

## Software Assignment 1

CPE 233

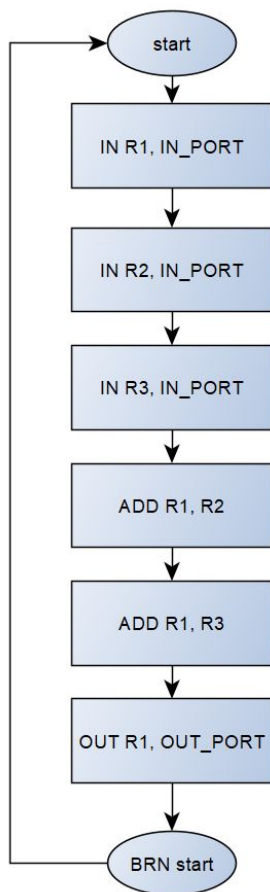
Luis Gomez & Brandon Grossman

### Behavior Description:

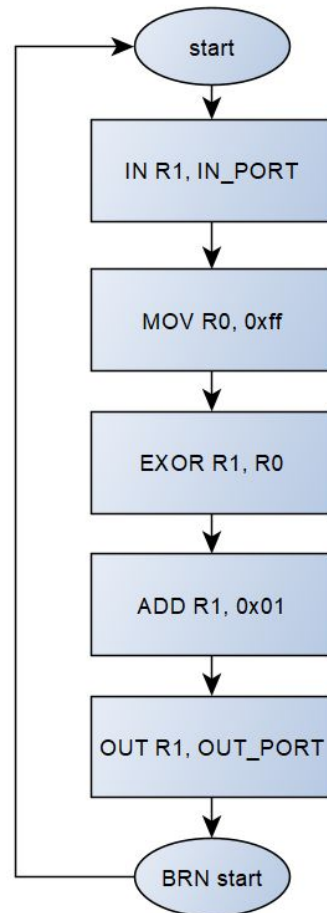
1. The program reads 3 values from port 0x30, adds them together, and outputs the result to port 0x40.
2. The program reads a value from port 0x30 and inverts the value. It outputs the value to port 0x40.

### Flow Chart:

1.



2.



**Verification:**

1.

Test	0x30 input	0x40 output	Explanation
1	0x01, 0x10, 0x15	0x26	$1+10+15=26 \Rightarrow 0x26$
2	0x00, 0x00, 0x1E	0x1E	$0+0+30=30 \Rightarrow 0x1E$
3	0x32, 0x64, 0x64	0xFA	$50+50+100=250 \Rightarrow 0xFA$
4	0x64, 0x64, 0x64	0x2C	$100+100+100=300-256=44 \Rightarrow 0x2C$ , Generates a carry

2.

Test	0x30 input	0x40 output	Explanation
1	0x01	0xFF	$0x01 \Rightarrow 1$ , $FF \Rightarrow 255$
2	0x32	0xCE	$0x32 \Rightarrow 50$ , $CE \Rightarrow 206$
3	0xFF	0x01	$0xFF \Rightarrow 255$ , $0x01 \Rightarrow 1$
4	0xF2	0x0E	$0xF2 \Rightarrow 242$ , $0x0E \Rightarrow 14$

**Assembly Source Code:**

1.

```

;-----
; Software Assignment Example
; Author: Luis Gomez and Brandon Grossman
; Date: 1/13/19

```

; Description: The program reads 3 values from port 0x30, adds them together, and outputs the result to port 0x40.

;

; Register uses:

; R0 - input value

; R1 - input value

; R2 - input value

;-----

.EQU IN\_PORT = 0x30

.EQU IN\_PORT = 0x40

.CSEG

.ORG 0x01

```
start:  IN R1, IN_PORT ; read val1
        IN R2, IN_PORT ; read val2
        IN R3, IN_PORT ; read val3
        ADD R1, R2 ; R1 + R2
        ADD R1, R3 ; Total sum
        OUT R1, OUT_PORT
        BRN start
```

2.

;-----

; Software Assignment Example

; Author: Luis Gomez and Brandon Grossman

; Date: 1/13/19

; Description: The program reads a value from port 0x30 and inverts the value. It outputs the value to port 0x40.

;

; Register uses:

; R0 - Necessary to invert values

; R1 - input/output value

;-----

.EQU IN\_PORT = 0x30

```
.EQU IN_PORT = 0x40
```

```
.CSEG
```

```
.ORG 0x01
```

```
start: IN    R1, IN_PORT    ; read 8-bit val
      MOV   R0, 0xff        ; loads 8-bit value (255)
      EXOR  R1, R0          ; inverts R0
      ADD   R1, 0x01        ; R0 + 1
      OUT   R1, OUT_PORT    ; output
      BRN  start
```