

# Lösningsförslag 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)  $X_{2k} = 2^{10} - 641 = 1024 - 641 = 383$   $\underline{X}$  motsvarar  $\underline{-383}$   
 ( $y_9 = 0$ , pos)  $\underline{Y} = \underline{451}$   
 Resultatet R är korrekt ( $V = 0$ ). Korrekt resultat om  $V = 0$ .

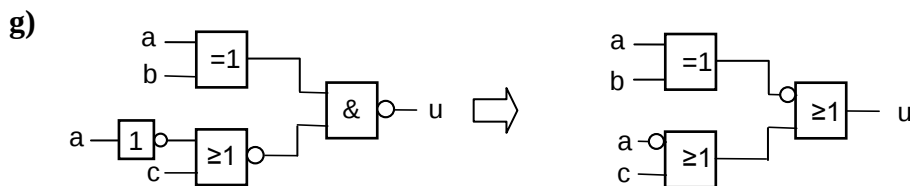
(1p)

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:  $0/111\ 1111\ 0/111\ 1111\ 11..11 = 7F7FFFFF_{16}$ .

(2p)



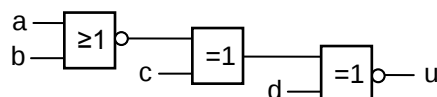
$$\underline{u} = (a \oplus b)' + a' + c = (a'b' + ab) + a'(b + b') + c = a'b' + ab + a'b + c = (ab')' + c = a' + b + c$$

(3p)

2.

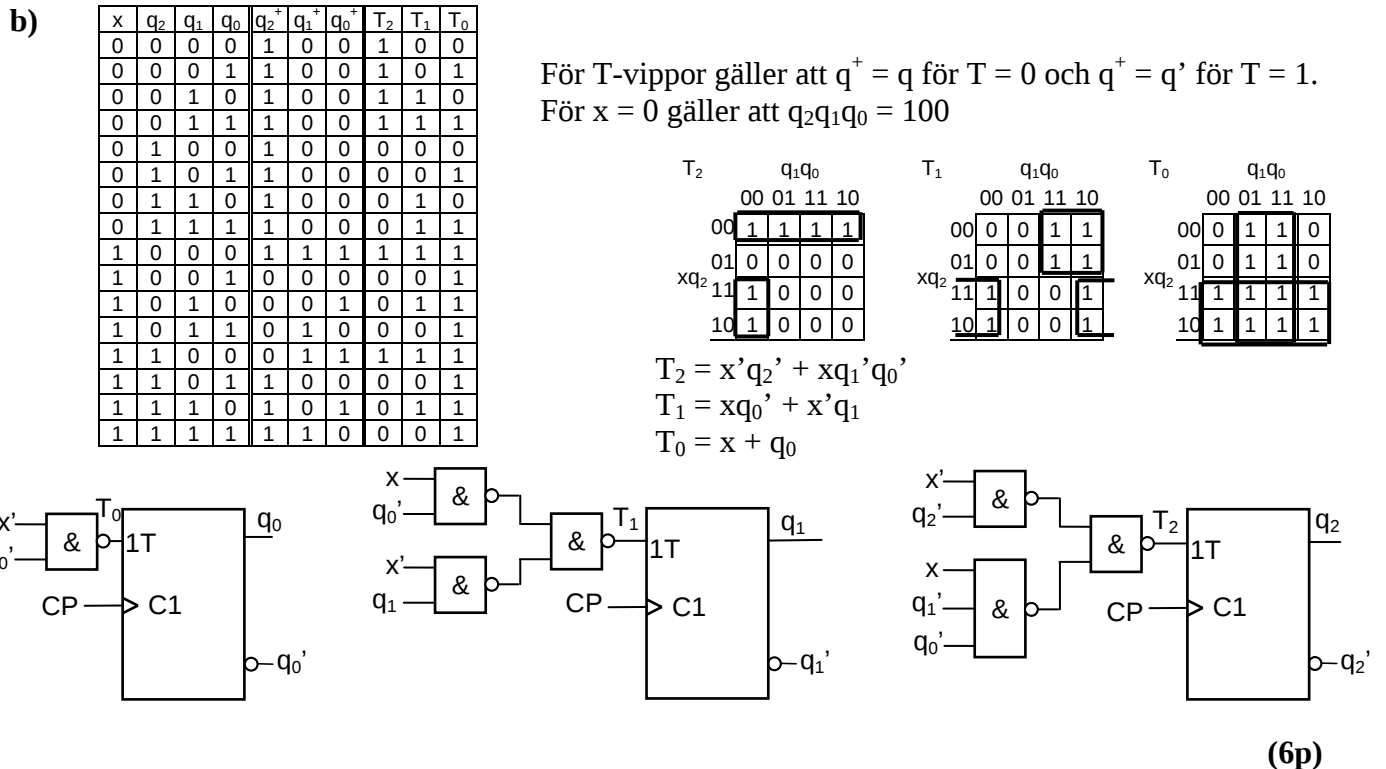
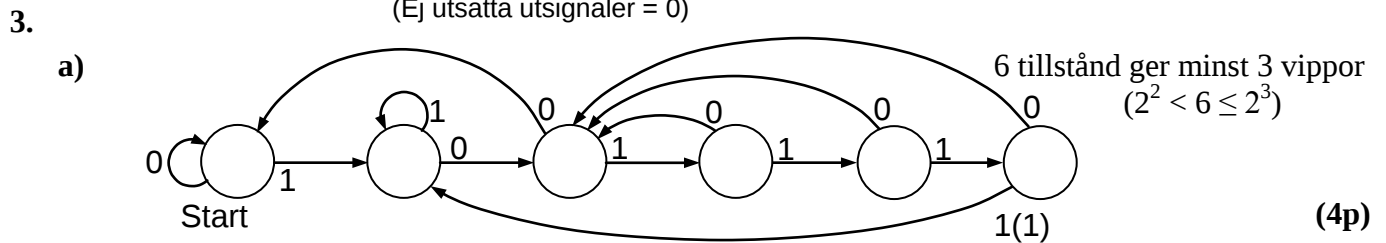
|    |    |    |    |    |    |
|----|----|----|----|----|----|
|    |    | cd |    |    |    |
|    |    | 00 | 01 | 11 | 10 |
| ab | 00 | 0  | 1  | 0  | 1  |
|    | 01 | 1  | 0  | 1  | 0  |
|    | 11 | 1  | 0  | 1  | 0  |
|    | 10 | 1  | 0  | 1  | 0  |

