

Lösningförslag tenta 2015-01-12

(Version 1 med reservation för eventuella fel. **Rapportera gärna fel, så uppdateras lösningen efterhand!** Uppdaterad 150119.)

1. $X = 01\ 0111_2$; $Y = 10\ 1000_2$ (6 bitars ordlängd)

a) $[0, 2^n - 1] = [0, 2^6 - 1] = [0, 63]$ (1p)

b) $[-2^{n-1}, +2^{n-1} - 1] = [-2^{6-1}, +2^{6-1} - 1] = [-32, +31]$ (1p)

c) $S = X + Y$

6543210	bitnummer
0000000	carry
010111	X
+101000	Y
111111	= S

(1p)

d) $N = s_5 = \underline{1}$
 $Z = \underline{0}$ ($S \neq 0$)
 $V = x_5 * y_5 * s_5' + x_5' * y_5' * s_5 = 0 * 1 * 1' + 0' * 1' * 1 = \underline{0}$
 $C = c_6 = \underline{0}$

NZVC = 1000 (1p)

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

6543210	bitnummer
0101111	carry
010111	X
+010111	Y _{1k}
101111	= D

(1p)

f) $N = d_5 = \underline{1}$
 $Z = \underline{0}$ ($D \neq 0$)
 $V = x_5 * y_{5k} * d_5' + x_5' * y_{5k}' * d_5 = 0 * 0 * 1' + 0' * 0' * 1 = \underline{1}$
 $C = c_6' = 0' = \underline{1}$

NZVC = 1011 (1p)

g) $\underline{X} = 01\ 0111_2 = 17_{16} = 1 \cdot 16 + 7 = \underline{23}_{10}$
 $\underline{Y} = 10\ 1000_2 = 28_{16} = 2 \cdot 16 + 8 = \underline{40}_{10}$
 $\underline{S} = 11\ 1111_2 = 3F_{16} = 3 \cdot 16 + 15 = \underline{63}_{10}$ Resultatet S är korrekt eftersom $C = 0$.
 $\underline{D} = 10\ 1111_2 = 2F_{16} = 2 \cdot 16 + 15 = \underline{47}_{10}$ Resultatet D är felaktigt eftersom $C = 1$. (1p)

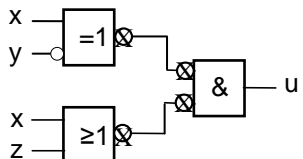
h) ($x_5 = 0$, pos) $\underline{X} = \underline{23}_{10}$.
 ($y_5 = 1$, neg) $\underline{Y}_{2k} = 2^6 - 40_{10} = 64_{10} - 40_{10} = 24_{10}$ \underline{Y} motsvarar $\underline{-24}_{10}$.
 ($s_5 = 1$, neg) $\underline{S}_{2k} = 2^6 - 63_{10} = 64_{10} - 63_{10} = 1$ \underline{S} motsvarar $\underline{-1}$. Resultatet S är ok eftersom $V = 0$.
 ($d_5 = 1$, neg) $\underline{D}_{2k} = 2^6 - 47_{10} = 64_{10} - 47_{10} = 17_{10}$ \underline{D} motsvarar $\underline{-17}_{10}$. D är felaktigt eftersom $V = 1$. (1p)

i) $C31BC000_{16} = 1/100\ 0011\ 0/001\ 1011\ 1100\ 0000\ 0000\ 0000_2$

s	c	f
---	---	---

$\underline{exp} = c - 127_{10} = 1000\ 0110_2 - 127_{10} = 128 + 6 - 127 = \underline{7}$

$\underline{N}_{flvt} = -1.001\ 1011\ 1100\ 0000\ 0000\ 0000_2 * 2^7 = -1001\ 1011.1100\ 0000\ 0000\ 0000_2 =$
 $= - (128 + 16 + 8 + 2 + 1 + 0,5 + 0,25) = \underline{-155,75}$ (2p)

