

# Lösningförslag tenta 2014-05-27

(Version 3 med reservation för eventuella fel. **Rapportera gärna fel, så uppdateras lösningen efterhand!** Uppdaterad 140529.)

1.  $X = 0110_2$ ;  $Y = 1011_2$  (4 bitars ordlängd)

a)  $[0, 2^n - 1] = [0, 2^4 - 1] = [0, 15]$  (1p)

b)  $[-2^{n-1}, +2^{n-1} - 1] = [-2^{4-1}, +2^{4-1} - 1] = [-8, +7]$  (1p)

c)  $S = X + Y$

43210	bitnummer
11100	carry
0110	X
+1011	Y
0001	= S

(1p)

d)  $N = s_3 = 0$   
 $Z = 0$  ( $S \neq 0$ )  
 $V = x_3 * y_3 * s_3' + x_3' * y_3' * s_3 = 0 * 1 * 0' + 0' * 1' * 0 = 0$   
 $C = c_4 = 1$

NZVC = 0001 (1p)

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

43210	bitnummer
01001	carry
0110	X
+0100	Y <sub>1k</sub>
1011	= D

(1p)

f)  $N = d_3 = 1$   
 $Z = 0$  ( $D \neq 0$ )  
 $V = x_3 * y_3 * d_3' + x_3' * y_3' * d_3 = 0 * 0 * 1' + 0' * 0' * 1 = 1$   
 $C = c_4' = 0' = 1$

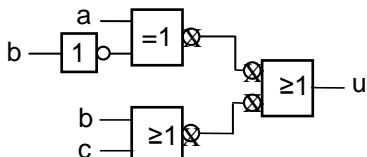
NZVC = 1011 (1p)

g)  $X = 0110_2 = 6_{16} = \underline{6}$   
 $Y = 1011_2 = B_{16} = \underline{11}_{10}$   
 $S = 0001_2 = \underline{1}$  Resultatet S är felaktigt eftersom  $C = 1$ .  
 $D = 1011_2 = \underline{11}_{10}$  Resultatet D är felaktigt eftersom  $C = 1$ . (1p)

h) ( $x_3 = 0$ , pos)  $X = \underline{6}$ .  
 ( $y_3 = 1$ , neg)  $Y_{2k} = 2^4 - 11 = 5$   $Y$  motsvarar  $\underline{-5}$ .  
 ( $s_3 = 0$ , pos)  $S = 1$  Resultatet S är korrekt eftersom  $V = 0$ .  
 ( $d_3 = 1$ , neg)  $D_{2k} = 2^4 - 11 = 5$   $D$  motsvarar  $\underline{-5}$ . Resultatet D är felaktigt eftersom  $V = 1$ . (1p)

i) Upplösningen beror av antalet bitar i mantissan  $52 + 1 = 53$ .  
 $2^{53} = 2^3 \cdot 2^{50} = 8 \cdot (2^{10})^5 \approx 8 \cdot (10^3)^5 = 8 \cdot 10^{15}$  (Antal olika kombinationer uttryckt decimalt.)  
 ( $1 \leq 8 < 10$ ) Alltså är upplösningen minst 15 decimala siffror. (2p)

2.

a) 

$$\begin{aligned} u &= (a \oplus b)' + b + c = \\ &= a'b' + ab + b + c = \\ &= a' + b + c \text{ (en maxterm)} \end{aligned} \quad \text{(Nr 4)}$$

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

(2p)

b) 

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

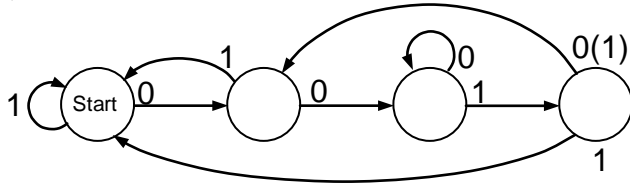
$$f = [d + (a \oplus b \oplus c)](a + d') \quad \text{(Nr 3)}$$

NOR-gindar  $\leftrightarrow$  "nollor"

(4p)

3.

a)



