

Lösningförslag tenta 2011-04-26 (Med reservation för eventuella fel)

1. $X = 10\ 1000\ 0001$; $Y = 01\ 1100\ 0011$ (10 bitars ordlängd)

a) $R = X+Y$

109876543210	bitnummer
11100000110	carry
1010000001	X
+ 0111000011	Y
0001000100	= R

(1p)

b) $\underline{N} = r_9 = \underline{0}$
 $\underline{Z} = \underline{0}$ ($R \neq 0$)
 $\underline{V} = x_9 * y_9 * r_9' + x_9' * y_9' * r_9 = 1 * 0 * 0' + 1' * 0' * 0 = \underline{0}$
 $\underline{C} = c_{10} = 1 = \underline{1}$

(1p)

c) $\underline{R} = 00\ 0100\ 0100_2 = 044_{16} = 4 * 16 + 4 = \underline{68}$
 $\underline{X} = 10\ 1000\ 0001_2 = 281_{16} = 2 * 256 + 8 * 16 + 1 = 512 + 128 + 1 = \underline{641}$
 $\underline{Y} = 01\ 1100\ 0011_2 = 1C3_{16} = 1 * 256 + 12 * 16 + 3 = 256 + 192 + 3 = \underline{451}$
 Resultatet R är felaktigt ($C = 1$). Korrekt resultat om $C = 0$.

(1p)

d) ($r_9 = 0$, pos) $\underline{R} = \underline{68}$
 ($x_9 = 1$, neg) $\underline{X}_{2k} = 2^{10} - 641 = 1024 - 641 = 383$ \underline{X} motsvarar $\underline{-383}$
 ($y_9 = 0$, pos) $\underline{Y} = \underline{451}$
 Resultatet R är korrekt ($V = 0$). Korrekt resultat om $V = 0$.

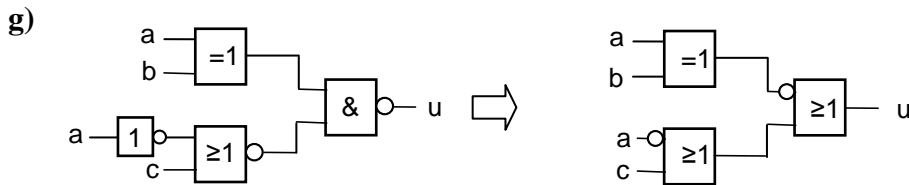
(1p)

e) $X + Y \geq 0$, för tal med tecken om $N \oplus V = 0$, dvs $\underline{f} = \underline{(N \oplus V)'}'$, efter additionen.

(3p)

f) Formatet är s/c/f, där s motsvarar tecknet (0 för +), c är karakteristikan ($c = \exp + 127$) och f är den normaliserade mantissan utan den inledande ettan. Det högsta c-värdet ($1111\ 1111_2 = 255$) är reserverat för att representera oändligheten. Det högsta användbara c-värdet är därför $1111\ 1110 = 254$. Bitmönstret för N_{\max} blir därför: $0/111\ 1111\ 0/111\ 1111\ 11.. \dots 11 = 7F7FFFFF_{16}$.

(2p)



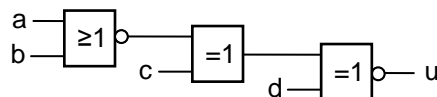
$$\underline{u} = (a \oplus b)' + a' + c = (a'b' + ab) + a'(b + b') + c = a'b' + ab + a'b + c = (ab')' + c = a' + b + c$$

(3p)

2.

