CPE 233: Software Assignment 2
Prof. Bridget Benson
Jared Rocha and Luis Gomez

Behavior Description:

1. 8-bit unsigned value input is read from port id 0x30. If the input is greater than or equal 128 divide the input by 4 ignore any remainder. If the input is less than 128 multiple the input by 2. Output result to port id 0x42.
2. 8-bit unsigned value input is read from port id 0x30. If input is a multiple of 4 all bits are inverted. I f input is odd add 17 to input and divide the result by 2. If input is neither subtract one from the value. Output result to port id 0x42.

Flowchart:
1.

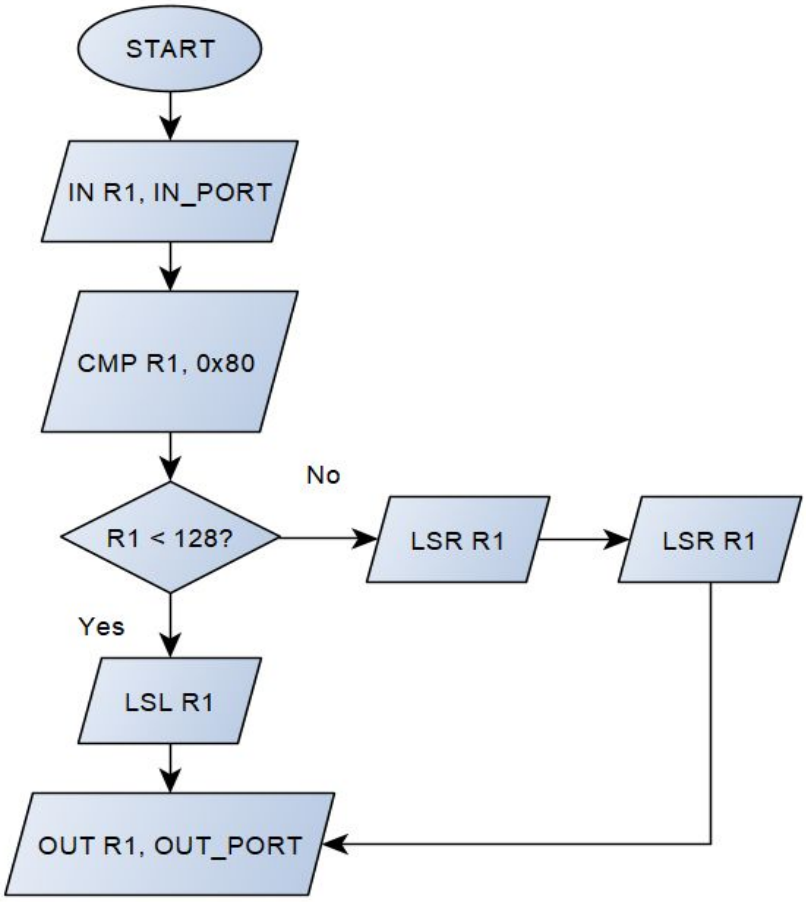

Figure 1: Flowchart for part 1

2.