2. a) 
 $u = (x'y' + xy)(x + z) = x'y'z + xy + xyz = x'y'z + xy$
 $\underline{u} = (x + z)(x + y')(x' + y)$ **(Nr 5)**

		yz			
		00	01	11	10
x	0	0	1	0	0
	1	0	0	1	1

(3p)

b)

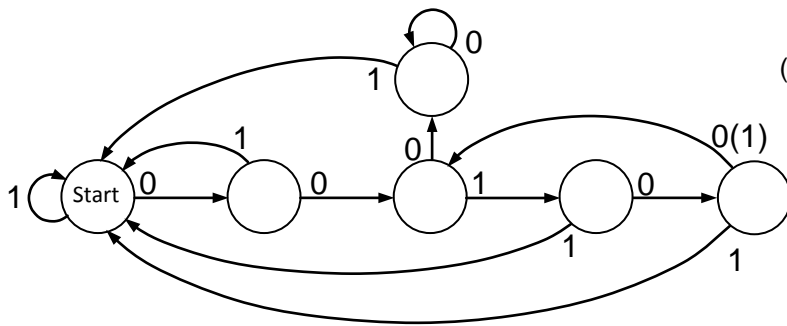
		cd			
		00	01	11	10
ab	00	1	0	0	1
	01	1	0	1	0
	11	-	-	-	-
	10	1	-	1	0

 NAND-gindar \leftrightarrow "ettor"
 $\underline{f} = c'd' + (a \oplus b \oplus d')c$ **(Nr 6)**

(4p)

3.

a)



(Ej utsatta utsignaler = 0)

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

(4p)

b)

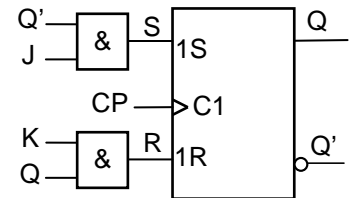
J	K	Q	Q ⁺	S	R
0	0	0	0	0	-
0	0	1	1	-	0
0	1	0	0	0	-
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	-	0
1	1	0	1	1	0
1	1	1	0	0	1

		S	Q
		0	1
JK	00	0	-
	01	0	0
	11	1	0
	10	1	-

		R	Q
		0	1
JK	00	-	0
	01	-	1
	11	0	1
	10	0	0

$S = JQ'$

$R = KQ$



(4p)

4.

CP	Styrsignaler (=1)	RTN	A(8)	B(8)	T(8)	R(8)
1	OE _A , LD _T	A → T	CD	32	?	?
2	OE _B , f ₃ , f ₂ , LD _R	B - T - 1 → R	CD	32	CD	?
3	OE _R , f ₃ , f ₁ , f ₀ , LD _R , LD _T	2R → R, R → T	CD	32	CD	64
4	OE _R , f ₃ , f ₁ , LD _R	R + T → R	CD	32	64	C8
5	OE _R , f ₃ , f ₁ , f ₀ , LD _R	2R → R	CD	32	64	2C
6	OE _B , LD _T	B → T	CD	32	64	58
7	OE _R , f ₃ , f ₁ , LD _R	R + T → R	CD	32	32	58
8	OE _R , LD _A	R → A	CD	32	32	8A
9	?	?	8A	32	32	8A

(4p)

5. a)

State	Summaterm	RTN-beskrivning	Styrsignaler
Q ₄	Q ₄ ·I _{xx}	M(PC) → TA, PC+1 → PC	MR, LD _{TA} , INC _{PC}
Q ₅	Q ₅ ·I _{xx}	M(TA) → T	MR, g ₁₄ , LD _T
Q ₆	Q ₆ ·I _{xx}	A + T + C → R, ALU(NZVC) → CC	OE _A , f ₃ , f ₁ , f ₀ , g ₁ , LD _R , LD _{CC}
Q ₇	Q ₇ ·I _{xx}	R → A, (New Fetch)	OE _R , LD _A , NF

Instruktionen är: ADCA Adr

(2p)

b)

