

# Lösningförslag tenta 2013-06-03

(Version 4 med reservation för eventuella fel. Uppdaterad 131210.)

1.  $X = 010\ 1011_2$ ;  $Y = 101\ 1111_2$  (7 bitars ordlängd)

a)  $[0, 2^n - 1] = [0, 2^7 - 1] = [0, 127]$  (1p)

b)  $[-2^{n-1}, +2^{n-1} - 1] = [-2^{7-1}, +2^{7-1} - 1] = [-64, +63]$  (1p)

c)  $S = X + Y$

76543210	bitnummer
11111110	carry
0101011	X
+1011111	Y
0001010	= S

(1p)

d)  $\underline{N} = s_6 = \underline{0}$   
 $\underline{Z} = \underline{0}$  ( $S \neq 0$ )  
 $\underline{V} = x_6 * y_6 * s_6' + x_6' * y_6' * s_6 = 0 * 1 * 0' + 0' * 1' * 0 = \underline{0}$   
 $\underline{C} = c_7 = \underline{1}$

NZVC = 0001 (1p)

e)  $D = X + Y_{1k} + 1$

76543210	bitnummer
01000111	carry
0101011	X
+0100000	Y <sub>1k</sub>
1001100	= D

(1p)

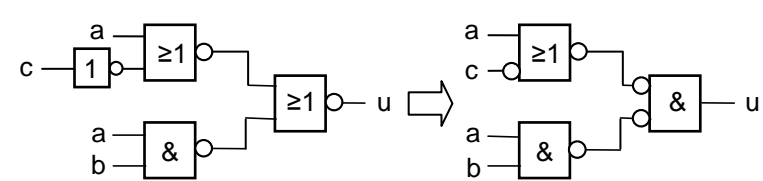
f)  $\underline{N} = d_6 = \underline{1}$   
 $\underline{Z} = \underline{0}$  ( $D \neq 0$ )  
 $\underline{V} = x_6 * y_{6k} * d_6' + x_6' * y_{6k}' * d_6 = 0 * 0 * 1' + 0' * 0' * 1 = \underline{1}$   
 $\underline{C} = c_7' = 0' = \underline{1}$

NZVC = 1011 (1p)

g)  $\underline{X} = 010\ 1011_2 = 2B_{16} = 2 * 16 + 11 = \underline{43}$   
 $\underline{Y} = 101\ 1111_2 = 5F_{16} = 5 * 16 + 15 = \underline{95}$   
 $\underline{S} = 000\ 1010_2 = 0A_{16} = 0 * 16 + 10 = \underline{10}$  Resultatet S är felaktigt eftersom  $C = 1$ .  
 $\underline{D} = 100\ 1100_2 = 4C_{16} = 4 * 16 + 12 = \underline{76}$  Resultatet D är felaktigt eftersom  $C = 1$ . (1p)

h) ( $x_6 = 0$ , pos)  $\underline{X} = 010\ 1011_2 = \underline{43}$   
 ( $y_6 = 1$ , neg)  $Y_{2k} = 2^7 - 95 = 128 - 95 = 33$   $\underline{Y}$  motsvarar  $\underline{-33}$   
 ( $s_6 = 0$ , pos)  $\underline{S} = 000\ 1010_2 = \underline{10}$  Resultatet S är korrekt eftersom  $V = 0$ .  
 ( $d_6 = 1$ , neg)  $D_{2k} = 2^7 - 76 = 128 - 76 = 52$   $\underline{D}$  motsvarar  $\underline{-52}$  Fel eftersom  $V = 1$ . (1p)

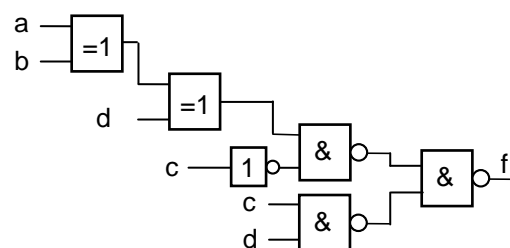
i)  $N = -56,375_{10} = -111000.011 = -1.11000011 * 2^5$  Format: s/c/f  
 $s = 1$  (neg);  $c = \exp + 2^{8-1} - 1 = 5 + 127 = 132 = 10000100_2$ ;  $f =$  mantissa - 1  
 $N_{flyt} = 1/100\ 0010\ 0/110\ 0001\ 1000\ 0000\ 0000\ 0000_2 = C2618000_{16}$  (2p)

2. a) 

$u = (a + c')(ab) = ab + abc' = ab(1 + c') = \underline{ab}$  (2p)

b)  $f = c'(a \oplus b \oplus d) + cd$

		cd			
		00	01	11	10
ab	00	0	1	1	0
	01	1	0	-	0
	11	-	1	-	0
	10	-	-	1	0



(4p)

