

Lösningsförslag tenta 2012-05-24 (Anpassad för FLISP)

(Version 2013-05-23 med reservation för eventuella fel.)

1. $X = 10011_2$; $Y = 01101_2$ (5 bitars ordlängd)

a) $[0, 2^n - 1] = [0, 2^5 - 1] = [0, 31]$ (1p)

b) $[-2^{n-1}, +2^{n-1} - 1] = [-2^{5-1}, +2^{5-1} - 1] = [-16, +15]$ (1p)

c) $S = X + Y$

$$\begin{array}{r} 543210 \text{ bitnummer} \\ 111110 \text{ carry} \\ 10011 \text{ X} \\ +01101 \text{ Y} \\ \hline 00000 = \underline{S} \end{array} \quad (1p)$$

d) $\underline{N} = s_4 = \underline{0}$
 $\underline{Z} = \underline{1} (S = 0)$
 $\underline{V} = x_4 * y_4 * s_4' + x_4' * y_4' * s_4 = 1 * 0 * 0' + 1' * 0' * 0 = \underline{0}$
 $\underline{C} = c_5 = \underline{1}$ (1p)

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

$$\begin{array}{r} 543210 \text{ bitnummer} \\ 100111 \text{ carry} \\ 10011 \text{ X} \\ +10010 \text{ Y}_{1k} \\ \hline 00110 = \underline{D} \end{array} \quad (1p)$$

f) $\underline{N} = d_4 = \underline{0}$
 $\underline{Z} = \underline{0} (D \neq 0)$
 $\underline{V} = x_4 * y_{4ik} * d_4' + x_4' * y_{4ik}' * d_4 = 1 * 1 * 0' + 1' * 1' * 0 = \underline{1}$
 $\underline{C} = c_5' = 1' = \underline{0}$ (1p)

g) $\underline{X} = 10011_2 = 13_{16} = 1 * 16 + 3 = \underline{19}$
 $\underline{Y} = 01101_2 = 0D_{16} = 0 * 16 + 13 = \underline{13}$
 $\underline{S} = 00000_2 = \underline{0}$ Resultatet S är felaktigt eftersom $C = 1$.
 $\underline{D} = 00110_2 = 06_{16} = 0 * 16 + 6 = \underline{6}$ Resultatet D är korrekt eftersom $C = 0$. (1p)

h) $(x_4 = 1, \text{neg}) \quad X_{2k} = 2^5 - 19 = 32 - 19 = 13 \quad \underline{X}$ motsvarar $\underline{-13}$
 $(y_4 = 0, \text{pos}) \quad \underline{Y} = 01101_2 = \underline{13}$
 $(s_4 = 0, \text{pos}) \quad \underline{S} = 00000_2 = \underline{0}$ Resultatet S är korrekt eftersom $V = 0$.
 $(d_4 = 0, \text{pos}) \quad \underline{D} = 00110_2 = \underline{6}$ Resultatet D är felaktigt eftersom $V = 1$. (1p)

i) Eftersom 9-komplementet används vid 10-komplementaritmetik (subtraktion och teckenbyte). (1p)

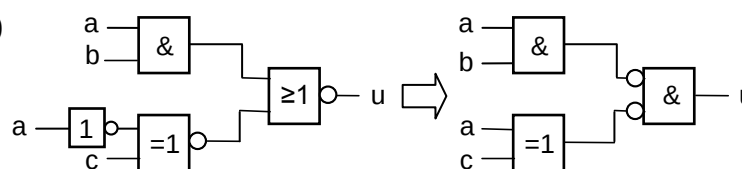
j) $N_{\text{flyt}} = C3C8D000_{16} = \begin{array}{ccc} 1/100 & 0011 & 1/100 \ 1000 \ 1101 \ 0000 \ 0000 \ 0000 \\ s & c & f \end{array}$

$s = 1 (-)$

$c = 135$; $\text{exp} = 135 - 127 = 8$;

$m = 1.f = 1.100 \ 1000 \ 1101 \ 0000 \ 0000 \ 0000$

$\underline{N}_2 = -1.100 \ 1000 \ 1101 \ 0000 \ 0000 \ 0000 * 2^8 = -1 \ 1001 \ 0001.101 = -(256+144+1+0,625) = \underline{-401,625}$ (2p)

k) 

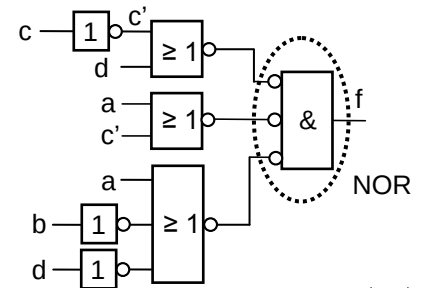
$$u = (ab)'(ac+a'c') = (a'+b')(ac+a'c') = a'c' + ab'c + a'b'c' \text{ som ger k-diagram:}$$

