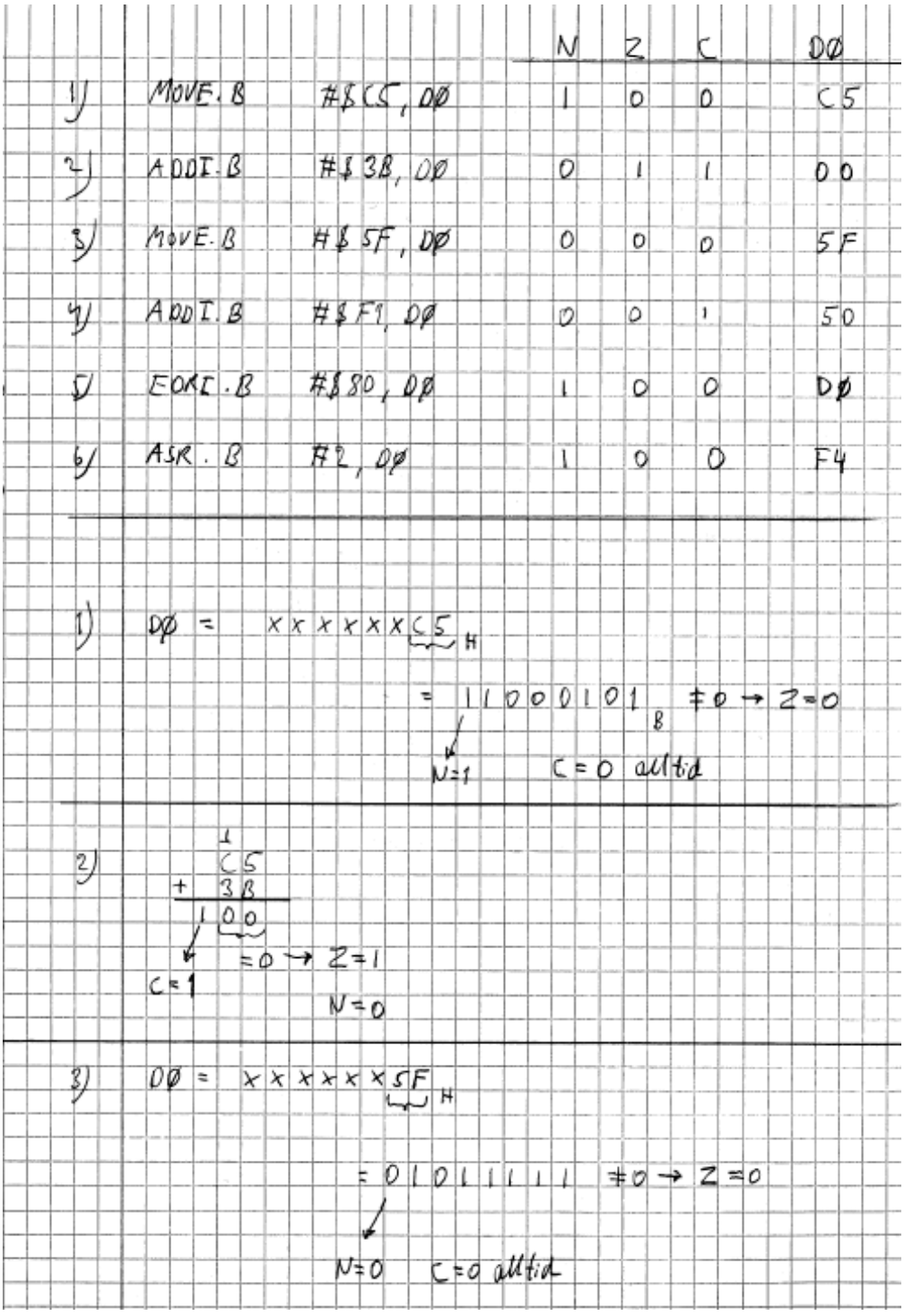
Facit/Lösningsförslag till Dugga 2 i Mikrodatorteknik 4p

Moment B, Assemblerprogrammering

1 Statusflaggors beteende (18p)

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



4)

$$\begin{array}{r} 1 \\ 5F \\ +\ F1 \\ \hline 150 \end{array}$$

$C = 1$

$= 01010000_B \neq 0 \rightarrow Z = 0$

$N = 0$

5)

$50_H = 01010000$

$80_H = 10000000$ (+)

$11010000 = D0 \neq 0 \rightarrow Z = 0$

$N = 1$ $C = 0$ altfel

6)