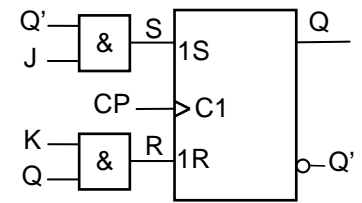
3.

a)

J	K	Q	Q <sup>+</sup>	S	R
0	0	0	0	0	-
0	0	1	1	-	0
0	1	0	0	0	-
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	-	0
1	1	0	1	1	0
1	1	1	0	0	1

		S	Q
		0	1
JK	00	0	-
	01	0	0
	11	1	0
	10	1	-

		R	Q
		0	1
JK	00	-	0
	01	-	1
	11	0	1
	10	0	0



$$S = JQ'$$

$$R = KQ$$

(3p)

b)

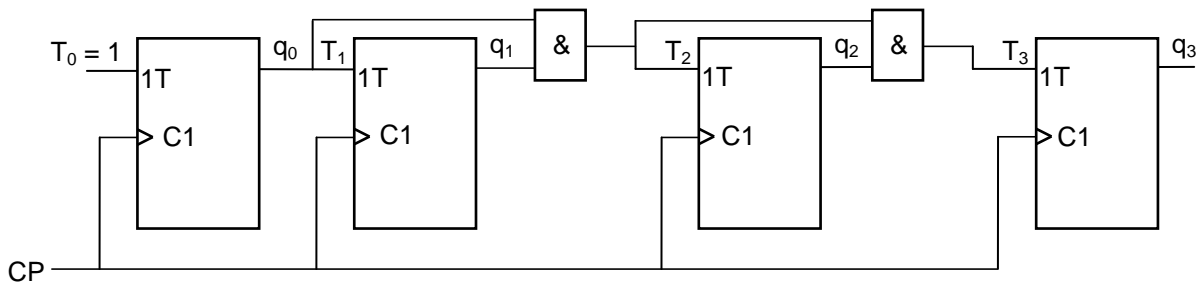
För T-vippor gäller att  $q^+ = Q$  för  $T = 0$  och  $q^+ = Q'$  för  $T = 1$ .  $Q$  ändras alltså om  $T = 1$ .

Räknares ut signaler :  $q_3q_2q_1q_0$

När man betraktar de binära värdena vid uppräknin ser man att bit  $q_0$  alltid ändras efter klockpuls och att övriga bitar bara ändrar sig om alla bitar med lägre nummer är ettor.

Det innebär att följande uttryck fås för  $T$ :  $T_0 = 1$ ;  $T_1 = q_0$ ;  $T_2 = q_1q_0$ ;  $T_3 = q_2q_1q_0$

Omskrivning ger:  $T_0 = 1$ ;  $T_1 = q_0$ ;  $T_2 = q_1T_1$ ;  $T_3 = q_2T_2$



(6p)

4.

CP	Styr signaler (=1)	RTN	Reg A	Reg B	Reg T	Reg R
1	OE <sub>A</sub> , LD <sub>T</sub>	A → T,	18	2C	?	?
2	OE <sub>B</sub> , f <sub>3</sub> , f <sub>1</sub> , LD <sub>R</sub>	B + T → R	"	"	18	?
3	OE <sub>R</sub> , f <sub>3</sub> , f <sub>1</sub> , f <sub>0</sub> , LD <sub>R</sub>	2R → R	"	"	"	44
4	OE <sub>R</sub> , f <sub>3</sub> , f <sub>1</sub> , f <sub>0</sub> , LD <sub>R</sub>	2R → R	"	"	"	88
5	OE <sub>R</sub> , f <sub>3</sub> , f <sub>2</sub> , LD <sub>R</sub>	R - T - 1 → R	"	"	"	10
6	OE <sub>R</sub> , f <sub>1</sub> , f <sub>0</sub> , LD <sub>R</sub>	R <sub>1k</sub> → R	"	"	"	F7
7	OE <sub>R</sub> , f <sub>3</sub> , g <sub>0</sub> , LD <sub>R</sub>	R + 1 → R	"	"	"	08
8	OE <sub>R</sub> , LD <sub>B</sub>	R → B	"	"	"	09
9	?		18	09	18	09

(1p)

(3p)

5. a) FLISP:

