

Lösningförslag tenta 2012-10-22

(Version 2 med reservation för eventuella fel.)

1. $X = 1\ 1011\ 0011_2$; $Y = 0\ 1010\ 0111_2$ (9 bitars ordlängd)

a) $[0, 2^n - 1] = [0, 2^9 - 1] = [0, 511]$ (1p)

b) $[-2^{n-1}, +2^{n-1} - 1] = [-2^{9-1}, +2^{9-1} - 1] = [-256, +255]$ (1p)

c) $S = X + Y$

9876543210	bitnummer
1101001110	carry
110110011	X
+010100111	Y
001011010	= S

(1p)

d) $\underline{N} = s_8 = 0$
 $\underline{Z} = 0$ ($S \neq 0$)
 $\underline{V} = x_8 * y_8 * s_8' + x_8' * y_8' * s_8 = 1 * 0 * 0' + 1' * 0' * 0 = 0$
 $\underline{C} = c_9 = 1$

(1p)

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

9876543210	bitnummer
1111100111	carry
110110011	X
+101011000	Y _{1k}
100001100	= D

(1p)

f) $\underline{N} = d_8 = 1$
 $\underline{Z} = 0$ ($D \neq 0$)
 $\underline{V} = x_8 * y_{8ik} * d_8' + x_8' * y_{8ik}' * d_8 = 1 * 1 * 1' + 1' * 1' * 1 = 0$
 $\underline{C} = c_9' = 1' = 0$

(1p)

g) $\underline{X} = 1\ 1011\ 0011_2 = 1B_{16} = 256 + 11 * 16 + 3 = 435$
 $\underline{Y} = 0\ 1010\ 0111_2 = A7_{16} = 10 * 16 + 7 = 167$
 $\underline{S} = 0\ 0101\ 1010_2 = 5A_{16} = 5 * 16 + 10 = 90$ Resultatet S är felaktigt eftersom $C = 1$.
 $\underline{D} = 1\ 0000\ 1100_2 = 10C_{16} = 256 + 12 = 268$ Resultatet D är korrekt eftersom $C = 0$. (1p)

h) ($x_8 = 1$, neg) $X_{2k} = 2^9 - 435 = 512 - 435 = 77$ \underline{X} motsvarar -77
 ($y_8 = 0$, pos) $\underline{Y} = 010100111_2 = 167$
 ($s_8 = 0$, pos) $\underline{S} = 001011010_2 = 90$ Resultatet S är korrekt eftersom $V = 0$.
 ($d_8 = 1$, neg) $D_{2k} = 2^9 - 268 = 512 - 268 = 244$ \underline{D} motsvarar -244 . Korrekt eftersom $V = 0$. (1p)

i) 6 decimala sifferpositioner krävs inklusive teckensiffra: $A = 047320$; $B = 058196$; $B_{9k} = 941803$
 Princip: $D = A + B_{9k} + 1$

543210	siffernnummer
001001	1 carry
047320	A
+941803	B _{9k}
989124	= D (Resultatet är negativt då $d_5 = 9$.)

För att kontrollera resultatet kan man 10-komplementera det för att se beloppet.
 $D_{10k} = D_{9k} + 1 = 010875 + 1 = 010876$; D motsvarar alltså -10876 (3p)

j) $N_{flyt} = C2CAC000_{16} = 1/100\ 0010\ 1/100\ 1010\ 1100\ 0000\ 0000\ 0000_2$
 s c f

$s = 1$ (-)
 $c = 128 + 5 = 127 + 6$; $exp = 127 + 6 - 127 = 6$
 $m = 1.f = 1.100\ 1010\ 1100\ 0000\ 0000\ 0000$

$\underline{N}_2 = -1.100\ 1010\ 1100\ 0000\ 0000\ 0000 * 2^6 = -(64 + 32 + 4 + 1 + ,25 + ,125) = -101,375$ (2p)

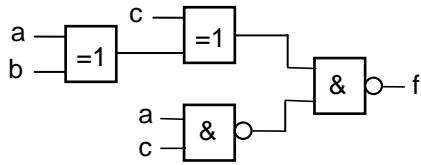
k)

$$\begin{aligned} \underline{u} &= (a'c)'ab + a'c(ab)' = \\ &= (a+c')ab + a'c(a'+b)' = \\ &= ab + abe' + a'c + a'b'e = \\ &= ab + a'c = (a+c)(a'+b) \end{aligned}$$