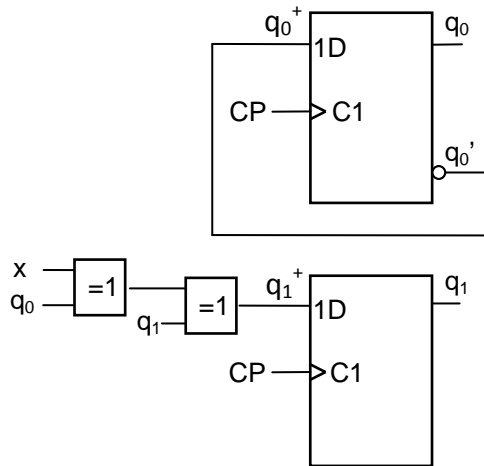
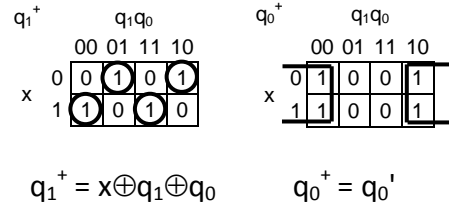
(Ej utsatta utsignaler = 0)

4 tillstånd ger minst 2 vipor  
( $2^2 = 4$ )

(4p)

b)

x	q <sub>1</sub>	q <sub>0</sub>	q <sub>1</sub> <sup>+</sup>	q <sub>0</sub> <sup>+</sup>
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0



(6p)

4.

CP	Styrsignaler (=1)	RTN	A(8)	B(8)	T(8)	R(8)
1	OE <sub>B</sub> , LD <sub>T</sub>	B → T	64	A0	?	?
2	OE <sub>A</sub> , f <sub>3</sub> , f <sub>1</sub> , LD <sub>R</sub> , LD <sub>T</sub>	A + T → R, A → T	64	A0	A0	?
3	OE <sub>R</sub> , f <sub>3</sub> , f <sub>1</sub> , f <sub>0</sub> , g <sub>0</sub> , LD <sub>R</sub>	2R + 1 → R	64	A0	64	04
4	OE <sub>R</sub> , f <sub>3</sub> , f <sub>2</sub> , LD <sub>R</sub> , LD <sub>T</sub>	R - T - 1 → R, R → T	64	A0	64	09
5	OE <sub>R</sub> , f <sub>3</sub> , f <sub>1</sub> , LD <sub>R</sub>	R + T → R	64	A0	09	A4
6	OE <sub>R</sub> , LD <sub>B</sub>	R → B	64	A0	09	AD
7	?		64	AD	09	AD

(4p)

5. a)

State	Summaterm	RTN-beskrivning	Styrsignaler
Q <sub>4</sub>	Q <sub>4</sub> ·I <sub>xx</sub>	M(PC) → T, PC+1 → PC	MR, LD <sub>T</sub> , INC <sub>PC</sub>
Q <sub>5</sub>	Q <sub>5</sub> ·I <sub>xx</sub>	M(SP+T) → T	MR, g <sub>12</sub> , LD <sub>T</sub>
Q <sub>6</sub>	Q <sub>6</sub> ·I <sub>xx</sub>	A - T + C' → R, ALU(NZVC) → CC	OE <sub>A</sub> , f <sub>3</sub> , f <sub>2</sub> , g <sub>1</sub> , g <sub>0</sub> , LD <sub>R</sub> , LD <sub>CC</sub>
Q <sub>7</sub>	Q <sub>7</sub> ·I <sub>xx</sub>	R → A, (New Fetch)	OE <sub>R</sub> , LD <sub>A</sub> , NF

Instruktionen är: SBCA n,SP

(2p)

b)