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q <sub>4</sub>	Q <sub>4</sub> ·I <sub>xx</sub>	M(PC) → T, PC+1 → PC	MR, LD <sub>T</sub> , INC <sub>PC</sub>
Q <sub>5</sub>	Q <sub>5</sub> ·I <sub>xx</sub>	M(T+Y) → T	MR, g <sub>13</sub> , LD <sub>T</sub>
Q <sub>6</sub>	Q <sub>6</sub> ·I <sub>xx</sub>	U = A ∧ T, ALU(N,Z,V,-) → CC, New Fetch	OE <sub>A</sub> , f <sub>2</sub> , f <sub>1</sub> , f <sub>0</sub> , g <sub>3</sub> , g <sub>2</sub> , LD <sub>CC</sub> , NF

Detta är: BITA n, Y

FLEX:

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q <sub>5</sub>	Q <sub>5</sub> ·I <sub>xx</sub>	X → T	OE <sub>X</sub> , LD <sub>T</sub>
Q <sub>6</sub>	Q <sub>6</sub> ·I <sub>xx</sub>	B + T → R	OE <sub>B</sub> , f <sub>3</sub> , f <sub>1</sub> , LD <sub>R</sub>
Q <sub>7</sub>	Q <sub>7</sub> ·I <sub>xx</sub>	R → MA	OE <sub>R</sub> , LD <sub>MA</sub>
Q <sub>8</sub>	Q <sub>8</sub> ·I <sub>xx</sub>	M → A, (Next Fetch)	MR, LD <sub>A</sub> , NF

Detta är: LDAA B, X

(2p)

b) FLISP:

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q <sub>4</sub>	Q <sub>4</sub> ·I <sub>DF</sub>	M(PC) → TA, PC+1 → PC	MR, LD <sub>TA</sub> , INC <sub>PC</sub>
Q <sub>5</sub>	Q <sub>5</sub> ·I <sub>DF</sub>	M(TA) → R	MR, g <sub>14</sub> , f <sub>3</sub> , f <sub>0</sub> , LD <sub>R</sub>
Q <sub>6</sub>	Q <sub>6</sub> ·I <sub>DF</sub>	M(PC) → TA, PC+1 → PC	MR, LD <sub>TA</sub> , INC <sub>PC</sub>
Q <sub>7</sub>	Q <sub>7</sub> ·I <sub>DF</sub>	M(TA) → T	MR, g <sub>14</sub> , LD <sub>T</sub>
Q <sub>8</sub>	Q <sub>8</sub> ·I <sub>DF</sub>	R + T → R, ALU(NZVC) → CC, 0 → T	OE <sub>R</sub> , f <sub>3</sub> , f <sub>1</sub> , f <sub>0</sub> , LD <sub>R</sub> , LD <sub>CC</sub> , CLR <sub>T</sub>
Q <sub>9</sub>	Q <sub>9</sub> ·I <sub>DF</sub>	M(PC) → TA, PC+1 → PC	MR, LD <sub>TA</sub> , INC <sub>PC</sub>
Q <sub>A</sub>	Q <sub>A</sub> ·I <sub>DF</sub>	R → M(TA), (New Fetch)	OE <sub>R</sub> , g <sub>14</sub> , MW, NF

OPKOD
Adr1
Adr2
Adr3

FLEX:

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q <sub>5</sub>	Q <sub>5</sub> ·I <sub>DF</sub>	PC → MA, PC+1 → PC	OE <sub>PC</sub> , LD <sub>MA</sub> , IncPC
Q <sub>6</sub>	Q <sub>6</sub> ·I <sub>DF</sub>	M → MA	MR, LD <sub>MA</sub>
Q <sub>7</sub>	Q <sub>7</sub> ·I <sub>DF</sub>	M → T	MR, LD <sub>T</sub>
Q <sub>8</sub>	Q <sub>8</sub> ·I <sub>DF</sub>	PC → MA, PC+1 → PC	OE <sub>PC</sub> , LD <sub>MA</sub> , IncPC
Q <sub>9</sub>	Q <sub>9</sub> ·I <sub>DF</sub>	M → MA	MR, LD <sub>MA</sub>
Q <sub>A</sub>	Q <sub>A</sub> ·I <sub>DF</sub>	M + T → R, Flags → CC	MR, f <sub>3</sub> , f <sub>1</sub> , LD <sub>R</sub> , LD <sub>CC</sub>
Q <sub>B</sub>	Q <sub>B</sub> ·I <sub>DF</sub>	PC → MA, PC+1 → PC	OE <sub>PC</sub> , LD <sub>MA</sub> , IncPC
Q <sub>C</sub>	Q <sub>C</sub> ·I <sub>DF</sub>	M → MA	MR, LD <sub>MA</sub>
Q <sub>D</sub>	Q <sub>D</sub> ·I <sub>DF</sub> ·Z	R → M, (Next Fetch)	OE <sub>R</sub> , MW, NF

(4p)

6.

