CS 380: Assignment 1

Jared Wasinger

January 23, 2017

1. (a) i. Maximum number of opcodes:
   Instruction Size: 1 word = 32 bits
   Answer: $2^{32} = 4294967296$

   ii. Address field size:
   $2^{17} = 131072 > 128000$
   Answer: 17 bits

   iii. Register field size:
   $2^5 = 32$ **Answer:** 5 bits

   iv. Opcode field size:
   32 - (opcode + register + address)
   $= 32 - (2 + 5 + 17)$
   $= 32 - 24$
   $= 8$ bits

   (b) i. PC = 17 bits (address size)
   ii. MAR = 17 bits (address size)
   iii. MDR = 32 bits (word size)
   iv. IR = 8 bits (opcode size)
   v. AC = 32 bits (word size)

2. (a) AND Y; $Y \leftarrow AC \wedge M(Y)$
   Fetch Cycle:
   - $MAR \leftarrow PC$
   - $MDR \leftarrow M(MAR), PC \leftarrow PC + 1$
   - $IR \leftarrow MDR_{opcode}, MAR \leftarrow MDR_{address}$

   Execute Cycle:
   - $MAR \leftarrow Y$
   - $MDR \leftarrow M(MAR)$
   - $AC \leftarrow AC \wedge MDR; MAR \leftarrow Y$
   - $MDR \leftarrow AC$
   - $M(MAR) \leftarrow MDR$

(b) ISZ Y; $M(Y) \leftarrow M(Y) + 1$, If(M(Y)+1=0) Then $PC \leftarrow PC + 1$
Fetch Cycle:

- $MAR \leftarrow PC$
- $MDR \leftarrow M(MAR), PC \leftarrow PC + 1$
- $IR \leftarrow MDR_{opcode}, MAR \leftarrow MDR_{address}$

Execute Cycle:

(c) DCA Y; $M(Y) \leftarrow AC; AC \leftarrow 0$
Fetch Cycle:

- $MAR \leftarrow PC$
- $MDR \leftarrow M(MAR), PC \leftarrow PC + 1$
- $IR \leftarrow MDR_{opcode}, MAR \leftarrow MDR_{address}$

Execute Cycle:

- $MAR \leftarrow Y$
- $MDR \leftarrow M(MAR)$
- $MAR \leftarrow MDR$, $MDR \leftarrow AC$
- $M(MAR) \leftarrow MDR$, $AC \leftarrow 0$

(d) JMS Y; $M(Y) \leftarrow PC, PC \leftarrow Y + 1$
Fetch Cycle:

- $MAR \leftarrow PC$
- $MDR \leftarrow M(MAR), PC \leftarrow PC + 1$
- $IR \leftarrow MDR_{opcode}, MAR \leftarrow MDR_{address}$

Execute Cycle:

- $MAR \leftarrow Y$, $MDR \leftarrow PC$
- $MDR \leftarrow M(MAR)$, $AC \leftarrow PC$
- $AC \leftarrow AC + 1$
- $PC \leftarrow AC$

3. **Execute Cycle**

- asdf

4. (i) MOV R1, R28
$R1 \leftarrow LOW(Y)$ $R1 \leftarrow 0x02$

(ii) LD R4, Y+
$R4 \leftarrow M(Y)$, $Y \leftarrow Y + 1$
$R4 \leftarrow 0x35$, $Y \leftarrow 0x0103$

(iii) LDI R4, 33
$R4 \leftarrow hex(33)$
$R4 \leftarrow 0x21$

(iv) ROL R3
$R3 = 0x07 = 0b00000111$
$R3 \leftarrow 0b00011111$
$R3 \leftarrow 0x1F$

SREG has carry bit set to 0:
$SREG \leftarrow 0b11111110$
$SREG \leftarrow 0xFE$

5.

```
    .ORG 0x000F
    LDI XH, high(CTR)
    LDI XL, low(CTR)
    LDI R31, 0xf0       ;R31 is the counter register
    CLR R5              ;Clear preset value of the result register
LOOP:
    CLC       ;Clear SREG carry bit (otherwise this loop wouldnt end)
    ROL R31
    BRCC SKIP ;If carry is cleared, there are no more 1's to be read
    INC R5    ;If carry is not cleared, add 1 to the result register
SKIP:
    CPI R31, 0x00   ;Check whether there are any 1s left to read
    BRNE LOOP       ;Repeat the original loop if there are
    ST X, R5        ;Store the resultant number of 1s in M(X)
DONE:
    JMP DONE
    .DSEG
CTR: .BYTE 1
```

Explanation:
This program counts the number of 1's in the number 0xF0. The program will loop through 'LOOP' four times, and then reach 'SKIP'. The statement 'CPI R31, 0x00' will evaluate to true causing the program to store the number of 1's in memory location pointed to by X and terminate. Memory location 'CTR' will hold 0x4 (the number of 1's in 0xF0).