör.

d) Virtuellt minne tillåter program att få sammanhängande minnesområden trots att det inte finns fysiska minnesområden som är stora nog. Virtuellt minne tillåter även delande av minnesareor mellan olika program och skydd mot skrivning i programminne.

e) En processor som har delay slot kommer utföra instruktionen efter ett villkorligt hopp oberoende om hoppet ska tas eller inte.

TSEA28 Datorteknik Y, lösningar till tentamen 170815 reviderad 180524

3. Använd D0 som räknare för hur många tecken som hittats hittills, använd A0 för att peka på var i tabellen vi är, och A1för att peka i strängen.

```
AntalTkn:   clr.l     D0          ; Nollställ D0 (32-bitar tal!)
NextStr:    move.l    (A0)+,A1    ; Hämta adress till start av sträng
            cmp.l     #0,A1       ; Ingen sträng (Adress = 0)?
            beq       Slut        ; Adress = 0 betyder inga fler strängar
NextChar:   cmp.b     #0,(A1)+    ; Är tecken = 0
            beq       NextStr     ; Tecken = 0, ta nästa sträng istället
            add.l     #1,D0       ; Vanligt tecken, öka teckenräknare
            bra       NextChar    ; Ta nästa tecken i strängen
Slut:       rts
```

4. a) Då 11011_{2C} har en etta i MSB så är det negativt. Om man byter tecken på talet fås $-(11011_{2C}) = 00100_2+1=00101_2 = 5_{10}$. Dvs $11011_{2C} = -5_{10}$. Det positiva heltalet $11011_2 = (16+8+2+1)_{10} = 27_{10}$.

b) Addition av tvåkomplementstal kräver att talen är lika långa, dvs 3-bitars talet behöver teckenförlängas till 6 bitars längd. Notera att 6-bitars talet är positivt och 3-bitars talet är negativt. Det går då inte att få något spill, så det räcker med 6 bitars resultat.

```
Carry-bitar:   11111   (decimalt)
                010010   (18)
               +111110   (-2)
              ---------
                010000   (16)
```

c) En 1:a i N-flaggan betyder inte att beräkningen gav ett negativt svar. En 1:a i N-flaggan kan även fås om två stora positiva tal adderas och svaret hamnar utanför talområdet.

5. Avbrottsrutin betyder att eventuella register som används måste sparas på stack så de kan återställas efter avbrottet. Vid rotation (rol.w) flyttas bitarna åt vänster, och de som ramlar över kanten till vänster placeras istället i höger kant.

```
Avbrott:    move.l    D0,-(A7)    ; spara D0 på stack
            move.b    $10080,D0   ; läs in siffror till vänster
            lsl.w     #8,D0       ; Flytta till bit position 15-8
            move.b    $10082,D0   ; hämta siffror till höger
            rol.w     #4,D0       ; Rotera 4 bitpositioner höger
            move.b    D0,$10082   ; högra siffrorna
            lsr.w     #8,D0       ; Hämta bit positioner 15-8
            move.b    D0,$10080   ; Nya vänstra siffrorna
            move.l    (A7)+,D0    ; återställ D0
            rte                   ; avsluta avbrott
```

TSEA28 Datorteknik Y, lösningar till tentamen 170815 reviderad 180524

6. a) 2048 byte cache totalt, varje cacheline är 32 byte => 2048/32=64 cachelines

b) 64 cachelines totalt, 2 vägs gruppassociativ => 32 möjliga indexvärden. För index används därför 5 bitar. För att peka på byte i cacheline används 5 bitar. Kvar av de 32 adressbitarna blir då 32-5-5=22 bitar.

c) Ökning av adressen sker i bit 8 ($100 = 0000...0000100000000$_2$), och tag börjar med bit 10. Dvs när adresserna $10000, $10100, $10200 och $10300 har samma tag men olika index. När adressen blir $10400 fås en ny tag och samma index som för adress $10000. Dock är cache:n 2-vägs gruppassociativ, så data kastas ur cachen först när samma tag fås en 3:e gång. Dvs när adressen $10800 fås så kommer cachen behöva kasta ut data från adress $10000. Totalt har då 8 pixlar lästs in.

d) Om startadressen är $1001E kommer en 32-bitars pixel hamna med 2 byte i cacheline med start på adress $10000 och 2 byte i cacheline med start på adress $10020. Nästa pixel börjar på adress $10100 vilket ännu inte hämtats. Dvs det blir dubbelt så många cachemissar.