State	S-term	RTN-beskrivning	Styrsignaler
Q ₄	Q ₄ ·I ₀₄	M(PC) → TA, PC+1 → PC	MR, LD _{TA} , INC _{PC}
Q ₅	Q ₅ ·I ₀₄	M(TA) → T	MR, g ₁₄ , LD _T
Q ₆	Q ₆ ·I ₀₄	M(PC) _{1k} → R, PC+1 → PC	MR, f ₂ , f ₀ , LD _R , INC _{PC}
Q ₇	Q ₇ ·I ₀₄	R AND T → R, ALU(NZ) → CC, 0 → V, C → CC	OE _R , f ₂ , f ₁ , f ₀ , LD _R , g ₅ , g ₃ , g ₂ , LD _{CC}
Q ₈	Q ₈ ·I ₀₄	R → M(TA), (New Fetch)	OE _R , g ₁₄ , MW, NF

OPKOD
Adr
mask

(5p)

6.

a) OP-koden har 8 bitar vilket ger $2^8=256$ olika kombinationer och 256 möjliga instruktioner. (1p)

b) Det villkorliga hoppet BMI Adr utförs om N-flaggan = 1. Det innebär att det är enklast att betrakta talen i additionen som tal med tecken. 64_{16} motsvarar då 100_{10} och talområdet är $[-128, 127]$.

Om overflow inte inträffar utförs hoppet om $100 + W < 0$, dvs. $W < -100$.

När hänsyn tas till talområdet blir intervallet: $-128 \leq W < -100$.

Detta motsvarar maskintalen: $256 - 128 \leq W < 256 - 100$, dvs. $128 \leq W < 156$.

Overflow inträffar vid additionen om: $100 + W \geq 128$, dvs. $W \geq 28$, som ger intervallet $28 \leq W \leq 127$ med hänsyn till talområdet. Även i detta fall får N-flaggan värdet 1 (= fel värde) och hoppet utförs.

De två intervallen kan slås samman och hoppet utförs alltså om $28 \leq W < 156$. (4p)

c)

Adr	Data	~				
40	90 53	2		LDX	#DATA	
42	F3 03	3		LDA	3,X	
44	E1 57	3		STA	COUNT	
46	F3 02	3	WAIT1	LDA	2,X	
48	08	3	WAIT2	DECA		
49	00	2		NOP		
4A	25 FC	4		BNE	WAIT2	48-4C = FC
4C	00	2		NOP		
4D	38 57	4		DEC	COUNT	
4F	25 F5	4		BNE	WAIT1	46-51 = F5
51	21 05	4		BRA	NEXT	58-53 = 05
53	00		DATA	FCB	0,1,2,4	
54	01					
55	02					
56	04					
57	--		COUNT	RMB	1	
58	?			NEXT		

Instruktionen består av 2 bytes.

Instruktionen består av 2 bytes.

Felet i instruktionslistan som markeras med rött ovan tas hänsyn till vid rättningen! (3p)

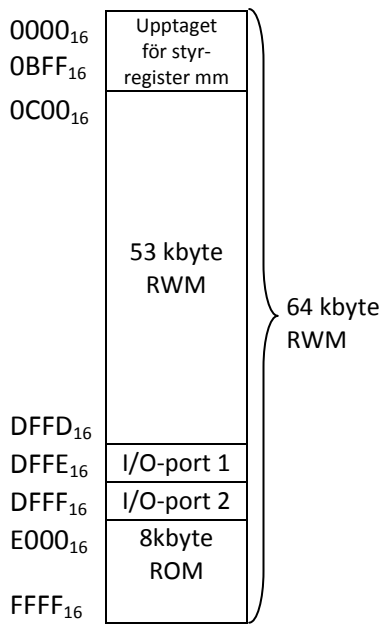
d) $T = 2 + 3 + 3 + [3 + (3 + 2 + 4)*2 + 2 + 4 + 4]*4 + 4 = 12 + [13 + 9*2]*4 = 12 + 31*4 = 136 \mu s$ (3p)

7.

