

# Lösningförslag 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) $N = s_4 = 0$
$Z = 1$ ( $S = 0$ )
$V = x_4 * y_4 * s_4' + x_4' * y_4' * s_4 = 1 * 0 * 0' + 1' * 0' * 0 = 0$
$C = c_5 = 1$
(1p)

e) $D = X + Y_{1k} + 1$	
543210	bitnummer
100111	carry
10011	X
+10010	Y <sub>1k</sub>
00110	= D
(1p)	

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

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

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

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

j)  $N_{fl\text{yt}} = C3C8D000_{16} = \begin{matrix} 1/100 & 0011 & 1/100 & 1000 & 1101 & 0000 & 0000 & 0000 \\ s & c & & & f & & & \end{matrix}$   
 $s = 1$  (-)  
 $c = 135$ ;  $\text{exp} = 135 - 127 = 8$ ;  
 $m = 1.f = 1.100 1000 1101 0000 0000 0000$   
 $N_2 = -1.100 1000 1101 0000 0000 0000 * 2^8 = -1 1001 0001.101 = -(256+144+1+0,625) = -401,625$  (2p)

k)

$$u = (ab)'(ac+a'c') = (a'+b')(ac+a'c') = a'c' + ab'c + a'b'c'$$

som ger k-diagram:

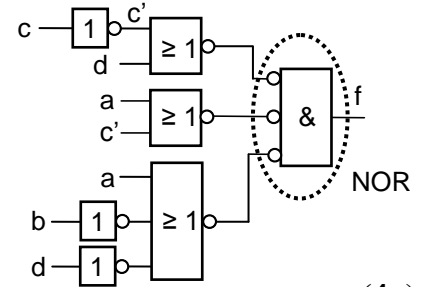
	bc			
	00	01	11	10
a	0	1	0	1
c	1	0	1	0

$$u = (a+c')(a'+c)(b'+c') = (a+c')(a'+c)(a'+b')$$

(4p)

2.  $f = (c'+d)(a+c')(a+b'+d')$

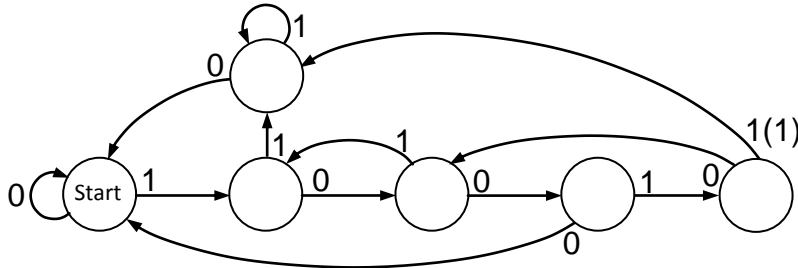
		cd			
		00	01	11	10
ab	00	1	1	0	0
	01	1	0	-	0
	11	-	1	-	0
	10	1	-	1	0



(4p)

3.

a)



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

(Ej utsatta utsignaler = 0)

(4p)

b)

			D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
q <sub>2</sub>	q <sub>1</sub>	q <sub>0</sub>	q <sub>2</sub> <sup>+</sup>	q <sub>1</sub> <sup>+</sup>	q <sub>0</sub> <sup>+</sup>
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  $q^+ = D$ .

		q <sub>1</sub> q <sub>0</sub>			
		00	01	11	10
q <sub>2</sub>	0	0	0	0	1
	1	0	1	1	1

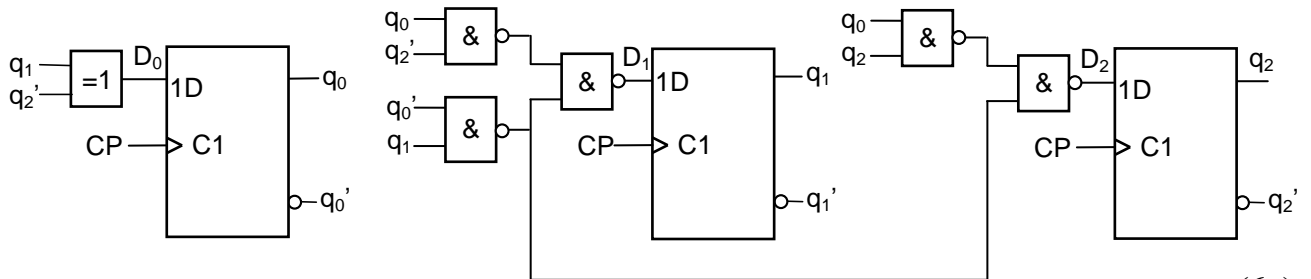
		q <sub>1</sub> q <sub>0</sub>			
		00	01	11	10
q <sub>2</sub>	0	0	1	1	1
	1	0	0	0	1

		q <sub>1</sub> q <sub>0</sub>			
		00	01	11	10
q <sub>2</sub>	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	Styr signaler (=1)	RTN	Reg A	Reg B	Reg T	Reg R
1	OE <sub>B</sub> , LD <sub>T</sub>	B → T	1B	32	?	?
2	OE <sub>A</sub> , f <sub>3</sub> , f <sub>1</sub> , g <sub>0</sub> , LD <sub>R</sub>	A+T+1 → R	1B	32	32	?
3	OE <sub>R</sub> , f <sub>3</sub> , f <sub>1</sub> , f <sub>0</sub> , LD <sub>R</sub> , LD <sub>B</sub>	2R → R, R → B	1B	32	32	4E
4	OE <sub>R</sub> , LD <sub>T</sub>	R → T	1B	4E	32	9C
5	OE <sub>A</sub> , f <sub>3</sub> , f <sub>2</sub> , g <sub>0</sub> , LD <sub>R</sub>	A - T → R	1B	4E	9C	9C
6	OE <sub>R</sub> , f <sub>2</sub> , f <sub>1</sub> , LD <sub>R</sub>	R AND T → R	1B	4E	9C	7F
7	OE <sub>R</sub> , LD <sub>A</sub>	R → A	1B	4E	9C	1C
8	?		1C	4E	9C	1C