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

(4p)

2. $f = ac + a'b'c' + a'bc + abc' + ab'c = ac + a'(b'c' + bc) + a(bc' + b'c) =$
 $= ac + a'(b \oplus c)' + a(b \oplus c) = ac + [a \oplus (b \oplus c)]' = ac + (a \oplus b \oplus c)'$



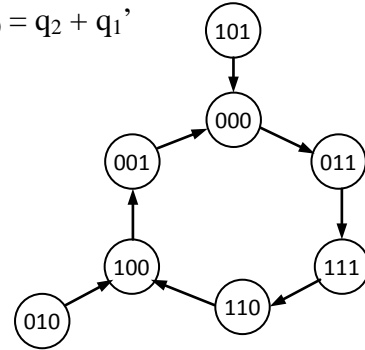
		cd			
		00	01	11	10
ab	00	1	1	0	0
	01	0	0	1	1
	11	1	1	1	1
	10	0	0	1	1

(4p)

3.

a) $J_2 = q_1, K_2 = q_1'; J_1 = (q_2 + q_0)', K_1 = q_0'; J_0 = q_1', K_0 = q_2 + q_1'$

q ₂	q ₁	q ₀	J ₂	K ₂	J ₁	K ₁	J ₀	K ₀	q ₂ ⁺	q ₁ ⁺	q ₀ ⁺
0	0	0	0	1	1	1	1	1	0	1	1
0	0	1	0	1	0	0	1	1	0	0	0
0	1	0	1	0	1	1	0	0	1	0	0
0	1	1	1	0	0	0	0	0	1	1	1
1	0	0	0	1	0	1	1	1	0	0	1
1	0	1	0	1	0	0	1	1	0	0	0
1	1	0	1	0	0	1	0	1	1	0	0
1	1	1	1	0	0	0	0	1	1	1	0



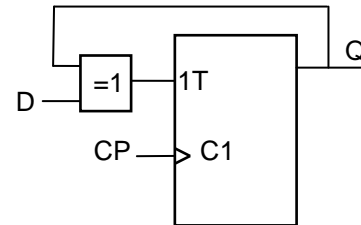
(5p)

b)

För T-vippor gäller att $q^+ = q$ för $T = 0$ och $q^+ = q'$ för $T = 1$.

D	Q	Q ⁺	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

$$T = D'Q + DQ' = D \oplus Q$$



(3p)

4. $10A - 8(B + 1) = 2A + 8(A - B - 1)$

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

(4p)

5. a)

State nr	S-term	RTN-beskrivning	Styr signaler (=1)
Q ₅	Q ₅ ·I _{xx}	PC → MA, PC+1 → PC, SP - 1 → SP	OE _{PC} , LD _{MA} , IncPC, DecSP
Q ₆	Q ₆ ·I _{xx}	M → T	MR, LD _T
Q ₇	Q ₇ ·I _{xx}	SP → MA	OE _{SP} , LD _{MA}
Q ₈	Q ₈ ·I _{xx}	PC → M, PC + T → R	OE _{PC} , MW, f ₃ , f ₁ , LD _R
Q ₉	Q ₉ ·I _{xx}	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, som också ökas med ett. SP minskas med 1.

Q₆: Dataordet efter OP-koden laddas i T-reg.

Q₇: Stackpekaren laddas i MA.

Q₈: PC skrivs på stack. PC och dataordet efter OP-koden adderas. Summan till R.

Q₉: Summan till PC. (Hopp)

Detta är BSR Adr (Branch to subroutine.)

(3p)

b)

State nr	S-term	RTN-beskrivning	Styr signaler (=1)
Q ₅	Q ₅ ·I _{FB}	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC
Q ₆	Q ₆ ·I _{FB}	M → R	MR, LD _R
Q ₇	Q ₇ ·I _{FB}	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC
Q ₈	Q ₈ ·I _{FB}	M → MA	MR, LD _{MA}
Q ₉	Q ₉ ·I _{FB}	R → M, (Next Fetch)	OE _R , MW, NF

OPKOD
Data
Adr

(4p)

6.