PCNT	PSHY		Spara register på stack
	CLR	HITCNT	Nollställ träffräknare
PLOOP	LDA	,Y+	Hämta data från tabell. Öka pekare
	CMPA	#\$FF	Tabellslut?
	BEQ	PEX	Ja, avsluta
	ANDA	11010101	Maska don't care-bitar
	CMPA	10010100	Testa bitmönster
	BNE	PLOOP	Ej träff, testa nästa
COUNT	INC	HITCNT	Träff, öka räknare
	BRA	PLOOP	Testa nästa
PEX	LDA	HITCNT	Returnera antal träffar
	PULY		Återställ register
	RTS		
HITCNT	RMB	1	Räknare för träff

(6p)

8. I denna uppgift är storleken på den passiva arean inte en 2-potens. Vi delar därför upp den i 3 st 1k-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	
3·1k = 3·2 ¹⁰	Start: 0000	=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Slut: 03FF	=	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	
	Start: 0400	=	0	0	0	0	0	1	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	
	Start: 0800	=	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
	Slut: 0BFF	=	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	

CS

I/O-port 1	DFFE	=	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0
I/O-port 2	DFFF	=	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1

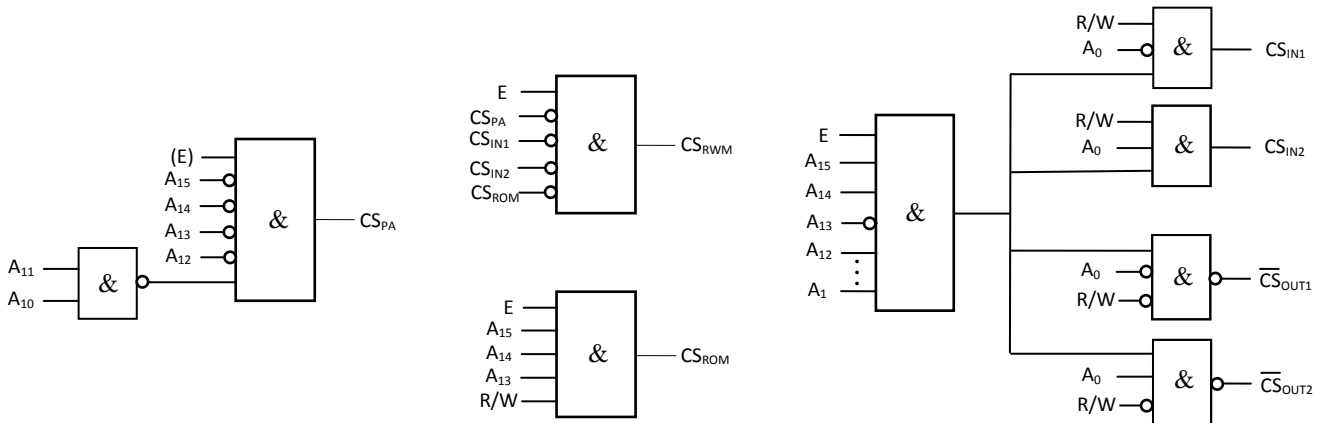
CS

ROM	Start: E000	=	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
8k = 2 ¹³	Slut: FFFF	=	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CS

$$CS_{PA} = A_{15}'A_{14}'A_{13}'A_{12}'A_{11}'A_{10}' + A_{15}'A_{14}'A_{13}'A_{12}'A_{11}A_{10} + A_{15}'A_{14}'A_{13}'A_{12}'A_{11}A_{10}' =$$

$$= A_{15}'A_{14}'A_{13}'A_{12}'(A_{11}'A_{10}' + A_{11}'A_{10} + A_{11}A_{10}') = A_{15}'A_{14}'A_{13}'A_{12}'(A_{11}A_{10})'$$



(Inverteringarna på utgångarna CS_{out} kan utelämnas.)

(7p)