

Lösningförslag tenta 2014-08-21

(Version 0 med reservation för eventuella fel. **Rapportera gärna fel, så uppdateras lösningen efterhand!** Uppdaterad 140821.)

1. $X = 011\ 1001_2$; $Y = 100\ 0111_2$ (7 bitars ordlängd)

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

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

c) $S = X + Y$

| | |
|----------|-----------|
| 76543210 | bitnummer |
| 11111110 | carry |
| 0111001 | X |
| +1000111 | Y |
| 0000000 | = S |

(1p)

d) $N = s_6 = 0$
 $Z = 1$ ($S = 0$)
 $V = x_6 * y_6 * s_6' + x_6' * y_6' * s_6 = 0 * 1 * 0' + 0' * 1' * 0 = 0$
 $C = c_7 = 1$

NZVC = 0101 (1p)

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

| | |
|----------|-----------------|
| 76543210 | bitnummer |
| 01110011 | carry |
| 0111001 | X |
| +0111000 | Y _{1k} |
| 1110010 | = D |

(1p)

f) $N = d_6 = 1$
 $Z = 0$ ($D \neq 0$)
 $V = x_6 * y_{6k} * d_6' + x_6' * y_{6k}' * d_6 = 0 * 0 * 1' + 0' * 0' * 1 = 1$
 $C = c_7' = 0' = 1$

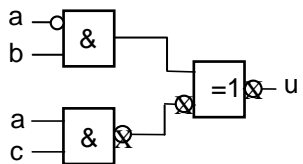
NZVC = 1011 (1p)

g) $X = 011\ 1001_2 = 39_{16} = 3 \cdot 16 + 9 = 57_{10}$
 $Y = 100\ 0111_2 = 47_{16} = 4 \cdot 16 + 7 = 71_{10}$
 $S = 000\ 0000_2 = 0$ Resultatet S är felaktigt eftersom $C = 1$.
 $D = 111\ 0010_2 = 72_{16} = 7 \cdot 16 + 2 = 114_{10}$ Resultatet D är felaktigt eftersom $C = 1$. (1p)

h) ($x_6 = 0$, pos) $X = 57_{10}$.
 ($y_6 = 1$, neg) $Y_{2k} = 2^7 - 71_{10} = 128 - 71_{10} = 57_{10}$ Y motsvarar -57_{10} .
 ($s_6 = 0$, pos) $S = 0$ Resultatet S är korrekt eftersom $V = 0$.
 ($d_6 = 1$, neg) $D_{2k} = 2^7 - 114_{10} = 14_{10}$ D motsvarar -14_{10} . D är felaktigt eftersom $V = 1$. (1p)

i) $N_{\max} = 1.111...1 * 2^{127} \approx 2 * 2^{127} = 2^{128} = 0,25 * 2^{130} = 0,25 * (2^{10})^{13} \approx 0,25 * (10^3)^{13} = 2,5 * 10^{38}$ (2p)

2.

a) 

$$\underline{u} = (a'b) \oplus (ac) = (a'b)'(ac) + (a'b)(ac)' = (a+b')ac + a'b(a'+c)' = ac + ab'e + a'b + a'be' = a'b + ac$$
(Nr 7) (2p)

b)

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

NOR-gindar \leftrightarrow "nollor"
 $f = [a+(b \oplus d)](a'+b)(a'+d)$ (Nr 6)

(4p)

3.

a)

| T | Q | Q ⁺ | S | R |
|---|---|----------------|---|---|
| 0 | 0 | 0 | 0 | - |
| 0 | 1 | 1 | - | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |

| S | Q |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 1 |
| 1 | 0 |

| R | Q |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

$S = T \oplus Q$
 eller
 $S = TQ'$

$R = (T \oplus Q)'$
 eller
 $R = TQ$ (AND-grindar)

(3p)

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

| q ₂ | q ₁ | q ₀ | T ₂ | T ₁ | T ₀ | q ₂ ⁺ | q ₁ ⁺ | q ₀ ⁺ |
|----------------|----------------|----------------|----------------|----------------|----------------|-----------------------------|-----------------------------|-----------------------------|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | - | - | - | - | - | - |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | - | - | - | - | - | - |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |

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

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

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

$T_2 = q_2q_1' + q_2'q_1 = q_2 \oplus q_1$
 $T_1 = q_1'q_0 + q_1q_0' = q_1 \oplus q_0$
 $T_0 = q_2q_0 + q_2'q_0' = (q_2 \oplus q_0)'$

(6p)

4.

