

Lösningsförslag tenta 2013-01-14

(Version 1 med reservation för eventuella fel.)

1. $X = 011\ 0011_2$; $Y = 100\ 1101_2$ (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) $S = X + Y$	
$\begin{array}{r} 76543210 \\ 11111110 \\ 0110011 \\ +1001101 \\ \hline 0000000 \end{array}$	$\begin{array}{l} \text{bitnummer} \\ \text{carry} \\ X \\ Y \\ S \end{array}$
$= S$	(1p)

d) $\underline{N} = s_6 = 0$	
$\underline{Z} = 1$ ($S = 0$)	
$\underline{V} = x_6 * y_6 * s_6' + x_6' * y_6' * s_6 = 0 * 1 * 0' + 0' * 1' * 0 = 0$	
$\underline{C} = c_7 = 1$	
NZVC = 0101	(1p)

e) $D = X + Y_{1k} + 1$	
$\begin{array}{r} 76543210 \\ 01100111 \\ 0110011 \\ +0110010 \\ \hline 1100110 \end{array}$	$\begin{array}{l} \text{bitnummer} \\ \text{carry} \\ X \\ Y_{1k} \\ D \end{array}$
$= D$	(1p)

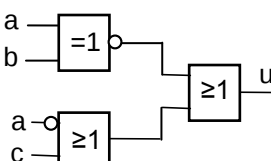
f) $\underline{N} = d_6 = 1$	
$\underline{Z} = 0$ ($D \neq 0$)	
$\underline{V} = x_6 * y_{6ik} * d_6' + x_6' * y_{6ik}' * d_6 = 0 * 0 * 1' + 0' * 0' * 1 = 1$	
$\underline{C} = c_7' = 0' = 1$	
NZVC = 1011	(1p)

g) $\underline{X} = 011\ 0011_2 = 33_{16} = 3 * 16 + 3 = \underline{51}$
 $\underline{Y} = 100\ 1101_2 = 4D_{16} = 4 * 16 + 13 = \underline{77}$
 $\underline{S} = 000\ 0000_2 = \underline{0}$ Fel eftersom $C = 1$.
 $\underline{D} = 110\ 0110_2 = 66_{16} = 6 * 16 + 6 = \underline{102}$ Fel eftersom $C = 1$ (1p)

h) ($x_6 = 0$, pos) $\underline{X} = 010100111_2 = \underline{51}$
 ($y_6 = 1$, neg) $Y_{2k} = 2^7 - 77 = 128 - 77 = 51$ $\underline{Y}$ motsvarar $\underline{-51}$
 ($s_6 = 0$, pos) $\underline{S} = 000\ 0000_2 = \underline{0}$ Ok eftersom $V = 0$.
 ($d_6 = 1$, neg) $D_{2k} = 2^7 - 102 = 128 - 102 = \underline{26}$ $\underline{D}$ motsvarar $\underline{-26}$. Fel eftersom $V = 1$ (1p)

i) Denna kod används när man kodar decimala siffror för "mekanisk" avläsning. Man eliminerar risken för felavläsning av de decimala siffrorna (4 bitar per sifferposition, BCD) på gränsen mellan två kodord eftersom koden för motsvarande siffra i angränsande kodord skiljer sig åt i endast en bitposition. (2p)

j) Antal siffror i mantissan = $23 + 1 = 24$.
 Antal olika värden hos mantissan = $2^{24} = 16 * 2^{20} \approx 16 * (10^3)^2 = 1,6 * 10^7$ vilket ger minst 7 siffror. (2p)

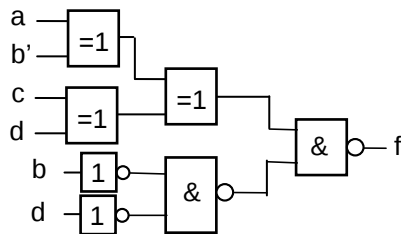
k) 

$$\underline{u} = (a'b' + ab) + a' + c = a'b' + ab + a' + c = a' + b + c$$

		bc			
		00	01	11	10
a	0	1	1	1	1
	1	0	1	1	1

(3p)