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

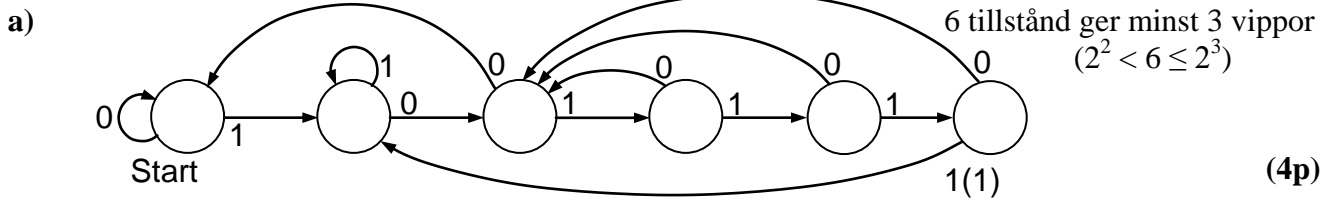
$$\begin{aligned} \underline{u} &= (a+b+c+d)(a+b+c'+d')(a'+c+d')(a'+c'+d)(b'+c+d')(b'+c'+d) = \\ &= [a+b+(c+d)(c'+d')][(a'+(c+d')(c'+d))(b'+(c+d')(c'+d))] = \\ &= [a+b+(c \oplus d)][(a'+(c \oplus d)')(b'+(c \oplus d)')] = \\ &= [a+b+(c \oplus d)][a'b'+(c \oplus d)'] = [(a+b)+(c \oplus d)][(a+b)'+(c \oplus d)'] = \\ &= (a+b) \oplus (c \oplus d) = [(a+b)' \oplus (c \oplus d)]' \end{aligned}$$



(Ring på utgången längst till höger i figuren ovan betyder att signalen inverteras efter XOR-grinden.)

(6p)

3. (Ej utsatta utsignaler = 0)



b)

x	q ₂	q ₁	q ₀	q ₂ ⁺	q ₁ ⁺	q ₀ ⁺	T ₂	T ₁	T ₀
0	0	0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	1	0	1
0	0	1	0	1	0	0	1	1	0
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	0	0	0	0
0	1	0	1	1	0	0	0	0	1
0	1	1	0	1	0	0	0	1	0
0	1	1	1	1	0	0	0	1	1
1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	1	0	1	1	0	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	1	0	0	0	1

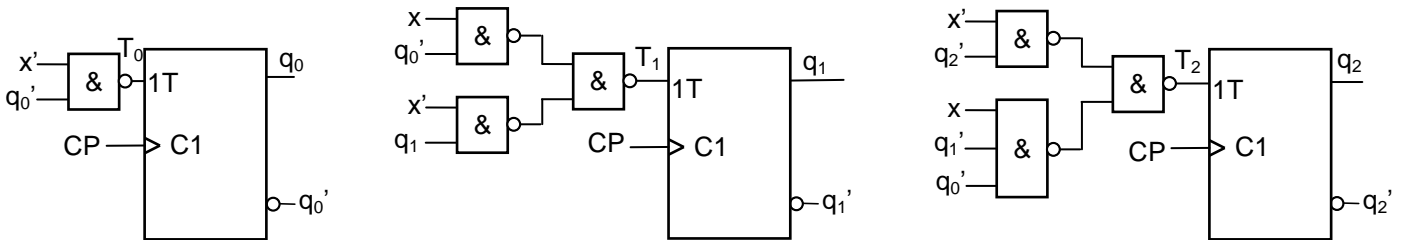
För T-vippor gäller att $q^+ = q$ för $T = 0$ och $q^+ = q'$ för $T = 1$.
För $x = 0$ gäller att $q_2q_1q_0 = 100$

T ₂	q ₁ q ₀	T ₁	q ₁ q ₀	T ₀	q ₁ q ₀
	00 01 11 10		00 01 11 10		00 01 11 10
00	1 1 1 1	00	0 0 1 1	00	0 1 1 0
01	0 0 0 0	01	0 0 1 1	01	0 1 1 0
11	1 0 0 0	11	1 0 0 1	11	1 1 1 1
10	1 0 0 0	10	1 0 0 1	10	1 1 1 1

$$T_2 = x'q_2' + xq_1'q_0'$$

$$T_1 = xq_0' + x'q_2$$

$$T_0 = x + q_0$$



(6p)

4. $6 \cdot A - 7 \cdot (B + 1) = 6(A - B - 1) - B - 1$

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, R → T	OE _R , f ₃ , f ₁ , f ₀ , LD _R , LD _T
4	R + T → R	OE _R , f ₃ , f ₁ , LD _R
5	2R → R	OE _R , f ₃ , f ₁ , f ₀ , LD _R
6	B → T	OE _B , LD _T
7	R - T - 1 → R	OE _R , f ₃ , f ₂ , LD _R
8	R → A	OE _R , LD _A

(5p)

5. a)

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

Instruktionen består av två ord. Andra ordet läses och placeras i T-registret. Innehålln i X- och T-registret adderas och summan placeras i R-registret. Resultatet placeras sedan i CC-registret. Detta är JMP n,X. (2p)

b)

