NAME: _____ STUDENT #: _____

EECE 259: Introduction to Microcomputers **QUIZ 2 – 201** **Feb 11, 2011**

Write a complete Nios II program that uses an infinite loop to control the LEDs on your DE1.

- Initially, all LEDs should be OFF. Assume only one switch can move at a time.
- When SW1 goes from 0 to 1, turn ON all red LEDs. If SW1 goes back to 0, leave them in their current state.
- When SW0 goes from 0 to 1, turn OFF all red LEDs. If SW0 goes back to 0, leave them in their current state.

```
.include "ubc-de1media-macros.s"
.global _start
.equ    SW0, 0x1
.equ    SW1, 0x2
.text

_start: movia   r23, IOBASE
        stwio   r0, LEDR(r23)          /* required */
        stwio   r0, LEDG(r23)          /* LEDG and HEX7SEG optional */
        stwio   r0, HEX7SEG(r23)       /* LEDG and HEX7SEG optional */

        ldwio   r2, SWITCH(r23) /* first switch reading */
        andi    r4, r2, SW1     /* r4 = remember previous SW1 setting */
        andi    r5, r2, SW0     /* r5 = remember previous SW0 setting */


loop:   ldwio   r2, SWITCH(r23)

check_SW1:
        andi    r6, r2, SW1            /* new SW1 reading */
        beq     r6, r0, check_SW0      /* if new SW1 == 0, check SW0 */
        bne     r4, r0, check_SW0      /* if prevSW1 != 0, check SW0 */
        movi    r2, 0x3ff              /* SW1 was a 0-to-1 change */
        stwio   r2, LEDR(r23)          /* turn on LEDR */

check_SW0:
        andi    r7, r2, SW0            /* new SW0 reading */
        beq     r7, r0, next_loop      /* if new SW0 == 0, restart the loop */
        bne     r5, r0, next_loop      /* if prevSW0 != 0, restart the loop */
        stwio   r0, LEDR(r23)          /* SW0 was 0-to-1 change, turn off LEDR */

next_loop:
        mov     r4, r6  /* new SW1 reading becomes previous SW1 */
        mov     r5, r7  /* new SW0 reading becomes previous SW0 */

        br      loop    /* read new SW positions */

.end
```

MARKING LEGEND

A (2 marks): SW1 functionality
B (2 marks): SW0 functionality
C (2 marks): Overall program functionality
L (2 marks): Loops infinitely
D (1 mark): Directives, movia r23 statement
S (1 mark): Valid syntax, valid switch masks

NAME: _____ STUDENT #: _____

EECE 259: Introduction to Microcomputers **QUIZ 2 – 202** **Feb 11, 2011**

Write a complete Nios II program that uses an infinite loop to control the LEDs on your DE1.

- Initially, all LEDs should be OFF. Assume only one switch can move at a time.
- If SW1 is a 1, all of the red LEDs are ON, otherwise all the red LEDs are OFF.
- When SW0 goes from 0 to 1, toggle the state of all green LEDs (turn them all ON if they were previously off, turn them all OFF if they were previously on). *Hint: xori with 0xff will toggle the lower 8 bits of a register.*

```
.include "ubc-de1media-macros.s"
.global _start
.equ    SW0, 0x1
.equ    SW1, 0x2
.text

_start: movia   r23, IOBASE
        stwio   r0, LEDR(r23)          /* required */
        stwio   r0, LEDG(r23)          /* required */
        stwio   r0, HEX7SEG(r23)       /* HEX7SEG optional */

        ldwio   r2, SWITCH(r23)        /* first switch reading */
        andi    r5, r2, SW0            /* r5 = remember previous SW0 setting */

        movi    r3, 0                  /* green LED state */

loop:   ldwio   r2, SWITCH(r23)

check_SW1:
        movi    r4, 0                  /* r4 = next LEDR state (default=OFF) */
        andi    r6, r2, SW1            /* new SW1 reading */
        beq     r6, r0, skip           /* if new SW1 == 0, do not change r4 */
        movi    r4, 0x3ff              /* r4 = next LEDR state = ON */
skip:   stwio   r4, LEDR(r23)          /* output r4 to LEDR */

check_SW0:
        andi    r7, r2, SW0            /* new SW0 reading */
        beq     r7, r0, next_loop      /* if new SW0 == 0, restart the loop */
        bne     r5, r0, next_loop      /* if prevSW0 != 0, restart the loop */
        xori    r3, r3, 0xff           /* toggle state of green LEDs */
        stwio   r3, LEDG(r23)          /* SW0 was 0-to-1 change, toggle LEDG */

next_loop:
        mov     r5, r7                 /* new SW0 reading becomes previous SW0 */

        br      loop                   /* read new SW positions */

.end
```

MARKING LEGEND

A (2 marks): SW1 functionality
B (2 marks): SW0 functionality
C (2 marks): Overall program functionality
L (2 marks): Loops infinitely
D (1 mark): Directives, movia r23 statement
S (1 mark): Valid syntax, valid switch masks