CPE 233: Software assignment 8

Prof. Bridget Benson

Luis Gomez, Isai Bolaños

Behavior

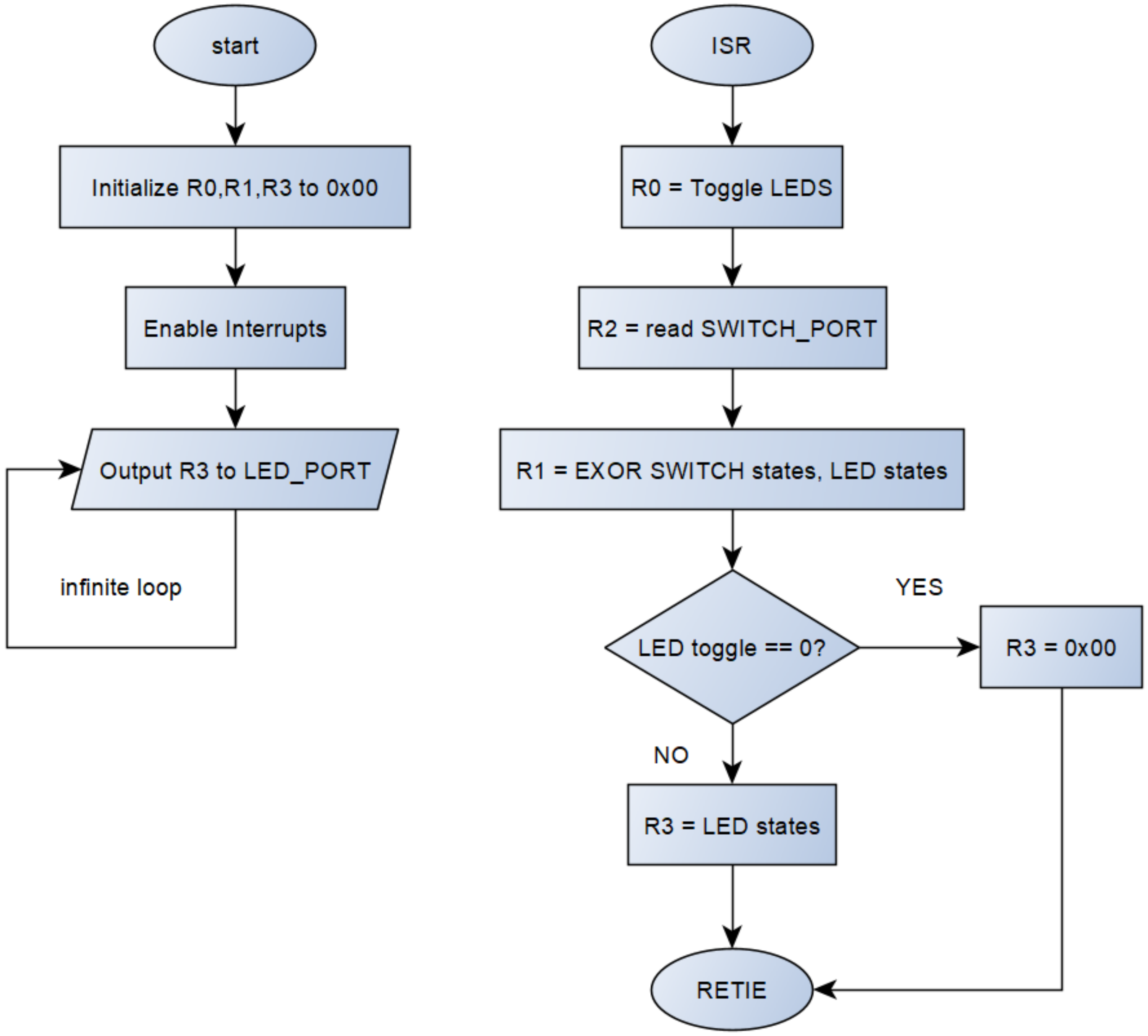
In this assignment, we wrote two Assembly programs and subroutines using the RAT simulator.

Program 1: A Rat Assembly Interrupt driven program. The interrupt driven program turns LEDS on and off. The Interrupt toggles LED output on and off, additionally the switch input is EXOR w/ the current LED output.

