California State University, Northridge

Department of Electrical & Computer Engineering

Lab 6

Reading Inputs with Logic Operations

October 20, 2022

ECE 425L

Assigned by: Neda Khavari

Written By: Cristian Robles & Pathum Addarapathirana

Introduction:

In lab 6 we are introduced to using external peripherals to get inputs and execute them on the LPC2148. While reading the external peripherals we practice reading signals through port pins, using logic operations, and using the Keil simulator to simulate inputs.

Procedure:

Equipment Used

- Keil uVision4
- Keil Debugger
- LPC2148 Education Board

Description of Procedure

1. Set up startup code. This is the code needed to start the lab that includes the header below and the mystatup code given in lab 3.

```
                GLOBAL Reset_Handler
                AREA    mycode, CODE, Readonly
Reset_Handler   ;this label is necessary for the first line of your code
```

2. In the next part of our code we declare our constants and name some registers using the EQU and RN directives. This allows for more readable code

```
Bit14       EQU     0x00004000
PINSEL0     EQU     0xE002C000
IO0PIN      EQU     0xE0028000
IO0SET      EQU     0XE0028004
IO0DIR      EQU     0xE0028008
IO0CLR      EQU     0XE002800C
A_var       RN      6
B_var       RN      7
C_var       RN      8

Reset
            ;Initially Assigning A & B
            MOV B_var,#0
            MOV A_var,#10
            MOV C_var,#0
```

3. In the next part of the code we use the defined constants to set the pins as GPIO and set the direction of the signal for them as well. All pins except 14 were outputs because it is also connected to the button that we use as an input

```
                                ;Selecting funcion as GPIO by writing all zer
            MOV     r0,#0       ;moves #0 into register r0
            LDR     r1,=PINSEL0 ;puts what is stored in register 0xE002C000 (
            STR     r0,[r1]     ;Selecting signal direction of each port pin,
            LDR     r0,LEDPINS  ;Assigning output for all pins except pin 14
            LDR     r1,=IO0DIR  ;puts 0xE0028008 or IO0DIR to r1 register
            STR     r0,[r1]     ;copies value stored in r0 (0x0000BF00) to me
            MOV     r0,#0x0000FF00
            LDR     r1,=IO0CLR
            STR     r0,[r1]
            MOV     r0,#0x0000FF00
            LDR     r1,=IO0SET
            STR     r0,[r1]
```

4. After setting the pins to GPIO and making pin 14 an input we check constantly to see when the button connected to pin 14 has been pressed using the TST instruction. If it is pressed we branch to turnLow

```
41  task1
42              CMP     B_var,#8        ;B_var = 8 BEQ next loop
43              BEQ     task2
44
45              ;Checking pin14 status
46              LDR     r0,=IO0PIN
47              LDR     r9,[r0]
48              TST     r9,#0x00004000  ;testing if pin 14 (bina
49              BEQ     turnLow         ;(if above is zero -> tu
50
51  Tests
52              B       task1
```

5. After the branch our code adds 1 to the B variable to keep count of how many times it has been pressed which will be needed for task 2. We then turn on the first LED, reset PIN 14 to be an input, and then loop back to the point where the code checks for the next press

```
54  turnLow                             ;clears next light if pin14 pr
55              LDR     r0,=0x00000100  ;binary 100000000 (bit 8)
56              LSL     r0,B_var        ;Logical shift left of B_var
57              LDR     r1,=IO0CLR      ;forcing low
58              STR     r0,[r1]
59
60              ADD     B_var,B_var,#1  ;+1 B_var
61
62              ;Set Pin 14 to 1 again
63              ;Assigns ALL PINS to Input/Output, not sure why this n
64              LDR r0,=0x0000FF00
65              LDR r1,=IO0DIR
66              STR r0,[r1]             ;Assigning all as output pins
67
68              CMP B_var,#7            ;;;;;;i guess if it is at 7 wh
69              BEQ Case
70
71              LDR r0,LEDPINS          ;setting output for all pins b
72              LDR r1,=IO0DIR
73              STR r0,[r1]
74
75              B   Tests
```

6. After all the LEDs have been turned on the previous code no longer resets the value of pin 14 to make it an input again so we need to set pin 14 once again and keep checking how many times it has been pressed so we can complete task 2.