(4p)

5. a)

State	S-term	RTN-beskrivning	Styr signaler (=1)
Q <sub>5</sub>	Q <sub>5</sub> 'I <sub>xx</sub>	PC → MA, T	OE <sub>PC</sub> , LD <sub>MA</sub> , LD <sub>T</sub>
Q <sub>6</sub>	Q <sub>6</sub> 'I <sub>xx</sub>	M+T+1 → R, PC+1 → PC	MR, f <sub>3</sub> , f <sub>1</sub> , g <sub>0</sub> , LD <sub>R</sub> , IncPC
Q <sub>7</sub>	Q <sub>7</sub> 'I <sub>xx</sub> '((N⊕V)+Z)	If [(N⊕V) + Z] = 1: (Next Fetch)	NF
	Q <sub>7</sub> 'I <sub>xx</sub> '((N⊕V)+Z)'	else: R → PC, (Next Fetch)	OE <sub>R</sub> , LD <sub>PC</sub> , NF

Från början pekar PC på minnesordet efter OP-koden.

Q<sub>5</sub>: Minnet adresseras med innehållet i PC. Adressen lagras också i T.

Q<sub>6</sub>: Dataordet efter OP-koden adderas med värdet från PC plus 1. Summan laddas i R. PC ökas.

Q<sub>7</sub>: 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	Styr signaler (=1)
Q <sub>5</sub>	Q <sub>5</sub> 'I <sub>FC</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>FC</sub>	M → MA	MR, LD <sub>MA</sub>
Q <sub>7</sub>	Q <sub>7</sub> 'I <sub>FC</sub>	M <sub>1k</sub> → R	MR, f <sub>1</sub> , f <sub>0</sub> , LD <sub>R</sub>
Q <sub>8</sub>	Q <sub>8</sub> 'I <sub>FC</sub>	R → T	OE <sub>R</sub> , LD <sub>T</sub>
Q <sub>9</sub>	Q <sub>9</sub> 'I <sub>FC</sub>	PC → MA	OE <sub>PC</sub> , LD <sub>MA</sub>
Q <sub>A</sub>	Q <sub>A</sub> 'I <sub>FC</sub>	M AND T, Flags → CC, PC+1 → PC	MR, f <sub>2</sub> , f <sub>1</sub> , LD <sub>CC</sub> , IncPC
Q <sub>B</sub>	Q <sub>B</sub> 'I <sub>FC</sub>	PC → MA, T	OE <sub>PC</sub> , LD <sub>MA</sub> , LD <sub>T</sub>
Q <sub>C</sub>	Q <sub>C</sub> 'I <sub>FC</sub>	M + T + 1 → R, PC+1 → PC	MR, f <sub>3</sub> , f <sub>1</sub> , g <sub>0</sub> , LD <sub>R</sub> , IncPC
Q <sub>D</sub>	Q <sub>D</sub> 'I <sub>FC</sub> 'Z	If Z = 1: R → PC, (Next Fetch)	OE <sub>R</sub> , LD <sub>PC</sub> , NF
	Q <sub>D</sub> 'I <sub>FC</sub> '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.

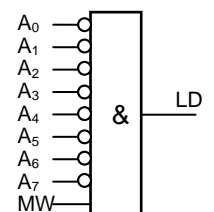
A-reg innehåller värdet 5 vid anropet, så yttre slingan (LOOP) utförs 5 gånger.

$$N = 7 + 5 + 5 + (4 + (4 + 5) * 10 + 4 + 5) * 5 + 4 + 4 + 4 = 29 + (13 + 9 * 10) * 5 = \underline{544 \text{ klockpulser } (\mu\text{s})}$$

(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	LDA TSTA BEQ	,X HXEXIT	Hämta data från textsträng Strängslut? Ja, återhopp
	ANDA SUBA BLO	#\$7F #\$30 EJHEX	Nej, maska bort bit 7 Justera för vanlig siffra 0-9 Ej hexsiffra
	CMPA BLS	#9 OKHEX	Testa om <= 9 Siffran är 0-9
	SUBA CMPA BLO	#7 #10 EJHEX	Justera för siffra A-F Nedre gräns för bokstavssiffra Ej hexsiffra
	CMPA BHI	#15 EJHEX	Övre gräns för bokstavssiffra Ej hexsiffra
OKHEX	STAA BRA	1,X+ HXLOOP	Uppdatera sträng med binärt värde 0-15 Fortsätt med nästa
EJHEX	LDA STAA BRA	#\$80 1,X+ HXLOOP	Uppdatera sträng med markör för ej hex Fortsätt med nästa
HXEXIT	LDA STAA PULX PULA RTS	#\$FF ,X	Skriv nytt strängslut \$FF Återställ register

(8p)