

Lösningsförslag tenta 2013-06-03

(Version 5 med reservation för eventuella fel. Uppdaterad 140416.)

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) $N = s_6 = 0$
 $Z = 0$ ($S \neq 0$)
 $V = x_6 * y_6 * s_6' + x_6' * y_6' * s_6 = 0 * 1 * 0' + 0' * 1' * 0 = 0$
 $C = c_7 = 1$
 NZVC = 0001 (1p)

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

76543210	bitnummer
01000111	carry
0101011	X
+0100000	Y _{1k}
1001100	$= D$

 (1p)

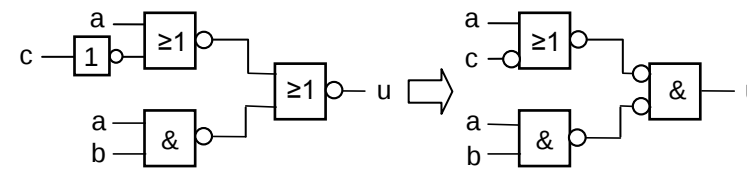
f) $N = d_6 = 1$
 $Z = 0$ ($D \neq 0$)
 $V = x_6 * y_{6ik} * d_6' + x_6' * y_{6ik}' * d_6 = 0 * 0 * 1' + 0' * 0' * 1 = 1$
 $C = c_7' = 0' = 1$
 NZVC = 1011 (1p)

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

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

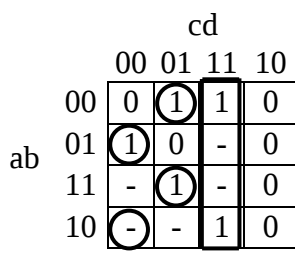
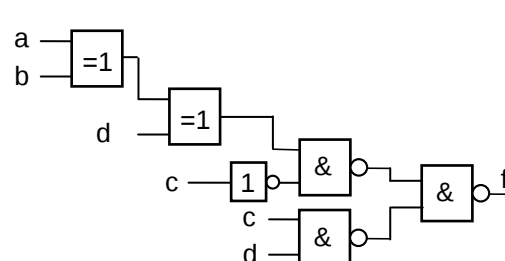
i) $N = -56,375_{10} = -111000.011 = -1.11000011 * 2^5$ Format: s/c/f
 $s = 1$ (neg); $c = \exp + 2^{8-1} - 1 = 5 + 127 = 132 = 10000100_2$; f = mantissa - 1
 $N_{\text{flyt}} = 1/100\ 0010\ 0/110\ 0001\ 1000\ 0000\ 0000\ 0000_2 = C2618000_{16}$ (2p)

2.

a) 

$$u = (a + c')(ab) = ab + abc' = ab(1 + c') = ab$$
 (2p)

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

(4p)

3.

a)

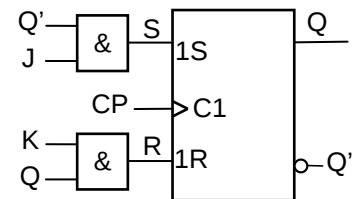
J	K	Q	Q ⁺	S	R
0	0	0	0	0	-
0	0	1	1	-	0
0	1	0	0	0	-
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	-	0
1	1	0	1	1	0
1	1	1	0	0	1

S	Q
0	1
0	-
0	0
1	0
1	-

$$S = JQ'$$

R	Q
0	1
-	0
1	1
0	0
0	0

$$R = KQ$$



(3p)

b)

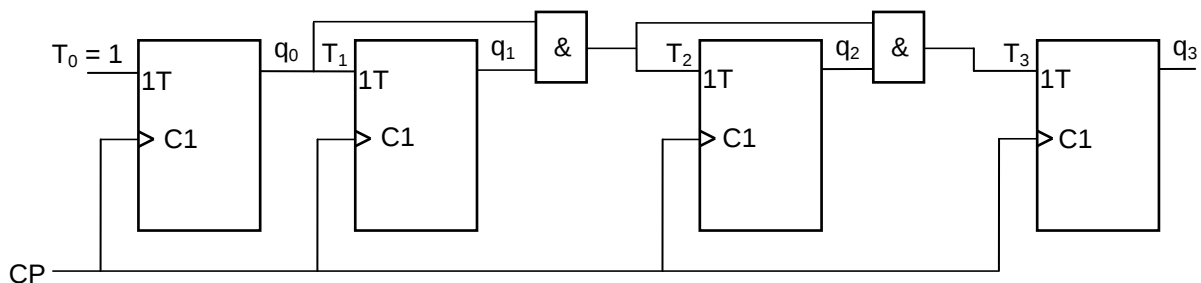
För T-vippor gäller att $q^+ = Q$ för $T = 0$ och $q^+ = Q'$ för $T = 1$. Q ändras alltså om $T = 1$.

Räknares ut signaler : $q_3q_2q_1q_0$

När man betraktar de binära värdena vid uppräknings 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	Styrsignaler (=1)	RTN	Reg A	Reg B	Reg T	Reg R
1	OE _A , LD _T	A → T,	18	2C	?	?
2	OE _B , f ₃ , f ₁ , LD _R	B + T → R	"	"	18	?
3	OE _R , f ₃ , f ₁ , f ₀ , LD _R	2R → R	"	"	"	44
4	OE _R , f ₃ , f ₁ , f ₀ , LD _R	2R → R	"	"	"	88
5	OE _R , f ₃ , f ₂ , LD _R	R - T - 1 → R	"	"	"	10
6	OE _R , f ₁ , f ₀ , LD _R	R _{1k} → R	"	"	"	F7
7	OE _R , f ₃ , g ₀ , LD _R	R + 1 → R	"	"	"	08
8	OE _R , LD _B	R → B	"	"	"	09
9	?		18	09	18	09