State	S-term	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	Q ₅ ·I _{F9}	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC
Q ₆	Q ₆ ·I _{F9}	M → MA	MR, LD _{MA}
Q ₇	Q ₇ ·I _{F9}	M → R	MR, f ₀ , LD _R
Q ₈	Q ₈ ·I _{F9}	X → MA	OE _X , LD _{MA}
Q ₉	Q ₉ ·I _{F9}	R → M	OE _R , MW
Q ₁₀	Q ₁₀ ·I _{F9}	X+1 → R	OE _X , f ₃ , g ₀ , LD _R
Q ₁₁	Q ₁₀ ·I _{F9}	R → X, Next Fetch	OE _R , LD _X , NF

(4p)

- 6.
- a) Under processorns EXECUTE-fas utför den operationen som bestäms av OP-koden i instruktionsregistret IR. Om instruktionen består av flera ord så ökas PC-värdet med ett för varje ord efter OP-koden. EXECUTE-fasen avslutas med ett hopp till FETCH-fasen. (1p)
- b) Det som skiljer instruktionerna åt är hoppvillkoret, som är $N = 1$ för BMI och $N \oplus V = 1$ för BLT. Det innebär i båda fallen att hopp utförs om den aritmetiska operation som sätter flaggorna ger ett korrekt negativt resultat för tal med tecken. Om overflow inträffar vid operationen blir resultatet felaktigt och N-flaggan får fel värde, varför BMI-instruktionen då fungerar fel medan BLT-instruktionen fungerar korrekt eftersom uttrycket $N \oplus V$ alltid ger korrekt tecken. (2p)
- c) Funktionen hos signalen NF är att ladda tillståndsräknaren med värdet 3 som motsvarar första tillståndet i FETCH. Om sista tillståndet i EXECUTE inte aktiverar NF så fortsätter tillståndsräknaren räkna tills den varvar och börjar om med RESET följt av FETCH, dvs processorn startar om på det aktuella programmets startadress. (2p)
- d) BGE avser tal med tecken. För 8-bitars tal gäller då att talintervallet är $[-128, 127]$.
 $\$50 = 80_{10}; \quad 80 - W \geq 0; \quad 80 \geq W; \quad -128 \leq W \leq 80$ (tolkning)

Eftersom negativa tal ersätts av 2-komplementet av motsvarande positiva tal så delar vi upp talintervallet i en positiv (inklusive 0) och en negativ del:

$$\underline{0 \leq W \leq 80} \\ -128 \leq W \leq -1 \text{ (tolkning); } \quad 256 - 128 \leq W \leq 256 - 1 \text{ (i verkligheten); } \quad \underline{128 \leq W \leq 255} \quad (3p)$$

e)

Adr	Data	~	Läge		
40	0F F6	4		LDAA #10	
42	11 F0	4	LOOP1	LDX #16	
44	76 04	6	LOOP2	LEAX 4,X	
46	50 00	5		CPX #0	
48	5E FA	5		BNE LOOP2	44 - 4A = FA
4A	41	4		INCA	
4B	5B F5	5		BMI LOOP1	42 - 4D = F5
4D	11 60	4	GOON	LDX #\$60	

(3p)

f) $t = [4 + (4 + (6 + 5 + 5) * 4 + 4 + 5) * 10 + 4] \mu s = [8 + (13 + 16 * 4) * 10] \mu s = [8 + 770] \mu s = \underline{778 \mu s}$ (3p)

7. TCHK	PSHX		Spara register på stack
	CLRA		Nollställ negativräknare
	CLR	BITCNT	Nollställ räknare för bitvärden
TLOOP	LDAB	1,X+	Hämta data från tabell. Öka pekare
	TSTB		Tabellslut?
	BEQ	TEX	Ja, avsluta
	BPL	POSIT	Positivt tal
	INCA		Öka negativräknare
POSIT	ANDB	01001000	Maska bort alla bitar utom nr 6 och 3
	CMPB	01000000	Bit nr 6 = 1 och bit 3 = 0?
	BNE	TLOOP	Ej träff, testa nästa
	INC	BITCNT	Öka räknare för bitvärden
	BRA	TLOOP	Testa nästa
TEX	LDAB	BITCNT	Hämta räknarvärde
	PULX		Återställ register
	RTS		
BITCNT	RMB	1	Räknare för bitmönster

(7p)