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

$$\underline{u} = (a+c')(a'+c)(b'+c') = (a+c')(a'+c)(a'+b')$$
 (4p)

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

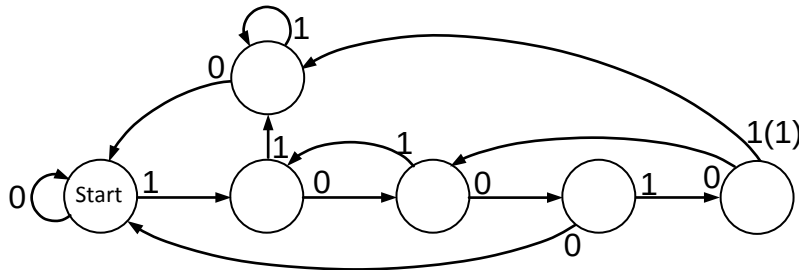
		cd			
		00	01	11	10
ab	00	1	1	0	0
	01	1	0	-	0
	11	-	1	-	0
	10	1	-	1	0



(4p)

3.

a)



6 tillstånd ger minst 3 vippor
($2^2 < 6 \leq 2^3$)

(Ej utsatta utsignaler = 0)

(4p)

b)

				D_2	D_1	D_0
q_2	q_1	q_0	q_2^+	q_1^+	q_0^+	
0	0	0	0	0	1	
0	0	1	0	1	1	
0	1	0	1	1	0	
0	1	1	0	1	0	
1	0	0	0	0	0	
1	0	1	1	0	0	
1	1	0	1	1	1	
1	1	1	1	0	1	

För D-vippor gäller att $q^+ = D$.

		D_2			
		q_1q_0			
		00	01	11	10
q_2	0	0	0	0	1
	1	0	1	1	1

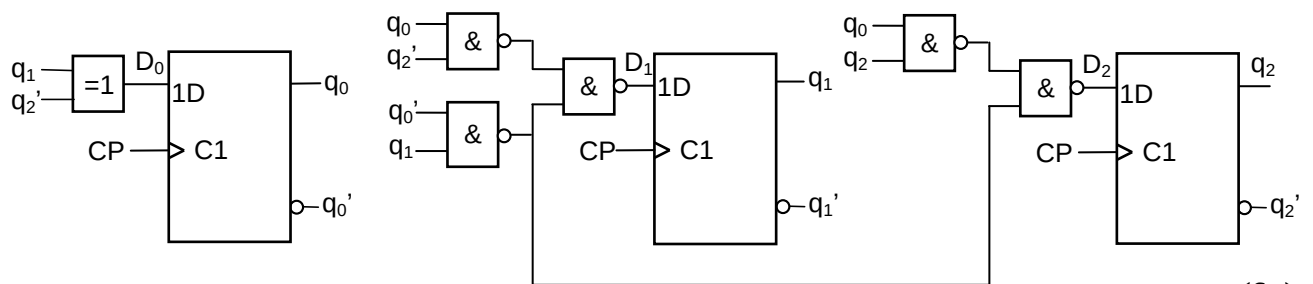
		D_1			
		q_1q_0			
		00	01	11	10
q_2	0	0	1	1	1
	1	0	0	0	1

D_0		q_1q_0			
		00	01	11	10
q_2	0	1	1	0	0
	1	0	0	1	1

$$D_2 = q_2q_0 + q_1q_0'$$

$$D_1 = q_2'q_0 + q_1q_0'$$

$$D_0 = q_2'q_1' + q_2q_1 = (q_2 \oplus q_1)'$$



(6p)

4.

CP	Styrsignaler (=1)	RTN	Reg A	Reg B	Reg T	Reg R
1	OE_B, LD_T	$B \rightarrow T$	1B	32	?	?
2	$OE_A, f_3, f_1, g_0, LD_R$	$A+T+1 \rightarrow R$	1B	32	32	?
3	$OE_R, f_3, f_1, f_0, LD_R, LD_B$	$2R \rightarrow R, R \rightarrow B$	1B	32	32	4E
4	OE_R, LD_T	$R \rightarrow T$	1B	4E	32	9C
5	$OE_A, f_3, f_2, g_0, LD_R$	$A - T \rightarrow R$	1B	4E	9C	9C
6	OE_R, f_2, f_1, LD_R	$R \text{ AND } T \rightarrow R$	1B	4E	9C	7F
7	OE_R, LD_A	$R \rightarrow A$	1B	4E	9C	1C
8	?		1C	4E	9C	1C

(4p)

5. a)

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q ₄	Q ₄ ·I _{xx}	PC → TA, T	OE _{PC} , LD _T , LD _{TA}
Q ₅	Q ₅ ·I _{xx}	M(TA)+T+1 → R, PC+1 → PC	MR, f ₃ , f ₁ , f ₀ , g ₀ , g ₁₄ , LD _R , INC _{PC}
Q ₆	Q ₆ ·I _{xx} ·(N⊕V)+Z)	If [(N⊕V) + Z] = 0: R → PC	OE _R , LD _{PC}
	Q ₆ ·I _{xx}	(Next Fetch)	NF

Från början pekar PC på minnesordet efter OP-koden.

