

Dugga i Mikrodatorteknik 4p

Moment B, Assemblerprogrammering

Hjälpmedel: 68000's instruktionsmanual och miniräknare
Tid: 1 timme och 45 min
Max: 100p
Prel. godkänd: ≥50p samt presenterat delar av inlämningsuppgiften.

OBS!

Skriv tydligt!

Skriv namn på varje blad.

Skriv bara på en sida av varje blad.

Redovisa alla steg i beräkningar och konstruktioner.

Rita snygga figurer!

1 Statusflaggors beteende (18p)

Vad blir C-, N- och Z- flaggornas värden samt D0:s värden efter varje instruktion i följande program. (redovisa beräkningar).

```
MOVE.B    #$C5, D0
ADDI.B    #$3B, D0
MOVE.B    #$5F, D0
ADDI.B    #$F1, D0
EORI.B    #$80, D0
ASR.B     #2, D0
```

2 Hoppinstruktioners beteende (24p)

Du ska bestämma exekveringsordningen för instruktionerna i följande program, redogör också varför vissa hopp utförs och vissa inte (motivera med registrens och statusflaggornas värden).
Tänk på att vissa instruktioner kan utföras flera gånger.

```
L1:    MOVE.W    #$A0A0, D1
L2:    BRA        L4
L3:    LSL.W      #1, D1
L4:    BMI        L3
L5:    ANDI.W     #$BEBF, D1
L6:    BEQ        L9
L7:    ASL.W      #$5, D1
L8:    BCS        L12
L9:    EORI.W     #$8000, D1
L10:   ASR.W      #$4, D1
L11:   BMI        L7
L12:   BRA        L12
```

3 Adresseringsmetoder (20p)

Vad kommer D3 och A0 att innehålla efter följande instruktion?

Innehållet i minnet framgår av tabellen. D3 = \$00000000 innan instruktionen, A0= \$00004004 (Koden består av enskilda instruktioner inte ett program).

Adress (Hexadecimal)	Data i minnet (Hex)
4000	39
4001	45
4002	A6
4003	F6
4004	D4
4005	AA
4006	67
4007	89
4008	34
4009	22
400A	00
400B	23
400C	16
400D	24
400E	90
400F	87

- I. MOVE.W \$400A, D3
- II. MOVE.B (A0), D3
- IV. MOVE.L -(A0), D3
- V. MOVE.W (A0)+, D3

4 Programkonstruktion I (25p)

Skriv en subrutin som beräknar skalära vektorprodukten S av två vektorer A och B , där varje element n i vektorerna representeras av 16 bitar med tecken (andra komplementet).

$$S = \sum_{n=0}^{N-1} A_n \cdot B_n, \text{ där } N \text{ är antal element i respektive vektor.}$$

Vektorerna skall överföras som parametrar via register A0 och A1 genom referens. Antal element N i varje vektor skall överföras via register D0 genom värde och vara representerat av 8 bitar. Resultatet skall returneras via register D0 genom värde och representeras av 32 bitar med tecken (andra komplementet). Motsvarande deklaration av en C-funktion skulle kunna se ut som följer.

```
long int scalarProduct( short int *A, short int *B, unsigned char N);
```

Redovisa flödesdiagram eller annan algoritmbeskrivning samt kommenterad assemblerkod.

5 Programkonstruktion II (13p)

Skriv en subrutin som adderar två 64-bitars tal a och b. Svaret skall returneras som ett 64-bitars tal och carryflaggan i statusregistret skall ha ett korrekt värde efter anropet.

```
long int add64( long int a, long int b);
```

Parameter a och b skall överföras via stacken genom värde. Returvärdet överförs via dataregister D0 och D1 genom värde. Inga register förutom D0, D1 och CCR får påverkas av ett anrop till subrutinen. Visa även med ett enkelt huvudprogram hur din subrutin skall anropas, samt hur parametrar tilldelas värden och tas bort från stacken efter anropet. Redovisa kommenterad assemblerkod.

Tips: ADDX

Lycka till!

/Benny