

Lösningsförslag tenta 2012-05-24

(Version 3 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) $S = X + Y$

543210	bitnummer
111110	carry
10011	X
+01101	Y
00000	S

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

543210	bitnummer
100111	carry
10011	X
+10010	Y _{1k}
00110	D

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

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

j) $N_{\text{flyt}} = C3C8D000_{16} =$

1/100	0011	1/100	1000	1101	0000	0000	0000
s	c			f			

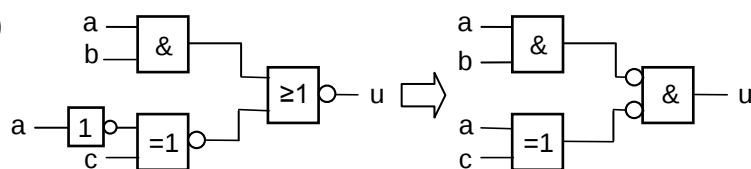
$s = 1 (-)$

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

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

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

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

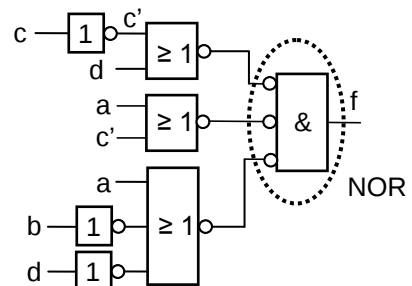


		bc			
		00	01	11	10
a	0	1	0	0	1
	1	0	1	0	0

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

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

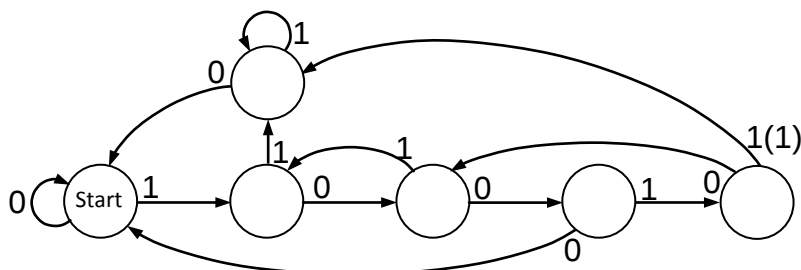
		cd			
		00	01	11	10
ab	00	1	1	0	0
	01	1	0	-	0
	11	-	1	-	0
	10	1	-	1	0



(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

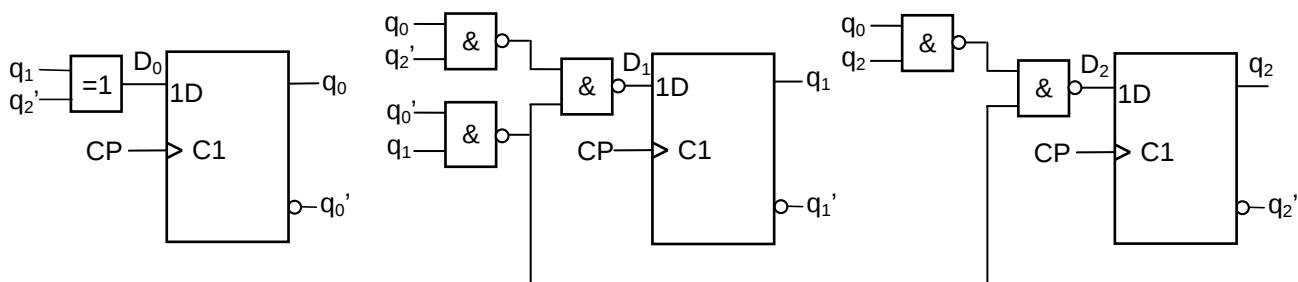
		D_1			
		q_1q_0			
		00	01	11	10
q_2	0	0	1	1	1
	1	0	0	0	1

		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 ₅	Q ₅ ·I _{xx}	PC → MA, T	OE _{PC} , LD _{MA} , LD _T
Q ₆	Q ₆ ·I _{xx}	M+T+1 → R, PC+1 → PC	MR, f ₃ , f ₁ , g ₀ , LD _R , IncPC
Q ₇	Q ₇ ·I _{xx} ·((N⊕V)+Z)	If [(N⊕V) + Z] = 1: (Next Fetch)	NF
	Q ₇ ·I _{xx} ·((N⊕V)+Z)'	else: R → PC, (Next Fetch)	OE _R , LD _{PC} , NF

Från början pekar PC på minnesordet efter OP-koden.

Q₅: Minnet adresseras med innehållet i PC. Adressen lagras också i T.

Q₆: Dataordet efter OP-koden adderas med värdet från PC plus 1. Summan laddas i R. PC ökas.

Q₇: Om ((N⊕V)+Z) = 1 utförs FETCH av nästa instruktion, annars laddas PC från R.

Detta är ett villkorligt branch. BGT Adr enligt instruktionslistan.

(2p)

b)

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q ₅	Q ₅ ·I _{FC}	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC
Q ₆	Q ₆ ·I _{FC}	M → MA	MR, LD _{MA}
Q ₇	Q ₇ ·I _{FC}	M _{1k} → R	MR, f ₁ , f ₀ , LD _R
Q ₈	Q ₈ ·I _{FC}	R → T	OE _R , LD _T
Q ₉	Q ₉ ·I _{FC}	PC → MA	OE _{PC} , LD _{MA}
Q _A	Q _A ·I _{FC}	M AND T, Flags → CC, PC+1 → PC	MR, f ₂ , f ₁ , LD _{CC} , IncPC
Q _B	Q _B ·I _{FC}	PC → MA, T	OE _{PC} , LD _{MA} , LD _T
Q _C	Q _C ·I _{FC}	M + T + 1 → R, PC+1 → PC	MR, f ₃ , f ₁ , g ₀ , LD _R , IncPC
Q _D	Q _D ·I _{FC} ·Z	If Z = 1: R → PC, (Next Fetch)	OE _R , LD _{PC} , NF
	Q _D ·I _{FC} ·Z'	else: (Next Fetch)	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	6C	5		PSHA	
81	6F	5		PSHX	
82	11 F6	4	LOOP	LDX #XVAL	XVAL = -10 (F6)
84	E1	4	LOOPX	INX	
85	5E FD	5		BNE LOOPX	84 – 87 = FD
87	44	4		DECA	
88	5E F8	5		BNE LOOP	82 – 8A = F8
8A	73	4		PULX	
8B	70	4		PULA	
8C	6A	4		RTS	

(3p)

c) JSR TIME tar 7 klockpulser.

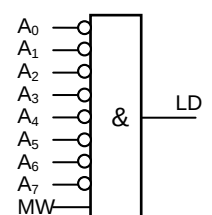
A-reg innehåller värdet 5 vid anropet, så yttre slingan (LOOP) utförs 5 gånger.

$N = 7 + 5 + 5 + (4 + (4 + 5) * 10 + 4 + 5) * 5 + 4 + 4 + 4 = 29 + (13 + 9 * 10) * 5 = 544$ 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 PSHX		Spara register på stack
HXLOOP	LDAA TSTA BEQ	,X HXEXIT	Hämta data från textsträng 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	STAA BRA	1,X+ HXLOOP	Uppdatera sträng med binärt värde 0-15 Fortsätt med nästa
EJHEX	LDAA STAA BRA	#\$80 1,X+ HXLOOP	Uppdatera sträng med markör för ej hex Fortsätt med nästa
HXEXIT	LDAA STAA PULX PULA RTS	#\$FF ,X 	Skriv nytt strängslut \$FF Återställ register

(8p)