

Lösningsförslag tenta 2009-08-20

1. $X = 1100111$; $Y = 0010001$ (7-bitars ordlängd)

a) $[-2^{7-1}, +2^{7-1}-1] = [-64, +63]$ (1p)

b) $[0, 2^7-1] = [0, 127]$ (1p)

c) $R = X + Y_{1k} + 1$

76543210	bitnummer
11011111	1 carry
1100111	X
+1101110	Y _{1k}
1010110	R

(1p)

d) $N = r_6 = 1$;
 $Z = 0$ ($R \neq 0$);
 $V = x_6 * y_6 * r_6' = 1 * 1 * 1' = 1 * 1 * 0 = 0$; (Vid "subtraktion" är y_6 motsvarande bit i Y_{1k} .)
 $C = c_7' = 1' = 0$ (1p)

e) $\underline{R} = 1010110_2 = 56_{16} = 5 * 16 + 6 = \underline{86}$;
 $\underline{X} = 1100111_2 = 67_{16} = 6 * 16 + 7 = \underline{103}$;
 $\underline{Y} = 0010001_2 = 16_{16} = 16 + 1 = \underline{17}$; Korrekt resultat om $\underline{C} = 0$. (1p)

f) $\underline{R} = 86 = 2^7 - 42 = 128 - 42$ vilket motsvarar $\underline{-42}$ ($r_6 = 1$, neg);
 $\underline{X} = 103 = 2^7 - 25 = 128 - 25$ vilket motsvarar $\underline{-25}$ ($x_6 = 1$, neg);
 $\underline{Y} = 17$ ($y_5 = 0$, pos); Korrekt resultat om $\underline{V} = 0$. (1p)

g) $\underline{-57}$ motsvarar $128 - 57 = 71 = \underline{1000111}_2$ (1p)

h) $-27,25 = -11011.01_2 = -1.101101 * 2^4$;
s/c/f; s = 1; c = 4 + 127 = 131; f = 10110100...0;
Flyttalet blir: $1/10000011/1011010000000000000000_2 = C1DA0000_{16}$ (3p)

i)

$$\underline{u} = (a'b + ab') \oplus (a'b') = (a'b' + ab)a'b' + (a'b + ab')(a'b')' = a'b' + (a'b + ab')(a + b) = a'b' + a'b + ab' = (ab)' = a' + b'$$

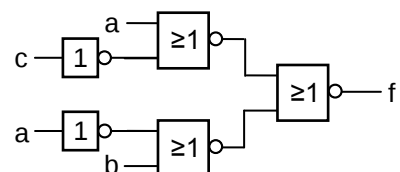
(2p)

j) $u = abcde = ((ab)''(cd)''e)''$

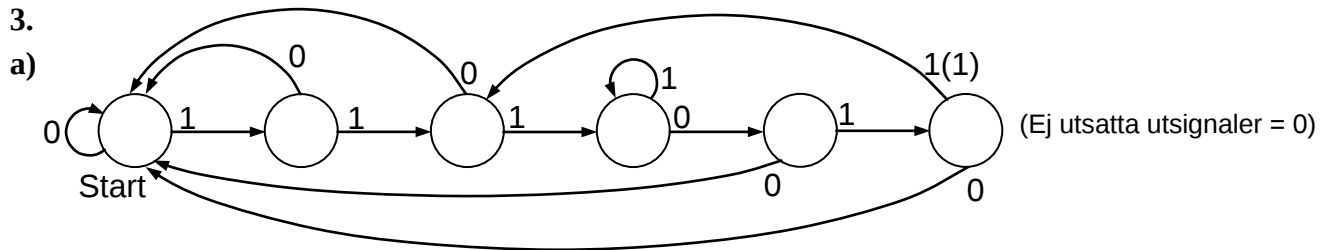
2. $f(a,b,c) = (a' + b + c)(b + c')(a + c')$

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

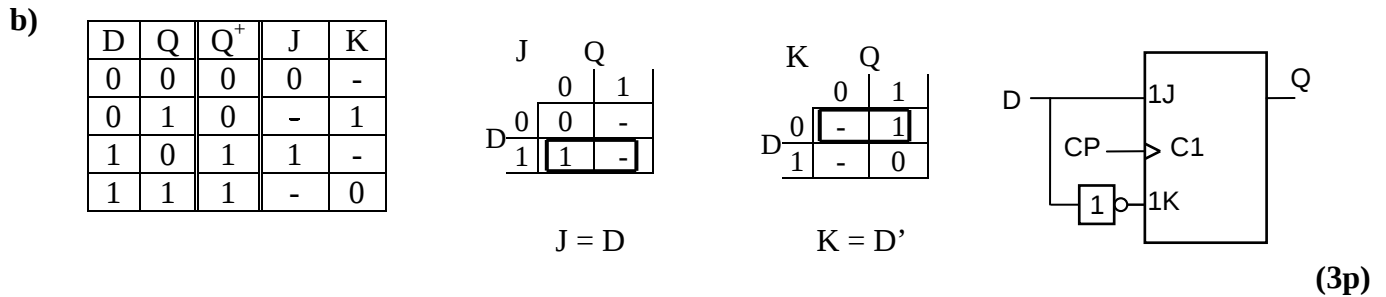
$f(a,b,c) = (a + c')(a' + b)$



(4p)