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



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

(6p)



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

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

5. a)

| State          | S-term                          | RTN-beskrivning    | Styrsignaler (=1)                                                   |
|----------------|---------------------------------|--------------------|---------------------------------------------------------------------|
| Q <sub>5</sub> | Q <sub>5</sub> ·I <sub>xx</sub> | PC → MA            | OE <sub>PC</sub> , LD <sub>MA</sub>                                 |
| Q <sub>6</sub> | Q <sub>6</sub> ·I <sub>xx</sub> | M → T              | MR, LD <sub>T</sub>                                                 |
| Q <sub>7</sub> | Q <sub>7</sub> ·I <sub>xx</sub> | X + T → R          | OE <sub>X</sub> , f <sub>3</sub> , f <sub>1</sub> , LD <sub>R</sub> |
| Q <sub>8</sub> | Q <sub>8</sub> ·I <sub>xx</sub> | R → PC, Next Fetch | OE <sub>R</sub> , LD <sub>PC</sub> , NF                             |

Instruktionen består av två ord. Andra ordet läses och placeras i T-registret. Innehållen 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 <sub>5</sub>  | Q <sub>5</sub> ·I <sub>F9</sub>  | PC → MA, PC+1 → PC | OE <sub>PC</sub> , LD <sub>MA</sub> , IncPC                         |
| Q <sub>6</sub>  | Q <sub>6</sub> ·I <sub>F9</sub>  | M → MA             | MR, LD <sub>MA</sub>                                                |
| Q <sub>7</sub>  | Q <sub>7</sub> ·I <sub>F9</sub>  | M → R              | MR, f <sub>0</sub> , LD <sub>R</sub>                                |
| Q <sub>8</sub>  | Q <sub>8</sub> ·I <sub>F9</sub>  | X → MA             | OE <sub>X</sub> , LD <sub>MA</sub>                                  |
| Q <sub>9</sub>  | Q <sub>9</sub> ·I <sub>F9</sub>  | R → M              | OE <sub>R</sub> , MW                                                |
| Q <sub>10</sub> | Q <sub>10</sub> ·I <sub>F9</sub> | X+1 → R            | OE <sub>X</sub> , f <sub>3</sub> , g <sub>0</sub> , LD <sub>R</sub> |
| Q <sub>11</sub> | Q <sub>10</sub> ·I <sub>F9</sub> | R → X, Next Fetch  | OE <sub>R</sub> , LD <sub>X</sub> , 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  $[-128, 127]$ .

$$50 = 80_{10}; \quad 80 - W \geq 0; \quad 80 \geq W; \quad -128 \leq W \leq 80 \text{ (tolkning)}$$

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:

$$0 \leq W \leq 80$$

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

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)

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

|         |      |            |                                       |
|---------|------|------------|---------------------------------------|
| 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)