2. $\underline{f} = b'd' + (a \oplus b \oplus c \oplus d)$

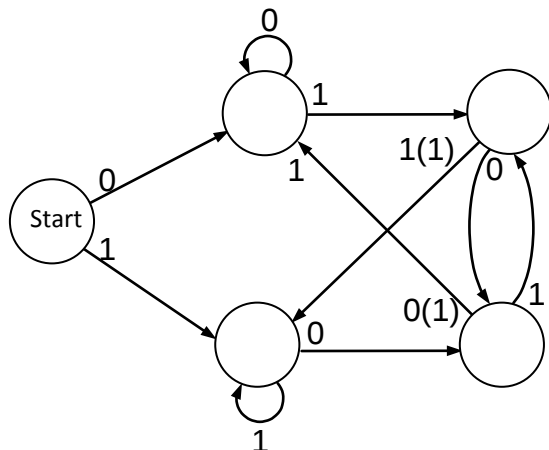


		cd			
		00	01	11	10
ab	00	1	1	0	1
	01	1	0	1	0
	11	0	1	0	1
	10	1	0	1	1

(4p)

3.

a)



(Ej angivna utsignaler = 0.)

Antal tillstånd = 5 vilket ger 3 st vipor.
($2^2 < 5 \leq 2^3$)

(5p)

b)

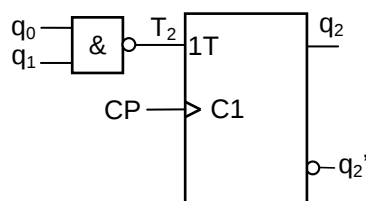
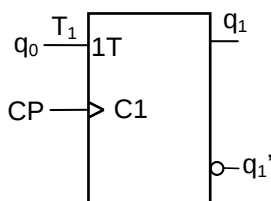
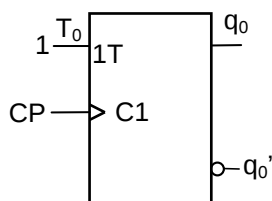
För T-vipor gäller att $q^+ = q$ för $T = 0$ och $q^+ = q'$ för $T = 1$.

q_2	q_1	q_0	q_2^+	q_1^+	q_0^+	T_2	T_1	T_0
0	0	0	1	0	1	1	0	1
0	0	1	1	1	0	1	1	1
0	1	0	1	1	1	1	0	1
0	1	1	0	0	0	0	1	1
1	0	0	0	0	1	1	0	1
1	0	1	0	1	0	1	1	1
1	1	0	0	1	1	1	0	1
1	1	1	1	0	0	0	1	1

	T_2 q_1q_0			
	00	01	11	10
q_2 0	1	1	0	1
1	1	1	0	1

	T_1 q_1q_0			
	00	01	11	10
q_2 0	0	1	1	0
1	0	1	1	0

$T_2 = q_1' + q_0'$; $T_1 = q_0$; $T_0 = 1$.



(6p)

4.

CP	Styrsignaler (=1)	RTN	Reg A	Reg B	Reg T	Reg R
1	OE_B, LD_T	$B \rightarrow T$	32	1E	?	?
2	OE_A, f_3, f_1, LD_R	$A + T \rightarrow R$	32	1E	1E	?
3	$OE_R, f_3, f_1, f_0, LD_R$	$2R \rightarrow R$	32	1E	1E	50
4	$OE_R, f_3, f_1, f_0, LD_R$	$2R \rightarrow R$	32	1E	1E	A0
5	$OE_R, f_3, f_2, g_0, LD_R$	$R - T \rightarrow R$	32	1E	1E	40
6	OE_R, LD_A	$R \rightarrow A$	32	1E	1E	22
7	?		22	1E	1E	22

(4p)

5. a)

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q ₅	Q ₅ ·I _{xx}	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC
Q ₆	Q ₆ ·I _{xx}	M → T	MR, LD _T
Q ₇	Q ₇ ·I _{xx}	X + T → R	OE _X , f ₃ , f ₁ , LD _R
Q ₈	Q ₈ ·I _{xx}	R → MA	OE _R , LD _{MA}
Q ₉	Q ₉ ·I _{xx}	A → M, (Next Fetch)	OE _A , MW, NF

Från början pekar PC på minnesordet efter OP-koden.

Q₅: Minnet adresseras med innehållet i PC. PC ökas sedan med ett.

Q₆: Dataordet efter OP-koden laddas i T-reg.

Q₇: X och dataordet efter OP-koden adderas. Summan laddas i R.

Q₈: Minnet adresseras med summan.

Q₉: Innehållet i A-reg skrivs i minnet.