Q₄: Minnesadressregistret TA och register T laddas med innehållet i PC.

Q₅: Dataordet efter OP-koden adderas med värdet från PC plus 1. Summan laddas i R. PC ökas.

Q₆: Om ((N⊕V)+Z) = 0 laddas PC från R, dvs. ett hopp utförs. Annars fortsätter processorn med instruktionen efter den nuvarande i minnet. Efter Q₆ utförs alltid FETCH av nästa instruktion.

Detta är ett villkorligt branch. BGT Adr enligt instruktionslistan.

(2p)

b)

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) _{1k} → R	g ₁₄ , MR, f ₂ , f ₀ , LD _R
Q ₆	Q ₆ ·I _{DF}	R → T	OE _R , LD _T
Q ₇	Q ₇ ·I _{DF}	M(PC) AND T, ALU(Z) → CC, PC+1 → PC	MR, f ₂ , f ₁ , f ₀ , g ₂ , g ₃ , g ₄ , g ₅ , g ₈ , g ₉ , g ₁₀ , g ₁₁ , LD _{CC} , INC _{PC}
Q ₈	Q ₈ ·I _{DF}	PC → T, TA, PC+1 → PC	OE _{PC} , LD _T , LD _{TA} , INC _{PC}
Q ₉	Q ₉ ·I _{DF} ·Z)	If Z = 0: (Next Fetch)	NF
	Q ₉ ·I _{DF} ·(·Z)	M(TA) + T + 1 → R	g ₁₄ , MR, f ₃ , f ₁ , f ₀ , g ₀ , LD _R
Q _A	Q _A ·I _{DF}	R → PC, (Next Fetch)	OE _R , LD _{PC} , NF

OPKOD

Adr1

mask

offset

(6p)

6.

a) Additionen $87_{16} + A3_{16} = 2A_{16}$ ger flaggvärdena NZVC = 0011. Detta medför att hoppet BMI inte utförs (N = 0) och att hoppet BLT utförs eftersom $N \oplus V = 0 \oplus 1 = 1$

(3p)

b)

Adr	Data	~	Läge		
80	10	3	TIME	PSHA	
81	11	3		PSHX	
82	90 F6	2	LOOP	LDX #XVAL	XVAL = -10 (F6)
84	CC 01	4	LOOPX	LEAX 1,X	
86	9C 00	3		CMPX #0	
88	25 FA	4		BNE LOOPX	84 - 8A = FA
8A	08	3		DECA	
8B	25 F5	4		BNE LOOP	82 - 8D = F5
8D	15	3	TEXIT	PULX	
8E	14	3		PULA	
8F	43	2		RTS	

(3p)

c) JSR TIME tar 4 klockpulser.

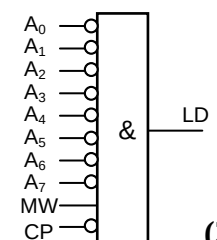
A-reg innehåller värdet 5 vid anropet, så yttre slingan (LOOP) utförs 5 gånger.

$\underline{N} = 4+3+3+(2+(4+3+4)*10+3+4)*5+3+3+2 = 18+(9+11*10)*5 = 613$ klockpulser (μs)

(3p)

d) Utporten skall laddas från databussen vid skrivning på adress $00_{16} = 00000000_2$

(Ring på ingången i figuren till höger betyder att signalen inverteras innan den når OCH-grunden.)



(2p)

7.

* Subrutin HXCONV

HXCONV	PSHA		Spara register på stack
	PSHX		
HXLOOP	LDA	0,X	Hämta data från textsträng
	BEQ	HXEXIT	Strängslut? Ja, återhopp
	ANDA	#\$7F	Nej, maska bort bit 7
	SUBA	#\$30	Justera för vanlig siffra 0-9
	BLO	EJHEX	Ej hexsiffra
	CMPA	#9	Testa om <= 9
	BLS	OKHEX	Siffran är 0-9
	SUBA	#7	Justera för siffra A-F
	CMPA	#10	Nedre gräns för bokstavssiffra
	BLO	EJHEX	Ej hexsiffra
	CMPA	#15	Övre gräns för bokstavssiffra
	BHI	EJHEX	Ej hexsiffra
OKHEX	STA	,X+	Uppdatera sträng med binärt värde 0-15
	BRA	HXLOOP	Fortsätt med nästa
EJHEX	LDA	#\$80	
	STA	,X+	Uppdatera sträng med markör för ej hex
	BRA	HXLOOP	Fortsätt med nästa
HXEXIT	LDA	#\$FF	
	STA	0,X	Skriv nytt strängslut \$FF
	PULX		Återställ register
	PULA		
	RTS		

(8p)