State	S-term	RTN-beskrivning	Styrsignaler
Q <sub>4</sub>	Q <sub>4</sub> ·I <sub>EF</sub>	M(PC) → TA, PC+1 → PC	MR, LD <sub>TA</sub> , INC <sub>PC</sub>
Q <sub>5</sub>	Q <sub>5</sub> ·I <sub>EF</sub>	M(TA) → T	MR, g <sub>14</sub> , LD <sub>T</sub>
Q <sub>6</sub>	Q <sub>6</sub> ·I <sub>EF</sub>	M(PC) OR T → R, ALU(NZ) → CC, 0 → V, PC+1 → PC	MR, f <sub>2</sub> , f <sub>1</sub> , LD <sub>R</sub> , g <sub>5</sub> , g <sub>3</sub> , g <sub>2</sub> , LD <sub>CC</sub> , INC <sub>PC</sub>
Q <sub>7</sub>	Q <sub>7</sub> ·I <sub>EF</sub>	R → M(TA), (New Fetch)	OE <sub>R</sub> , g <sub>14</sub> , MW, NF

OPKOD
Adr
mask

(5p)

6.

a) Båda talen har ordlängden 8 bitar och därmed talområdet  $[-128_{10}, +127_{10}]$ , eftersom hoppvillkoret för BPL avser tal med tecken.

Det villkorliga hoppet BPL Adr utförs om N-flaggan = 0, dvs. om additionen ger ett resultat  $\geq 0$ . Värdet  $7A_{16} = 122_{10}$  och motsvarar +122.

Hoppvillkoret blir:  $122 + W \geq 0$ ;  $W \geq -122$ ;  $-122 \leq W \leq 127$

Overflow inträffar dock om  $122 + W \geq 128$  dvs.  $W \geq 128 - 122 = 6$ .

I detta fall blir N = 1 och hoppet utförs ej.

Hoppet utförs alltså om W tillhör intervallet  $-122 \leq W < 6$ .

Dela upp intervallet i ett positivt och ett negativt intervall.

$0 \leq W < 6$  och  $-122 \leq W \leq -1$

Eftersom 2-komplementrepresentation används för negativa tal tar vi fram de verkliga värdena (= maskintalen) för dessa.

$256 - 122 \leq W \leq 256 - 1$  som ger  $134 \leq W \leq 255$

Hoppet utförs alltså i de två intervallen:  $0 \leq W < 6$  och  $134 \leq W \leq 255$

(4p)

6.(forts.)

b)

Adr	Data	~	Läge	Assemblerkod	Offset
				Operation	Operand
				ORG	\$30
30	91 0A	2	START	LDY	#10
32	90 41	2		LDX	#TAB
34	F5	4	LOOP	LDA	,X+
35	E1 FB	3		STA	\$FB
37	CD FF	4		LEAY	-1,Y
39	9D 00	3		CMPY	#0
3B	23 F7	4		BPL	LOOP 34-3D=F7
3D	F1 0C	3		LDA	12
3F	21 0B	4		BRA	NEXT 4C-41=0B
41	-	-	TAB	RMB	11
4C	00	2	NEXT	NOP	

(3p)

c) 
$$\underline{T} = 2 + 2 + (4 + 3 + 4 + 3 + 4)11 + 3 + 4 + 2 = 13 + 18 \cdot 11 = 13 + 198 = \underline{211 \mu s}$$

(3p)

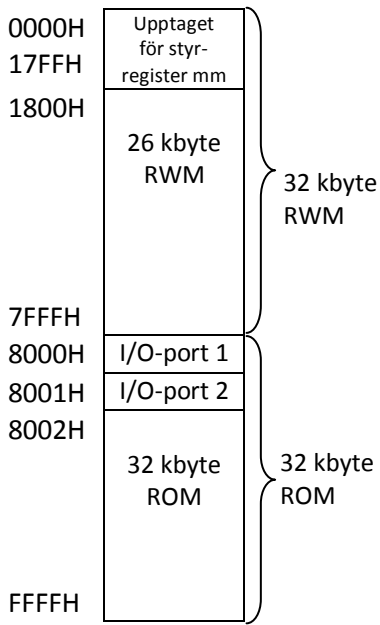
7.