Detta är STAA n,X

(2p)

b)

State	S-term	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	Q ₅ ·I _{FD}	U = X, Flags → CC	OE _X , f ₀ , LD _{CC}
Q ₆	Q ₆ ·I _{FD}	PC → MA, T, PC + 1 → PC	OE _{PC} , LD _{MA} , LD _T , IncPC
Q ₇	Q ₇ ·I _{FD} ·Z Q ₇ ·I _{FD} ·Z'	If Z = 1: M + T + 1 → R else: Next Fetch	MR, f ₃ , f ₁ , g ₀ , LD _R NF
Q ₈	Q ₈ ·I _{FD}	R → PC, Next Fetch	OE _R , LD _{PC} , NF

OPKOD
Offset

(5p)

6.

a) Antalet tillstånd hos 4-bitarsräknaren i styrenheten är $2^4 = 16$. RESET och FETCH tar tillsammans fem tillstånd. De övriga 11 tillstånden kan användas av EXECUTE. (2p)

b) Operationskoden är ett 8-bitars ord som kan ha $2^8 = 256$ olika värden. Antalet olika instruktioner är då ≤ 256 . (1p)

c) Subtraktionen $A6_{16} - 75_{16}$ ger flaggsättningen NZVC = 0010.

BPL använder hoppvillkoret N = 0 och hopp utförs alltså.

BLE använder hoppvillkoret $(N \oplus V) + Z = 1$. I detta fall är $N \oplus V = 1$ och hopp utförs. (2p)

d)

Adr (Hex)	Data (Hex)	~	Läge		
-			BVAL	EQU 2	
-				ORG \$30	
30	6C	5	DELAY	PSHA	
31	6D	5		PSHB	
32	6F	5		PSHX	
33	11 42	4		LDX #DETAB	
35	83	6		LDAA A,X	
36	10 02	4	LOOPA	LDAB #BVAL	
38	45	4	LOOPB	DECB	
39	5E FD	5		BNE LOOPB	38-3B=FD
3B	44	4		DECA	
3C	5E F8	5		BNE LOOPA	36-3E=F8
3E	73	4	DEXIT	PULX	
3F	71	4		PULB	
40	70	4		PULA	
41	6A	4		RTS	
42	05 0A 14 32 64 C8	-	DETAB	FCB 5,10,20,50,100,200	

(3p)

e) $T = 7+5+5+5+4+6+(4+(4+5)*2+4+5)*100+4+4+4+4 = 48+(13+18)*100 = 3148$ klockpulser (μs) (3p)

f)

Adr (Hex)	Data (Hex)		
-		ORG	\$80
80	0B 41	LDAA	\$61 Hämta PLB
82	30 43	SUBA	\$63 Subtrahera QLB
84	13 45	STAA	\$65 Lagra RLB
86	0B 40	LDAA	\$60 Hämta PHB
88	34 42	SBCA	\$62 Subtrahera QHB och borrow
8A	13 44	STAA	\$64 Lagra RHB

(3p)

7.

Alternativ 1:

Lösning med given tabell Bin -> Gray

GRAY2BIN

	PSHB		Spara reg på stack
	PSHX		
	LDX	#GTAB	Pekare till tabell Bin -> Gray
	CLRB		Räknare för tabellindex
	ANDA	#\$0F	Behåll bitarna 3-0
LOOP	CMPA	,X	Jämför indata med graykod i tabell
	BEQ	GREXIT	Träff
	INX		Ej träff. Fortsätt med nästa
	INCB		Öka tabellindex
	BRA	LOOP	Testa nästa
GREXIT	TFR	B,A	Tabellindex = utdata
	PULX		Återställ reg
	PULB		
	RTS		Retur med binärvärde i A-reg
GTAB	FCB		0,1,3,2,6,7,5,4,12,13,15,14,10,11,9,8

Alternativ 2:

Lösning med ny tabell Gray -> Bin

GRAY2BIN

	PSHX		Spara reg på stack
	LDX	#BTAB	Pekare till tabell Gray-bin
	ANDA	#\$0F	Behåll bitarna 3-0
	LDAA	A,X	Hämta graykodens binära motsv
	PULX		Återställ reg
	RTS		Retur med binärvärde i A-reg
BTAB	FCB		0,1,3,2,7,6,4,5,15,14,12,13,8,9,11,10

(5p)