$D0 = 11010000$

11111010

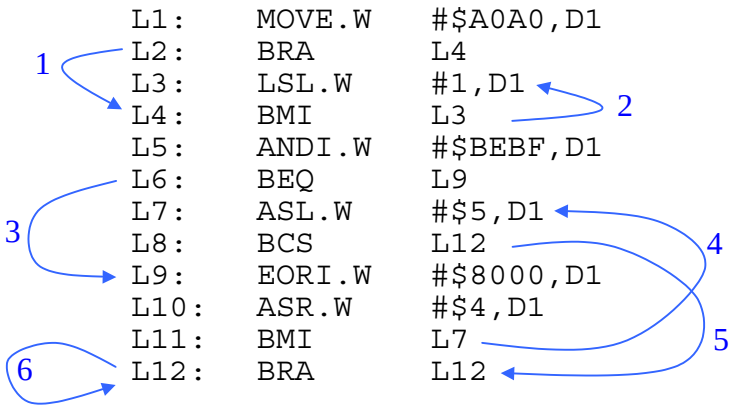
$C = 0$

$X = 0$

$= F4 \neq 0 \rightarrow Z = 0$

$N = 1$

2 Hoppinstruktioners beteende (24p)



Steg	Instruktion	D1	N Z C	Hopp
1	L1	XXXXA0A0	1 0 0	
2	L2	XXXXA0A0	1 0 0	1
3	L4	XXXXA0A0	1 0 0	2
4	L3	XXXX4140	0 0 1	
5	L4	XXXX4140	0 0 1	
6	L5	XXXX0000	0 1 0	
7	L6	XXXX0000	0 1 0	3
8	L9	XXXX8000	1 0 0	
9	L10	XXXXF800	1 0 0	
10	L11	XXXXF800	1 0 0	4
11	L7	XXXX0000	0 1 1	
12	L8	XXXX0000	0 1 1	5
13	L12	XXXX0000	0 1 1	6

1. Detta hopp utförs ovillkorligen
2. Z-flaggan blev etta redan vid steg 1 och därför utförs detta hopp
3. Hoppet utförs därför att Z-flaggan ett-ställdes vid steg 6
4. Hoppet görs därför att N-flaggan ett-ställdes vid steg 9
5. Vid steg 11 ett-ställdes Carry-flaggan och därför görs hoppet
6. Detta hopp är ovilkorligt och fortsätter att utföras i all evighet….

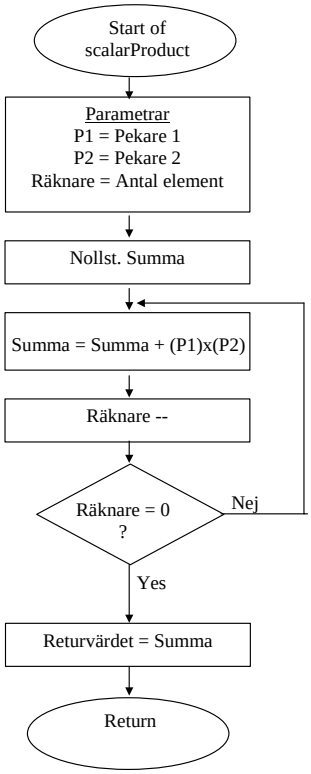
3 Adresseringsmetoder (20p)

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

D3 = $00000000 , A0= $00004004 innan instruktionen

	Instruktion	D3	A0
I	MOVE.W $400A, D3	00000023	00004004
II	MOVE.B (A0), D3	000000D4	00004004
IV	MOVE.L -(A0), D3	3945A6F6	00004000
V	MOVE.W (A0)+, D3	0000D4AA	00004006

4 Programkonstruktion I (25p)



4.1 Källkod

Följande källkod innehåller även ett huvudprogram som anropar subrutinen scalarProduct och därmed multiplicerar vektorerna Avektor och Bvektor. Endast koden för subrutinen (markerad blå) är det som efterfrågades i uppgiften.

```
    ORG $4000

main:

    LEA Avektor,A0
    LEA Bvektor,A1
    MOVE.B #9,D0
    BSR scalarProduct
    STOP #$2700


scalarProduct:
    MOVE.B D0,D3      * Räknare = Antal element
    CLR.L D0          * Nollst. Summa
again:
    MOVE.W (A0)+,D1   * Element (P1)
    MOVE.W (A1)+,D2   * Element (P2)
    MULS.W D1,D2      * (P1)x(P2)
    ADD.L  D2,D0      * Summa = Summa + (P1)x(P2)

    SUBQ.B #1,D3      * Räknare --
    BNE    again      * Hoppa om Räknare != 0
    RTS


Avektor    DC.W  1,2,3,4,5,6,7,8,9
Bvektor    DC.W  9,8,7,6,5,4,3,2,1
```

5 Programkonstruktion II (13p)

```
    ORG $4000


appstart:

    MOVE.L #1,-(SP);          * High word of a
    MOVE.L #$FFFFFFFF,-(SP); * Low word of a
    MOVE.L #$FFFFFFFF,-(SP); * High word of b
    MOVE.L #4,-(SP);          * Low word of b
    BSR add64
                              * High order result in D1 and low in D0
    STOP #$2700


add64:
    MOVEM.L D2/D3,-(SP);      * Save registers on stack
    MOVE.L (12,SP),D0         * Get low order bits of b, a31-a0
    MOVE.L (16,SP),D1         * Get high order bits of b, a63-a32
    MOVE.L (20,SP),D2         * Get low order bits of a, a31-a0
    MOVE.L (24,SP),D3         * Get high order bits of a, a63-a32
    ADD.L  D2,D0              * Sum the low order bits
    ADDX.L D3,D1              * Sum the high order bits and previous carry
    MOVEM.L (SP)+,D2/D3       * Restore registers
    RTS                       * Return with high order result in D1 and
                              * low order in D0
```