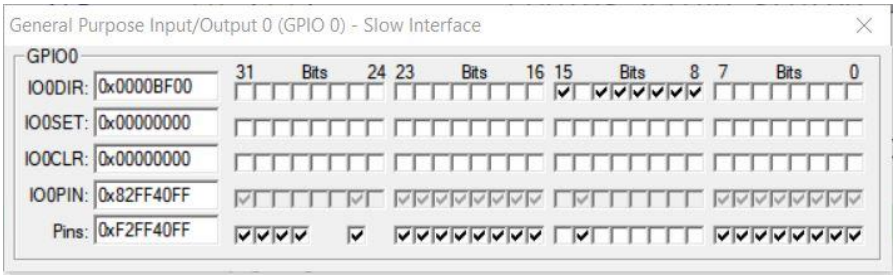
```
79
80              ;Set Pin 14 to 1 again
81              ;Assigns ALL PINS to Input/Output, not sure why th
82              LDR r0,=0x0000FF00
83              LDR r1,=IO0DIR
84              STR r0,[r1]             ;Assigning all as output p
85
86  check14
87              LDR r0,LEDPINS          ;setting output for all pi
88              LDR r1,=IO0DIR
89              STR r0,[r1]
90
91              LDR     r0,=IO0PIN
92              LDR     r9,[r0]
93              TST     r9,#0x00004000  ;testing if pin 14 (binary
94              BEQ     addone          ;(if above is zero -> turn
95              B       check14
```

7. We check for the value of B and when it is higher than 10 we branch to this part of the code which turns off all the LEDs which was the requirement for task 2

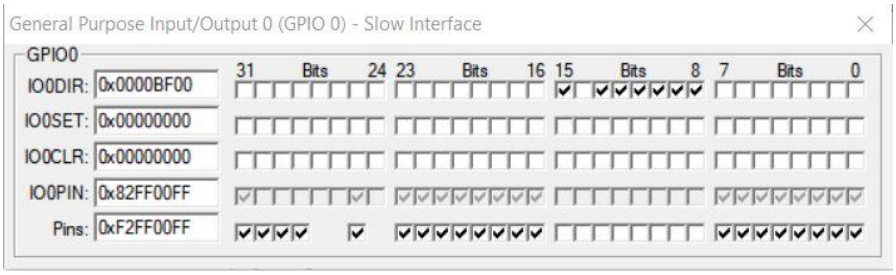
```
100
101         MOV r0,#0x0000FF00      ;forcing high with IO0SET to turn off all
102         LDR r1,=IO0SET
103         STR r0,[r1]
104
105         ADD     C_var,C_var,#1
106         CMP     C_var,#5
107         BEQ     task2b
108
109
110         B   FlashLights
111
112 task2b  ;checking if B_var=10 then end
113         ADD B_var,B_var,#1
114         CMP B_var,A_var
115         B   stop
```

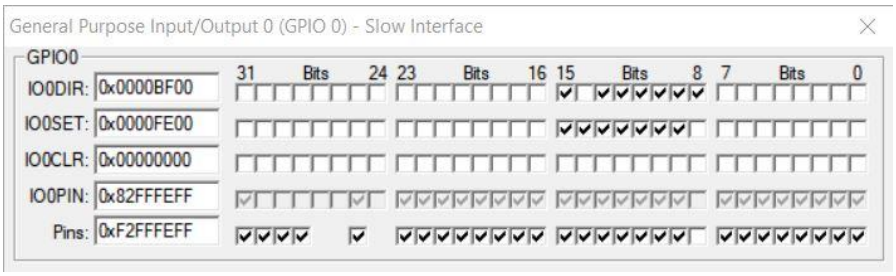
Results:

Task 1

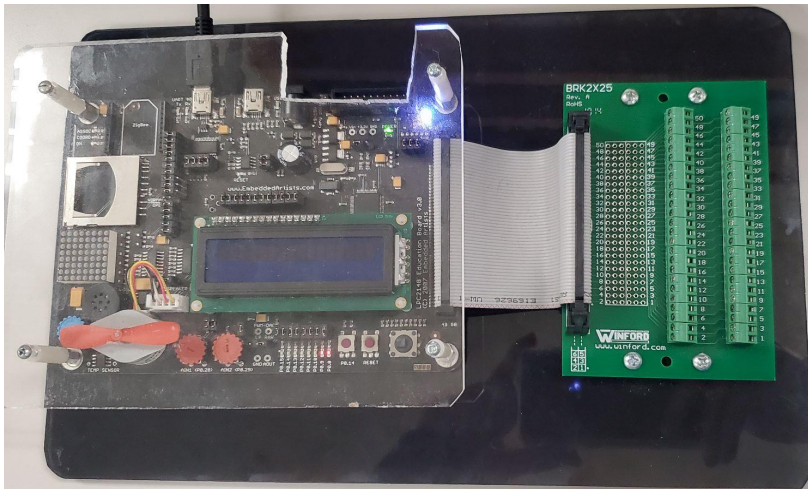


Simulated pins at the beginning of the program when pin 14 is the only one configured to be an input pin