a) Med hoppinstruktionen BRA Adr istället för JMP Adr blir inte programmet beroende av startadressen. Det kan då flyttas i minnet. (1p)

b) Subtraktionen  $65_{16} - 97_{16} = CE_{16}$  ger flaggvärdena  $NZVC = 1011$ . Detta medför att båda hoppen utförs. BMI ( $N = 1$ ) och BGT ( $(N \oplus V) + Z = (1 \oplus 1) + 0 = 0$ ). (2p)

c)

Adr	Data	~	Läge	Assemblerkod		Offset
				Operation	Operand	
		5		LDA	#-10	
				BSR	DELAY	
A0	11	3	DELAY	PSHX		
A1	10	3		PSHA		
A2	90 05	2	LOOP	LDX	#INLP	
A4	CC FF	4	XLOOP	LEAX	-1,X	
A6	9C 00	3		CMPX	#0	
A8	BF	4		EXG	X,SP	
A9	BF	4		EXG	X,SP	
AA	23 F8	4		BPL	XLOOP	A4 - AC = F8
AC	07	3		INCA		
AD	25 F3	4		BNE	LOOP	A2 - AF = F3
AF	14	3	DELX	PULA		
B0	15	3		PULX		
B1	43	2		RTS		

(3p)

d) BSR DELAY tar 5 klockpulser.

A-reg innehåller värdet -10 vid anropet, så yttre slingan (LOOP) utförs 10 gånger.

$$T = 5 + 3 + 3 + [2 + (4 + 3 + 4 + 4 + 4) \cdot 6 + 3 + 4] \cdot 10 + 3 + 3 + 2 = 19 + [9 + 19 \cdot 6] \cdot 10 = \underline{1249 \mu s}$$

(3p)

7.

MODIFY	PSHX PSHY		Spara register på stack
	CLR	COUNT	Räknare för inledande ettor
MOLOOP	LDA	,X+	Hämta data från BLOCK1
	BEQ	MODEX	Dataord = 0? Ja, avsluta
	BPL	NOCNT	Bit7 = 0. Räknas ej
	INC	COUNT	Bit7 = 1. Öka räknare
NOCNT	ANDA	##%01111011	Nolla bit7 o bit2
	ORA	##%00100000	Etta bit5
	EORA	##%00010000	Invertera bit4
	STA	,Y+	Skriv till BLOCK2
	BRA	MOLOOP	Behandla nästa dataord
MODEX	CLR	0,Y	Slutmarkering för BLOCK2
	LDA	COUNT	Hämta räknaren
	PULY		Återställ register
	PULX		
	RTS		
COUNT	RMB	1	Räknare för inledande ettor

(8p)

8.

0000H	Upptaget för styrregister mm	} 32 kbyte ROM
03FFH		
0400H	Inport1	
0401H	Utport	
0402H	ROM	
7FFFH		
8000H		
E000H		
FFFDH	Inport2	
FFFEH		
FFFFH		Inport3

ROM  
32k = 2<sup>15</sup>

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 0000H =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 7FFFH =	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CS	15 st															

RWM  
8k = 2<sup>13</sup>

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: E000H =	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: FFFFH =	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CS	13 st															

Inport1:

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: 0400H =	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
CS																

Utport:

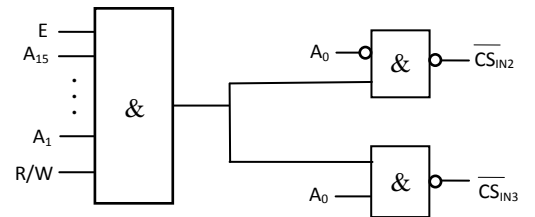
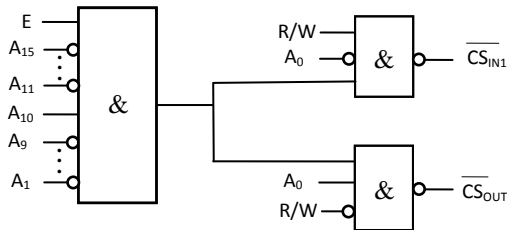
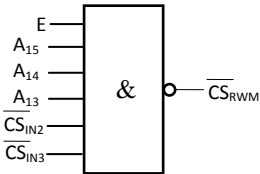
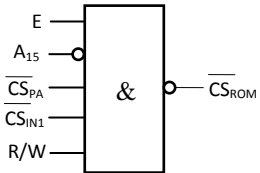
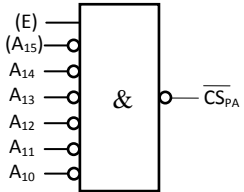
A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: 0401H =	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
CS																

Inport2:

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: FFFEH =	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
CS																

Inport3:

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: FFFFH =	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CS																



(8p)