(1p)

(3p)

5. a) FLISP:

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

Detta är: BITA n,Y

FLEX:

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q ₅	Q ₅ ·I _{xx}	X → T	OE _X , LD _T
Q ₆	Q ₆ ·I _{xx}	B + T → R	OE _B , f ₃ , f ₁ , LD _R
Q ₇	Q ₇ ·I _{xx}	R → MA	OE _R , LD _{MA}
Q ₈	Q ₈ ·I _{xx}	M → A, (Next Fetch)	MR, LD _A , NF

Detta är: LDAA B,X

(2p)

b) FLISP:

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

OPKOD
Adr1
Adr2
Adr3

FLEX:

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q ₅	Q ₅ ·I _{DF}	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC
Q ₆	Q ₆ ·I _{DF}	M → MA	MR, LD _{MA}
Q ₇	Q ₇ ·I _{DF}	M → T	MR, LD _T
Q ₈	Q ₈ ·I _{DF}	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC
Q ₉	Q ₉ ·I _{DF}	M → MA	MR, LD _{MA}
Q _A	Q _A ·I _{DF}	M + T → R, Flags → CC	MR, f ₃ , f ₁ , LD _R , LD _{CC}
Q _B	Q _B ·I _{DF}	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC
Q _C	Q _C ·I _{DF}	M → MA	MR, LD _{MA}
Q _D	Q _D ·I _{DF} ·Z	R → M, (Next Fetch)	OE _R , 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.

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

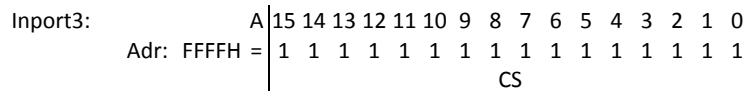
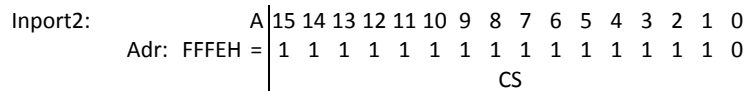
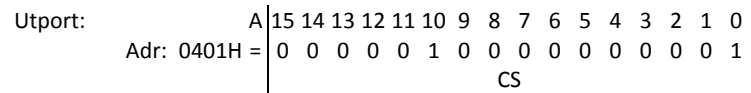
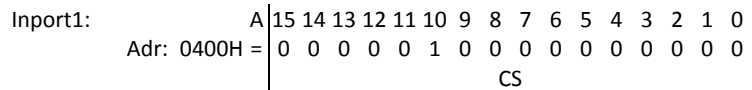
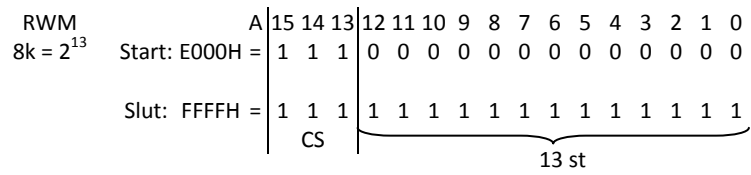
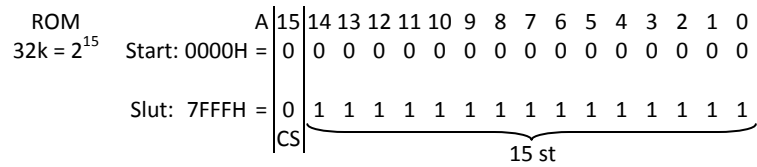
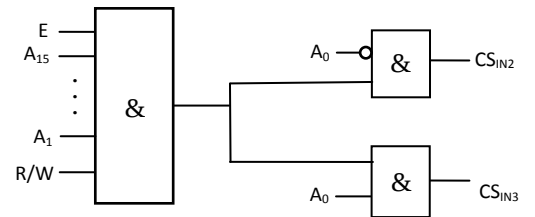
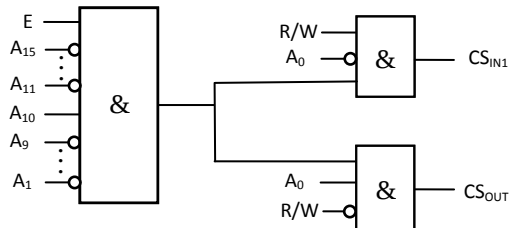
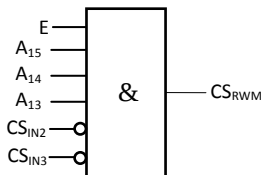
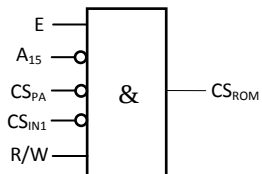
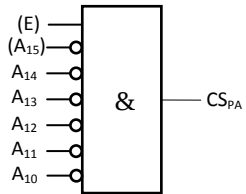
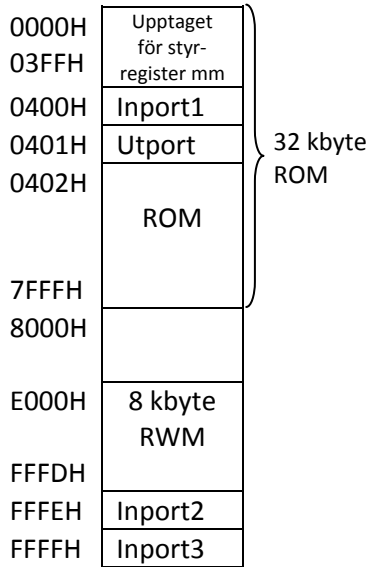
(3p)

7.

MODIFY	PSHX		Spara register på stack
	PSHY		
	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.



(8p)