(4p)



4. $5A - 3(B + 1) = 2A + 3(A - B - 1)$

CP	RTN	Styrsignaler (=1)
1	$B \rightarrow T$	OE_B, LD_T
2	$A - B - 1 \rightarrow R$	OE_A, f_3, f_2, LD_R
3	$2R \rightarrow R, R \rightarrow T$	$OE_R, f_3, f_1, f_0, LD_R, LD_T$
4	$R + T \rightarrow R$	OE_R, f_3, f_1, LD_R
5	$A \rightarrow T$	OE_A, LD_T
6	$R + T \rightarrow R$	OE_R, f_3, f_1, LD_R
7	$R + T \rightarrow R$	OE_R, f_3, f_1, LD_R
8	$R \rightarrow B$	OE_R, LD_B

(5p)

5. a)

State	S-term	RTN-beskrivning	Styrsignaler (=1)
Q ₅	Q ₅ ·I _{xx}	X→T, SP-1→SP	OE _X , LD _T , DecSP
Q ₆	Q ₆ ·I _{xx}	SP→MA	OE _{SP} , LD _{MA}
Q ₇	Q ₇ ·I _{xx}	PC→M	OE _{PC} , MW
Q ₈	Q ₈ ·I _{xx}	B+T→R	OE _B , f ₃ , f ₁ , LD _R
Q ₉	Q ₉ ·I _{xx}	R→PC, NF	OE _R , LD _{PC} , NF

(1p)

- b)
- Q₅: Värdet i X-reg kopieras till T-reg. Innehållet i stackpekaren SP minskas med 1.
 - Q₆: Innehållet i stackpekaren SP kopieras till MA-registret.
 - Q₇: Innehållet i programräknaren PC skrivs på stacken (en återhoppadress).
 - Q₈: Innehållen i B- och X-registren adderas och summan placeras i R-registret.
 - Q₉: Innehållet i R-registret (summan ovan) kopieras till PC-registret, dvs. ett hopp utförs.

Detta är JSR B,X (2p)

c)

State	S-term	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	Q ₅ ·I _{xx}	PC → MA, T, PC + 1 → PC	OE _{PC} , LD _{MA} , LD _T , IncPC
Q ₆	Q ₆ ·I _{xx}	M + T + 1 → R	MR, f ₃ , f ₁ , g ₀ , LD _R
Q ₇	Q ₇ ·I _{xx} [(N⊕V)+Z] Q ₇ ·I _{xx}	If [(N⊕V)+Z] = 1: R → PC, Next Fetch else: Next Fetch	OE _R , LD _{PC} NF (NF utförs alltid här!)

(5p)

6.

- a) JMP Adr utför hopp till adressen som finns med som andra ord i instruktionen . Det innebär att denna adress är fast och därmed låser programmet till en bestämd startadress om hoppet görs till en adress inom programmet. JMP är snabbare att utföra än BRA och Adr är enkel att bestämma vid handassemblering. BRA Adr utför också hopp till adressen Adr, men i detta fall innehåller instruktionens andra ord avståndet till destinationsadressen. Om programmet flyttas till en annan startadress fungerar hoppet (inom programmet) fortfarande eftersom hoppavståndet inte ändras. Hoppavståndet måste alltså räknas ut vid översättningen till maskinkod och detta komplicerar ju handassemblering något. (2p)
- b) Minnesorienterad I/O används av FLEX-processorn. Man behöver då inte ha speciella I/O-instruktioner utan kan använda vanliga instruktioner för läsning och skrivning i minnet. (2p)
- c) Tillstånden bestäms av en 4-bitars räknare med $2^4 = 16$ olika tillstånd. RESET och FETCH kräver $3 + 2 = 5$ tillstånd. $16 - 5 = 11$ tillstånd är därför tillgängliga för EXECUTE. Svar: 11 tillstånd. (2p)

a)

Adr	Data	~	Läge	Assemblerspråk
30	10 0A	4		LDAB #10
32	11 3E	4		LDX #TAB
34	7B	5	LOOP	LDAA 1,X+
35	13 FE	5		STAA \$FE
37	45	4		DECB
38	5C FA	5		BPL LOOP
3A	0B 0C	5		LDAA 12
3C	5A 0B	5		BRA NEXT
3E	-----	-	TAB	RMB 11
49	00	3	NEXT	NOP

$$34 - 3A = FA$$

$$49 - 3E = 0B$$

e)

$$t = [4 + 4 + 11(5 + 5 + 4 + 5) + 5 + 5 + 3] \mu s = [21 + 11 \cdot 19] \mu s = [21 + 209] \mu s = 230 \mu s$$

(3p)

(3p)

7.

CCOUNT	PSHA PSHX	Spara register på stack
	CLRB	Nollställ "C"-räknare
CCLOOP	LDAA 1,X+ TSTA BEQ CCEXIT	Hämta data från textsträng Strängslut? Ja, återhopp
	CMPA #'C' BNE CCLOOP	Nej, testa om "C" Nej, fortsatt med nästa
	INCB BRA CCLOOP	Ja, öka "C"-räknare Fortsatt med nästa
CCEXIT	PULX PULA RTS	Återställ register

(8p)