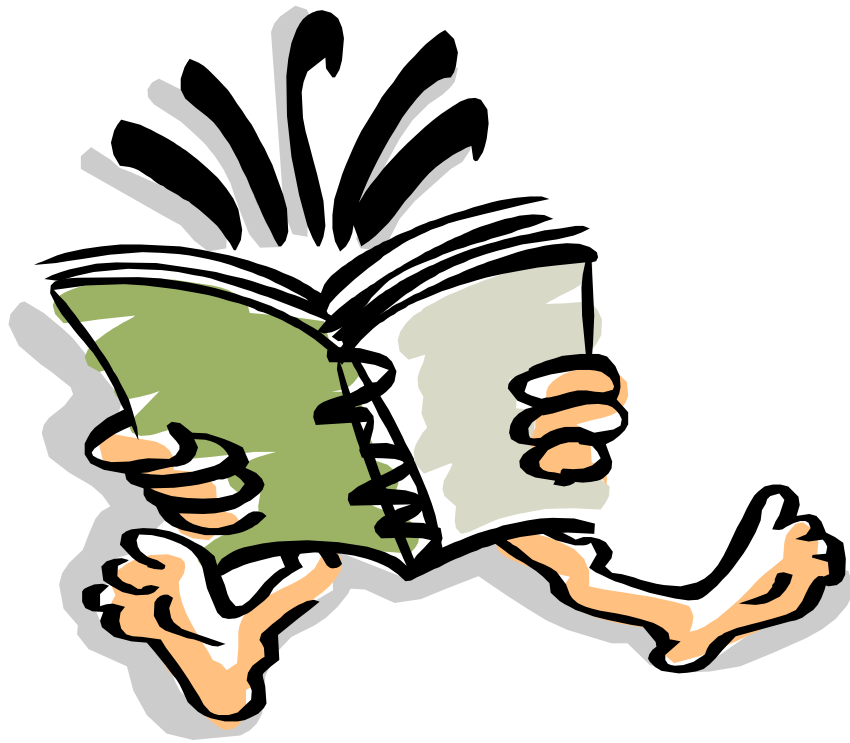




Tentamen

Mikrodator teknik, 4p

2000-02-19

Skrivtid:	5 timmar
Ansvarig lärare:	Mattias O'Nils
Telefon:	070-695 7668
Hjälpmedel:	The 68000's Instruction Set Miniräknare
Program:	Systemteknikprogrammet, ST3

OBS!

Skriv tydligt!

Skriv namn och kod på varje blad.

Skriv bara på en sida av varje blad.

Redovisa alla steg i beräkningar och konstruktioner.

Rita snygga figurer!

Lycka till,
/Mattias

Del A (50p)

1. Beskriv Motorola 68000 processorns instruktionsexekvering. 6p
2. Beskriv skillnaden mellan MOVE.W #30, D2 och MOVE.W 30, D2, rita en figur för att förklara. 4p
3. Beskriv von Neumann arkitekturen, vad som karakteriserar denna och beskriv instruktionsexekveringen för denna. 6p
4. a) Beskriv i stora drag hur en pipeline fungerar.
b) Beskriv i stora drag hur ett cacheminne fungerar. 6p
5. Nämn två sätt att synkronisera I/O operationer samt beskriv dessa. 6p
6. Nämn tre olika metoder att överföra parametrar till en funktion, samt för- och nackdelar med dessa. 6p
7. Antag att register D1 innehåller \$00000077. Ange D1, C-flaggans samt Z- flaggans värde efter instruktionerna nedan har exekverats. 5p
 - a) ADDI.B #11,D1
 - b) ADDI.B #\$89,D1
 - c) ANDI.B #\$80,D1
 - d) SUBL.B #\$64,D1
 - e) LSL.B #1,D1
8. Beskriv hur adressen till en undantagsrutin beräknas i 68000. 5p
9. Vad kommer D2 att innehålla efter att instruktionen har körts. Innehållet i minnet är specificerat av tabellen nedan. (A3 = \$4004)
 - a) MOVE.W (\$4000).L, D2
 - b) MOVE.B (A3), D2
 - c) MOVE.B -(A3), D2
 - d) MOVE.L -(A3), D2
 - e) MOVE.W -2(A3), D2
 - f) MOVE.B 4(A3), D2

6p

Adress (Hexadecimal)	Innehåll (Hexadecimal)
4000	54
4001	32
4002	FF
4003	AD
4004	4F
4005	6A
4006	34
4007	78
4008	99
4009	2B

Del B (50p)

10. Implementera följande IF-sats i 68000-assembler. Alla variabler är av 16-bit storlek och kan ersättas med register.

5p

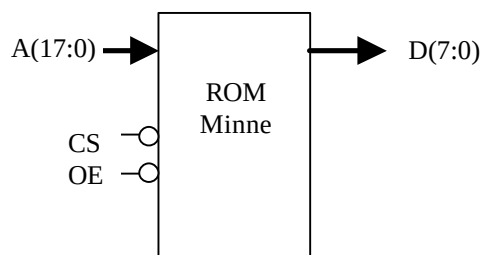
```
if (tal1 > tal2)
    max = tal1;
else
    max = tal2;
```

11. Skriv ett program räknar antalet pulser på in-porten IN_1 med adressen (\$F0000) (varje puls detekteras på positiv flank), och skriver antal pulser på ut-porten OUT_1 (binärt) som har adressen (\$F0001).

12p

12. Sätt samman ett ROM-minne på 2 Mbyte (16-bitar bred), med hjälp av minneskomponenter 256kx8bit (se komponenten). Generera bara CS signalen.

12p



13. Konstruera en fullständig minnesavkodare med följand minneskarta. Använd en eller flera 4-16 avkodaren 74HC154 beskriven på nästa sida. (Dvs. generera CS signaler)

9p

ROM	00 0000 -	01 FFFF
RAM1	02 0000 -	09 FFFF
RAM2	0A 0000 -	0F FFFF
IO_0	0F 0000 -	0F 01FF
IO_1	0F 0200 -	0F 03FF
IO_2	0F 0400 -	0F 07FF
IO_3	0F 0800 -	0F 0FFF

14. Skriv en subrutin (assembler) som beräknar medelvärdet av en sammanhängande vektor av heltal i minnet, `int mean_value(int *vec, int no_data)`. Funktionsparametrar skickas via stacken, `vec` pekar på första elementet i vektorn och `no_data` indikerar hur många tal vektorn innehåller. Data i vektorn samt `no_data` är 16-bitar. (Inga register ska påverkas av subrutinen)

12p

