**Answer Sheet for NIOS Lab 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Engr 304 Lab** | | **Your Name:** | **Daniel Ackuaku** |
| **Q #** | **Description** | | **Answer** |
| Q1 | DATA reg value for certain SW | | DATA = 0x00024422 |
| Q2 | DATA reg value for certain LEDs | | DATA = 0x00012c4a |
| Q3 | DATA & EDGE for KEY3&1 | | DATA = 0x00000000  EDGE = 0x0000000e |
| Q4 | Why different? How used? | | The data port in memory stores the stores the for bit value that in pressed and held down on the corresponding keypad. The edge port in memory stores each keypress. |
| Q5 | DATA reg value for HEX 3..0 middle segment? | | DATA = 0x40404040 |
| Q6 | What do bits 7, 15, 23, 31 do? | | These bits control the middle segment of the HEX display. When they are one the display is one and it is off when they are 0. |
| Q7 | GP loading instruction? | | movia gp, 0x8000 |
| Q8 | DATA reading instruction? | | stwio r8, 0(gp) |
| Q9 | EDGE reading instruction? | | stwio r9, 12(gp) |
| Q10 | Describe what you expect the red LEDs to do | | I expected to see the LEDs count to 0x40000 in binary. |
| Q11 | What did the red LEDs do without the breakpoint? | | All the LEDs remain on constantly. |
| Q12 | DELAYMAXCNT value for 0.5 sec update period | | DELAYMAXCNT = 0x777777 |
| Q13 | PERIOD value in hex | | PERIOD = 0x17D7840 |
| Q14 | PERIODH and PERIODL values in hex | | PERIODH = 017D PERIODL = 7840 |

Signature validating Part I (dEAdbEEF): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature validating Part III (or email link to video): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Name: Daniel Ackuaku \*/

/\* Course: ENGR Lab #:5 \*/

/\* Date: 12th March, 2019 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.include "nios\_macros.s"

.include "nios\_defs.s" /\* .equ statements specific to this system \*/

/\* Place all constant definitions here (using ".equ" assembler directive) \*/

/\* The lines shown are just examples for you to look at. \*/

.equ MS100, 5000000 /\* number of clock cycles in 100 msec provided as example \*/

.equ MS100LOW, 0x4b40 /\* 16 least signif bits \*/

.equ MS100HIGH, 0x4c /\* 16 most signif bits \*/

.equ TIMER\_CONTROL\_VALUE, 0b0110

.equ BITMASK\_TIMEOUT, 0x01

.equ BITMASK\_KEY\_PRESS\_EDGE, 0b01000

.text

.org RESET\_VECTOR /\* Place the main routine at the reset address \*/

.global \_start

\_start: /\* Program start location must be identified \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MAIN \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* the main program startup code goes here \*/

MAIN\_PROG\_INIT:

movi r13, TIMER\_CONTROL\_VALUE

movi r14, 0x017d /\* the least significant bit for the timer\*/

movia r15, 0x7840 /\* the most significant bit for the timer\*/

movia r20, LEDR\_BASE /\* base location of PIO registers for LED device \*/

movia r22, TIMER\_BASE /\* base location for timer I/O device \*/

movia r23, KEY\_BASE /\* base location for timer KEY device \*/

movia r25, SW\_BASE /\* base location for timer SW\_BASE \*/

movia r10, 0x40000 /\* the limit for the counter i\*/

stwio r15, PERIODL(r22) /\* stores the most sig period bits into period H \*/

stwio r14, PERIODH(r22) /\* stores the least sig period bits into period H \*/

stwio r13, CONTROL(r22) /\* stores the stores into period H \*/

stwio r0, STATUS(r22)

mov r9, r0

/\* main program code goes here \*/

MAIN\_PROG:

br CHECK\_KEY\_PRESS

LED\_RESET:

mov r9, r0 /\*reset the counter i\*/

/\* initialize the timer with the proper timeout period \*/

TIMER:

ldwio r19, STATUS(r22) /\* reads the value of STATUS(TIMER) into r19 \*/

andi r19, r19, BITMASK\_TIMEOUT /\* uses a mask to get the value of TO \*/

bne r19, r0, DISPLAY

br CHECK\_KEY\_PRESS

DISPLAY:

stwio r0, STATUS(r22)

beq r9, r10, LED\_RESET

stwio r9, 0(r20)

addi r9, r9, 1

br TIMER

/\* Checks for a key press\*/

CHECK\_KEY\_PRESS:

ldwio r24, EDGE(r23) /\* reads the value of EDGE(KEYPRESS) into r23 \*/

andi r24, r24, BITMASK\_KEY\_PRESS\_EDGE /\* uses a mask to get the value of Key3 \*/

bne r24, r0, UPDATE\_TIMER

br TIMER

UPDATE\_TIMER:

stwio r0, EDGE(r23) /\* resets the value of EDGE(KEYPRESS) \*/

ldwio r24, DATA(r25) /\* EDGE(KEYPRESS) \*/

stwio r24, PERIODH(r22) /\* Update the period High\*/

stwio r13, CONTROL(r22) /\* resets the value of timer CONTROL \*/

mov r24, r0

br DISPLAY

.data

/\* if any global variables are needed, place them here \*/

.end