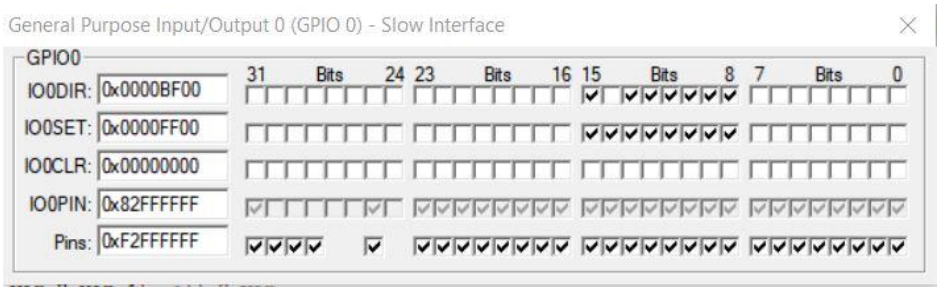




After the pin was pressed and we ran through the code we confirmed that our code would turn on the first LED and then set pin 14 back to what it was before it was pressed making it an input again

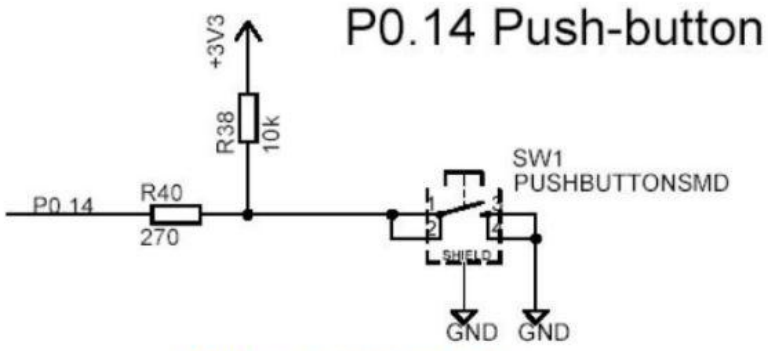


Task 2



After the count for the number of presses stored in B_var was higher than 10 we were able to turn off all the LEDs connected to pins 8-15 successfully

Questions:



1. Push-button design on board circuit.

Question 1: What logic level does the microcontroller read from its pin P0.14 when the switch (labeled as SW1 in the figure) is not pressed? What signal will be generated on pin P0.14 when the switch is pressed?

When the pin is not pressed the signal generated on pin 14 is 1. Once it is pressed the signal generated becomes 0

Question 2: You need to configure the pin P0.14 as input pin or output pin so that the microcontroller can read the status of the switch? (Hint: input or output is from the perspective of the microcontroller.)

For the microcontroller to be able to read the pin we need to configure the pin as an input

Question 3: What do you do if you need to clear bit 14 of the I/O direction register IO0DIR? Note that IO0DIR is assigned with a memory address therefore it can be treated as a memory location. Write a code segment to do that.

```
LDR   r0, 0x0000BF00
LDR   r1,=IO0DIR
STR   r0,[r1]
```

Question 4: Should you check or uncheck the pin P0.14 to simulate a button pressing action?

To simulate the button being pressed within the Keil Debugger you would uncheck the box

Conclusion:

In conclusion, we were able to successfully read input pin values on pin 14 on the LPC 2148 using logic operations. Since the pin is connected to both an LED and a button we had to take extra steps to make sure the light was an output pin when it needed to be and it was an input pin when it needed to be. For this we had to check constantly for when it was pressed and make sure to reset the pin to an input after the corresponding LEDs turned on. One thing that could improve this lab would be adding a debouncer to the button as on the physical board one press would sometimes register multiple presses. This project furthered our knowledge of GPIO and logic operations.