| CP | Styrsignaler (=1) | RTN | A(8) | B(8) | T(8) | R(8) |
|----|---|----------------------|------|------|------|------|
| 1 | OE _B , LD _T | B → T | B4 | 30 | ? | ? |
| 2 | OE _A , f ₃ , f ₁ , LD _R , LD _T | A + T → R, A → T | B4 | 30 | 30 | ? |
| 3 | OE _R , f ₃ , f ₁ , f ₀ , g ₀ , LD _R | 2R + 1 → R | B4 | 30 | B4 | E4 |
| 4 | OE _R , f ₃ , f ₁ , f ₀ , LD _R | 2R → R | B4 | 30 | B4 | C9 |
| 5 | OE _R , f ₃ , f ₂ , LD _R , LD _T | R - T - 1 → R, R → T | B4 | 30 | B4 | 92 |
| 6 | OE _R , f ₃ , f ₁ , LD _R | R + T → R | B4 | 30 | 92 | DD |
| 7 | OE _R , LD _B | R → B | B4 | 30 | 92 | 6F |
| 8 | ? | | B4 | 6F | 92 | 6F |

(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} | U = A - T, ALU(NZVC) → CC, (New Fetch) | OE _A , f ₃ , f ₂ , g ₀ , LD _{CC} , NF |

Instruktionen är: CMPA Adr

(2p)

b)

| State | S-term | RTN-beskrivning | Styrsignaler |
|----------------|------------------------------------|---------------------------|---|
| Q ₄ | Q ₄ ·I _{DF} | M(PC) → TA, PC+1 → PC | MR, LD _{TA} , INC _{PC} |
| Q ₅ | Q ₅ ·I _{DF} | U = M(TA), ALU(NZVC) → CC | MR, g ₁₄ , f ₃ , f ₀ , LD _{CC} |
| Q ₆ | Q ₆ ·I _{DF} | M(PC) → T, PC+1 → PC | MR, LD _T , INC _{PC} |
| | Q ₆ ·I _{DF} Z' | (New Fetch) | NF |
| Q ₇ | Q ₇ ·I _{DF} | PC + T → R | OE _{PC} , f ₃ , f ₁ , f ₀ , LD _R |
| Q ₈ | Q ₈ ·I _{DF} | R → PC, (New Fetch) | OE _R , LD _{PC} , NF |

| |
|--------|
| OPKOD |
| Adr1 |
| offset |

(6p)

6.

a) FLIS(FLEX)-datorn använder minnesorienterad I/O. Fördelen med det är att vanliga instruktioner som läser eller skriver i minnet kan användas för I/O. I fallet separatadresserad I/O krävs speciella I/O-instruktioner. (2p)

b) Det villkorliga hoppet BPL Adr utförs om N-flaggan = 0.

Subtraktionen $50_{16} - D8_{16}$ med 8 bitar ger resultatet 78_{16} med N-flaggan = 0. Alltså utförs hoppet. (2p)

c)

| Adr | Data | ~ | Läge | Assemblerkod | | Offset |
|-----|-------|---|-------|--------------|---------|----------|
| | | | | Operation | Operand | |
| | | | | ORG | \$68 | |
| 68 | 10 | 3 | TIME | PSHA | | |
| 69 | 10 | 3 | LOOP1 | PSHA | | |
| 6A | F0 FB | 2 | | LDA | #NUMB | |
| 6C | 07 | 3 | LOOP2 | INCA | | |
| 6D | 00 | 2 | | NOP | | |
| 6E | 22 FC | 4 | | BMI | LOOP2 | 6C-70=FC |
| 70 | 14 | 3 | | PULA | | |
| 71 | 08 | 3 | | DECA | | |
| 72 | 00 | 2 | | NOP | | |
| 73 | 25 F4 | 4 | | BNE | LOOP1 | 69-75=F4 |
| 75 | 14 | 3 | TIMEX | PULA | | |
| 76 | 43 | 2 | | RTS | | |

(3p)

d) $T = 4 + 3 + [3 + 2 + (3 + 2 + 4) \cdot 5 + 3 + 3 + 2 + 4] \cdot 10 + 3 + 2 = 12 + 62 \cdot 10 = \underline{632} \text{ (}\mu\text{s)}$ (3p)

| | | | |
|--------|------|--------|--------------------------------|
| 7. | | | |
| TSUM | PSHY | | Spara register |
| | TFR | X,Y | Tabellpekare till Y |
| | STA | COUNT | Byteräknare |
| | CLRA | | Nollställ låg byte av summa |
| | LDX | #0 | Nollställ hög byte av summa |
| | TST | COUNT | Färdigt? |
| LOOP | BEQ | TSUMEX | Inga tal att addera |
| | ADDA | 0,Y | Addera tabellvärde till summan |
| | BCC | NOCY | Kolla carry |
| | LEAX | 1,X | Det blev carry. Öka hög byte |
| NOCY | LEAY | 1,Y | Öka tabellpekare |
| | DEC | COUNT | Minska byteräknare |
| | BRA | LOOP | Nästa |
| TSUMEX | PULY | | Återställ register |
| | RTS | | Retur med resultat i X:A |
| COUNT | RMB | 1 | Plats för byteräknare |

(6p)