; Subrutin COUNT

COUNT	PSHY		Spara pekare på stack
	CLR	TOTCNT	Nollställ totalräknare
	CLR	DIGCNT	Nollställ siffreräknare
CLOOP	LDA	,Y+	Hämta data från textsträng
	BEQ	CEXIT	Strängslut?
	INC	TOTCNT	Räkna ASCII
	ANDA	#\$7F	Ej strängslutt, maska bort bit 7
	CMPA	#'0'	Testa om ASCII-tecken för siffran "0"
	BLO	CLOOP	Ej siffra, fortsätt med nästa
	CMPA	#'9'	Testa om "9"
	BHI	CLOOP	Ej siffra, fortsätt med nästa
	INC	DIGCNT	Ja, öka siffreräknare
	BRA	CLOOP	Fortsätt med nästa
CEXIT	LDX	DIGCNT	Hämta resultat för retur
	LDA	TOTCNT	
	PULY		Återställ pekare
	RTS		
TOTCNT	RMB	1	Plats för räknare av alla ASCII-tecken
DIGCNT	RMB	1	Plats för räknare av sifvertecken

(6p)

8. Det som gör denna uppgift lite "knepig" är att storleken på den passiva arean inte är en 2-potens. Vi delar därför upp den i 3 st 2k-moduler och ser vad det leder till.



Passiv	Hex	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
3·2k = 3·2 <sup>11</sup>	Start: 0000	=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Slut: 07FF	=	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
	Start: 0800	=	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	Slut: 0FFF	=	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	Start: 1000	=	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	Slut: 17FF	=	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

CS

RWM	Start: 0000	=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32k = 2 <sup>15</sup>	Slut: 7FFF	=	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CS

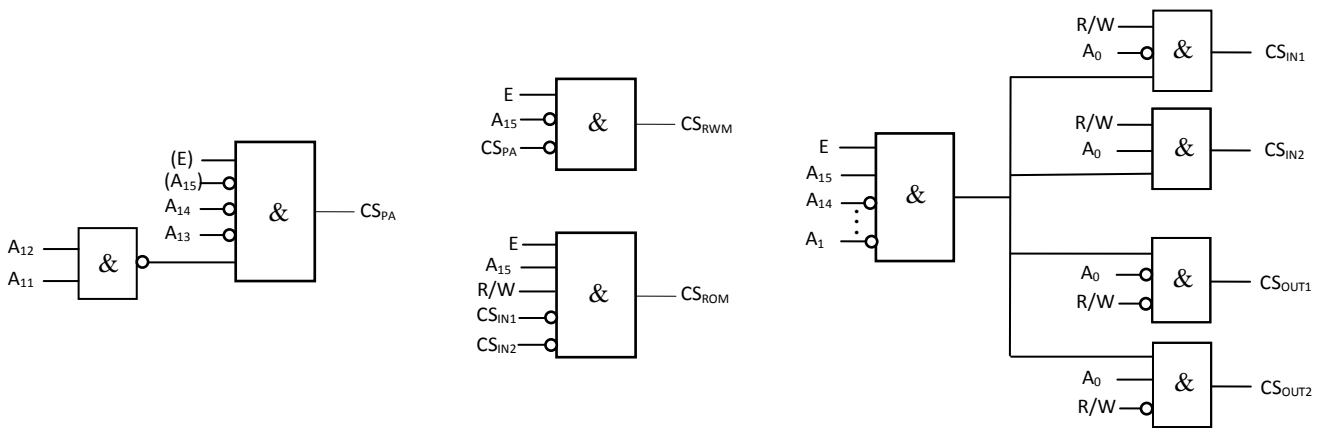
ROM	Start: 8000	=	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32k = 2 <sup>15</sup>	Slut: FFFF	=	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CS

I/O-port 1	8000	=	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I/O-port 2	8001	=	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

$$CS_{PA} = A_{15}'A_{14}'A_{13}'A_{12}'A_{11}' + A_{15}'A_{14}'A_{13}'A_{12}'A_{11} + A_{15}'A_{14}'A_{13}'A_{12}A_{11}' =$$

$$= A_{15}'A_{14}'A_{13}'(A_{12}'A_{11}' + A_{12}'A_{11} + A_{12}A_{11}') = A_{15}'A_{14}'A_{13}'(A_{12}' + A_{11}') = A_{15}'A_{14}'A_{13}'(A_{12}A_{11})'$$



(7p)