Input: 8-bit switch input from SWITCH_PORT.
Output: 8-bit led output from LED_PORT

Flowchart

Program 1



Verification

Program 1 Verification

Switch Input	Interrupt	LED output
-	0	00000000
10101010	1	10101010
-	0	10101010
10101011	1	00000000
-	0	00000000
00000000	1	00000001
-	0	00000001
11110001	1	00000000
-	0	00000000
00001111	1	00001111
-	0	00001111

Source Code

PROGRAM 1: 8b BCD converter

```
; This is an interrupt driven program that turns LEDS on and off
; Interrupt toggles LED output on and off, additionally the
; switch input is EXOR w/ the current LED output
; Input: 8-bit switch input from SWITCH_PORT
; Output: 8-bit led output from LED_PORT
.EQU LED_PORT = 0x42
.EQU SWITCH_PORT = 0x9A
.CSEG
.ORG 0x01
; Registers Used
;    R0 = LED TOGGLE
;    R1 = LED states
;    R2 = SWITCH states
;    R3 = Output
         MOV R0, 0x00
         MOV R1, 0x00
         MOV R3, 0x00
         SEI
loop:    OUT R3, LED_PORT  ; Output LEDS
         BRN loop

ISR: IN R2, SWITCH_PORT
         EXOR R0, 0x01     ; Toggle LEDs
         EXOR R1, R2           ; EXOR switches, LEDs
         CMP R0, 0x00      ; Are LEDs Enabled?
         BRNE tog_on
         MOV R3, R0            ; Output = Zero
         BRN exit
tog_on:  MOV R3, R1            ; Output = LED states
exit:    RETIE

.ORG 0x3FF
         BRN ISR
```