- a) Assemblatorn arbetar rad för rad, uppifrån och ner. Den måste veta startadressen för att kunna bestämma värdet hos de symboler som är lägesnamn. Symbolen START måste därför definieras i början av programmet. Symbolen INPORT kan definieras var som helst i programmet. (2p)
- b) Programräknaren PC innehåller adressen till nästa instruktion eller del av instruktion. PC håller alltså reda på var programmet finns i minnet. (2p)
- c) SP-registret innehåller alltid adressen till det översta värdet på stacken. När ett värde skrivs på stacken minskas först SP och vid läsning ökas SP efter läsningen. Skrivning på stacken görs vid subrutinanrop och vid lagring av register på stacken (push). Läsning görs vid återhopp från subrutin och vid hämtning av registerinnehåll från stack (pull). (2p)
- d) BLE (\leq) avser tal med tecken. För 8-bitars tal gäller då talintervallet $[-128, 127]$. $\$60 = 96$ NEGA utför $-W$. Hoppvillkoret blir $-W - 96 \leq 0$ eller $-96 \leq W$, dvs $-96 \leq W \leq 127$.

Eftersom negativa värden ersätts med 2-komplementet av motsvarande positiva värde delar vi upp intervallet i en positiv och en negativ del:

$$0 \leq W \leq 127 \text{ och } -96 \leq W \leq -1.$$

Verkligt värde för den negativa delen blir då: $256 - 96 \leq W \leq 256 - 1$ eller $160 \leq W \leq 255$

(3p)

6. (forts.)

e)

Adr	Data (Hex)	~	Läge		
-			START	EQU	\$10
-			INPORT	EQU	\$FD
-				ORG	START
10	11 21	4		LDX	#TAB
12	0B FD	5	INLOOP	LDAA	INPORT
14	13 20	5		STAA	ACNT
16	84	6	ALOOP	LDAB	A,X
17	42	4	BLOOP	INCB	
18	5B FD	5		BMI	BLOOP
					17 - 1A = FD
1A	46 20	6		DEC	ACNT
1C	5E F8	5		BNE	ALOOP
					16 - 1E = F8
1E	5A F2	5		BRA	INLOOP
					12 - 20 = F2
20	-		ACNT	RMB	1
21	00,08,FE,F6,FB,0A,FF		TAB	FCB	0,8,-2,-10,-5,10,\$FF

(3p)

f) $T = 5+5+(6+(4+5)*10+6+5)*3+5 = 15+(17+90)*3 = 15 + 321 = \underline{336 \text{ klockpulser } (\mu\text{s})}$

(3p)

7.

START	EQU	\$10	Programstart
BOS	EQU	\$FB	Bottom of stack
VARIABL	EQU	\$FC	Variabel för upp- och nedräkning
DIPSW	EQU	\$FD	Inport för DIPSWITH
HEXDISP	EQU	\$FE	Utport för HEXDISPLAY
	ORG	START	
	LDS	#BOS	Init stack
	LDAA	#127	Begynnelsevärde för VARIABEL
	STAA	VARIABL	
LOOP	LDAA	DIPSW	Läs switchar
	ANDA	##10000001	Maska fram bit 0 och 7
	CMPA	##00000001	Mönster för ökning
	BNE	NOINC	Ingen ökning, kolla minskning
	LDAA	VARIABL	Kolla maxvärde för variabel
	CMPA	#255	Maxvärde?
	BEQ	SHOW	Ja, ingen ökning. Visa värde
	INC	VARIABL	Ej maxvärde, öka variabel
	BRA	SHOW	Visa värde
NOINC	CMPA	##10000000	Minskning?
	BNE	SHOW	Nej, visa värde
	TST	VARIABL	Kolla minvärde (0) för variabel
	BEQ	SHOW	Visa värdet om det var 0
	DEC	VARIABL	Ej minvärde, minska variabel
SHOW	LDAA	VARIABL	Visa variabelvärde på HEXDISPLAY
	STAA	HEXDISP	
	JSR	DELAY	Vänta
	BRA	LOOP	Ett varv till!

(5p)