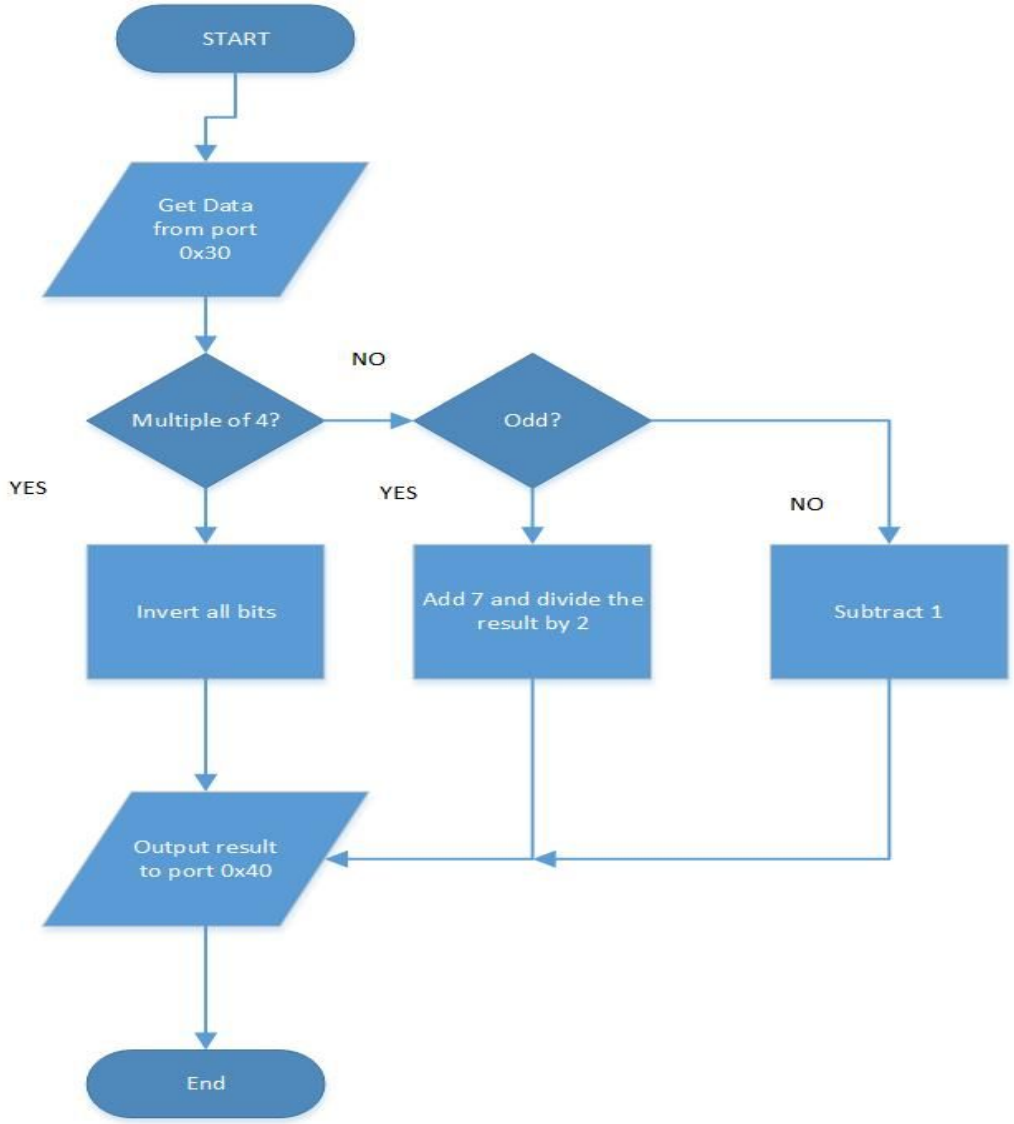

Figure 2: Flowchart for part 2

Verification

Figure 3 and 4 shows tested input values to test all zeros, all ones and target values.

Test #	Inputs to Port 0x30	Output to Port 0x42	C flag	Z flag	Explanation
1	0x00	0x00	0	0	Testing all zeros
2	0xFF	0xBF	1	0	Testing all ones
3	128	0x20	0	0	Testing 128 target value
4	127	0xFF	0	0	Testing 127 near target value

Figure 3: Test Table Part 1

Test #	Input to Port 0x30	Output to Port 0x42	C flag	Z flag	Explanation
1	0x00	0xFF	0	0	Testing all zeros
2	0xFF	0x88	0	0	Testing all ones
3	0x03	0x0A	0	0	Testing target value
4	0x02	0x01	0	0	Testing target value

Figure 4: Test Table Part 2

Assembly Source Code:

1.

```
;-------------------------------------------------------------------------------
; software assignment 2a
; by Jared Rocha and Luis Gomez
; date : 1/21/19
; description: Read an 8-bit unsigned value input from port id 0x30. If the
; input is greater than or equal to 128, the value is divided by 4. You can
; ignore any remainder. If the value is less than 128, the value is multiplied
; by 2. The result should be output to port id 0x42.
; registers used
; R1 data in
; R2 stores value 128
;-------------------------------------------------------------------------------
.EQU IN_PORT = 0x30
.EQU OUT_PORT = 0x42
.CSEG
.ORG 0x01
        IN   R1, IN_PORT
        ADD  R2, 128
        CMP  R1, R2
        BRCC divide         ; branch if input is >= 128
        lsl  R1             ; multiply R1 by 2
        BRN  end
divide: lsr  R1             ; divide R1 by 2
        lsr  R1             ; divide R1 by 2
end:    OUT  R1, OUT_PORT
```

Figure 5: Assembly code for part 1

2.

```
;-------------------------------------------------------------------------------
; software assignment 2b
; by Jared Rocha and Luis Gomez
; date : 1/21/19
; description: Read an 8-bit unsigned value input from port id 0x30. If the
; input value is a multiple of 4, all of the bits should be inverted,
; otherwise if the input value is odd, add 17 and divide the result by 2,
; otherwise subtract 1 from the value. The result should be output to port id
; 0x42.
; registers used
; R1 stores input value
; R3 stores value 0xFF
;-------------------------------------------------------------------------------
.EQU IN_PORT = 0x30
.EQU OUT_PORT = 0x42
.CSEG
.ORG 0x01
              IN   R1, IN_PORT
              TEST R1, 0x03        ; z == 1, R1 is a multiple of 4
              BREQ FOUR            ; branching if z == 1
              TEST R1, 0x01        ; z == 0, R1 is odd
              BRNE ODD             ; branching if z == 0
              SUB  R1, 1
              BRN  END
FOUR:         ADD  R3, 0xFF
              EXOR R1, R3          ; inverting all bits
              BRN  END
ODD:          ADD  R1, 17
              LSR  R1              ; divide R1 by 2
              BRN  END
END:          OUT  R1, OUT_PORT
              BRN  END
```

Figure 6: Assembly code from part 2