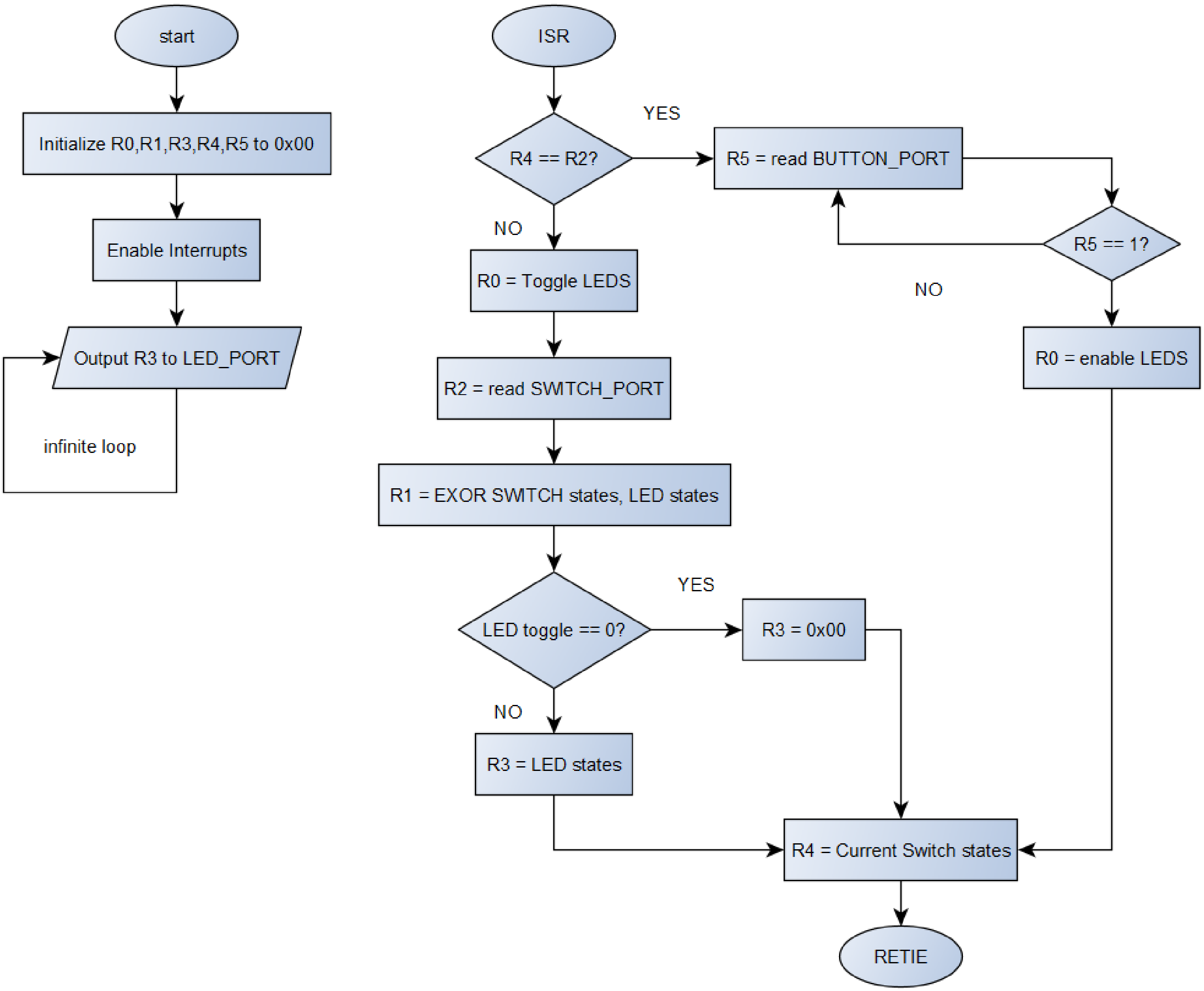
Behavior

Program 2: A Rat Assembly Interrupt driven program. The interrupt driven program turns LEDS on and off. The Interrupt toggles LED output on and off, additionally the switch input is EXOR w/ the current LED output. When the switch inputs are identical during two consecutive interrupt cycles the program toggles all LEDS off until a 1 is read from the 0-bit at BUTTON_PORT. When the button is pressed, the LED outputs the states before the system locked up.

Input: 8-bit switch input from SWITCH_PORT.
Input: 8-bit button input from BUTTON_PORT.
Output: 8-bit led output from LED_PORT

Flowchart

Program 2



Verification

Program 2 Verification

Switch Input	Button Input	Interrupt	LED output
-	-	0	00000000
10101010	-	1	10101010
-	-	0	10101010
10101010	-	1	10101010
-	0	-	10101010
-	1	-	10101010
-	-	0	10101010
10101011	-	1	00000000
-	-	0	00000000
10101011	-	1	00000000
-	1	-	00000000

PROGRAM 2: 16b Multiplier

```
; This is an interrupt driven program that turns LEDS on and off
; Interrupt toggles LED output on and off, additionally the
; switch input is EXOR w/ the current LED output
; Input: 8-bit switch input from SWITCH_PORT
; Output: 8-bit led output from LED_PORT
.EQU LED_PORT = 0x42
.EQU SWITCH_PORT = 0x9A
.EQU BUTTON_PORT = 0x9B
.CSEG
.ORG 0x01
; Registers Used
;     R0 = LED TOGGLE
;     R1 = LED states
;     R2 = SWITCH states
;     R3 = Output
;     R4 = BUTTON TOGGLE
;     R5 = Button state
           MOV R0, 0x00
           MOV R1, 0x00
           MOV R2, 0x00
           MOV R3, 0x00
           MOV R4, 0x00
           SEI
loop:      OUT R3, LED_PORT            ; Output LEDs
           BRN loop
;--------------------------------
ISR:       IN R2, SWITCH_PORT     ; Does current Switch input match previous?
           CMP R4, R2
           BREQ button
           EXOR R0, 0x01          ; Toggle LED output
           EXOR R1, R2            ; EXOR switches, leds
           CMP R0, 0x00
           BRNE tog_on
           MOV R3, R0             ; LED toggle == OFF, output is 0x00
           BRN exit
tog_on:    MOV R3, R1             ; LED toggle == ON, output is LED states
           BRN exit
button:    IN R5, BUTTON_PORT     ; Read button input until toggled
           CMP R5, 0x01
           BRNE button
           MOV R0, 0x01           ; Enable LEDS
exit:      MOV R4, R2             ; Previous Switch state = Current Switch State
           RETIE
;--------------------------------
.ORG 0x3FF
           BRN ISR
```