EE2016: Experiment 8

Group:19

Sparsh Gupta	Charan Srikkanth	Kaushik Iyer
EE23B117	EE23B127	EE23B135

October 23, 2024

1 Introduction

This experiment focuses on how to use the LPC2378 and write ARM code to display a specific number in its binary representation.

2 Objectives

The primary objectives of this experiment are:

1. Learning about the various special registers that are present in the LPC2378 in order to setup the IO pins
2. Using these registers to light up LEDs according to the number which had been inputted

The way we do this is:

1. Find the register addresses of PINSEL10,PINSEL4,FIO2DIR,FIO2PIN
2. Set PINSEL10 to 0 to disable the ETM function
3. PINSEL4 is used to set how the pins are configured ie whether they are GPIO pins,RX-TX pins,etc.In order to set all the pins as GPIO pins we set PINSEL4 as 0
4. FIO2DIR register is used to set the data direction of the pins by setting the last 8 bits.1 is considered as output and 0 as input.Set FIO2DIR accordingly
5. FIO2PIN register is used to set the output.To set the pin as high we make the bit 1 and 0 to set the pin low.Set those bits to 1 where you want the LED to glow.

2.1 Assembly code

```
AREA LED,CODE,READONLY
ENTRY
EXPORT SystemInit
EXPORT __main
; Fill the port register addresses from the LPC2378 manual
    PINSEL10 EQU 0xE002C028
    FIO2DIR EQU 0x3FFFC041  ;This is Dir1 for Px8 to Px15
    PINSEL4 EQU 0xE002C010
    FIO2PIN EQU 0x3FFFC055 ;This is for Px8 to Px15
SystemInit

; use PINSEL10 first to disable ETM function
; of FIO2 port pins (see p. 166 of user manual)
    ldr R1, PINSEL10 ; load the value at PINSEL10 into R1
    LDR R0,=PINSEL10
    MOV R2,#0x00000000; load appropriate constant
    STR R2,[R0]

    ;PINSEL4 is used is used to set the what the pins are used for here as they are used as GPIO pins
    LDR R0,=PINSEL4; load the address of appropriate PINSELx
    MOV R2,#0x00000000; load appropriate constant
    STR R2,[R0]

    ;This is used to set the Direction of the pins by setting the last 8 bits.I have set all the bits
    LDR R0,=FIO2DIR
    MOV R2,#0x000000FF
    STR R2,[R0]

; Add code designed above (that sets various registers)
__main
; Code to display a number on LEDs begins


forever
    LDR R0,=FIO2PIN; use the appropriate register to set values
;Set those bits high which you want the LED to turn on.
    MOV R2, #0x00000003 ;Will display number 3
    STR R2,[R0]

b forever
END
```

2.2 Blinking LED

The bonus assignment was to display the number given as input and blink the LEDs

1. In order to do this we set FIO2PIN as the number we want.
2. Set a delay by a loop according to how many clock cycles we want.
3. Set FIO2PIN to 0 to switch off all the LEDs
4. Set a delay again.
5. Run all the steps in loop

2.3 Assembly Code

```
AREA LED,CODE,READONLY
ENTRY
EXPORT SystemInit
EXPORT __main
; Fill the port register addresses from the LPC2378 manual
    PINSEL10 EQU 0xE002C028
    FIO2DIR EQU 0x3FFFC041  ;This is Dir1 for Px8 to Px15
    PINSEL4 EQU 0xE002C010
    FIO2PIN EQU 0x3FFFC055 ;This is for Px8 to Px15
SystemInit

; use PINSEL10 first to disable ETM function
; of FIO2 port pins (see p. 166 of user manual)
    ldr R1, PINSEL10 ; load the value at PINSEL10 into R1
    LDR R0,=PINSEL10
    MOV R2,#0x00000000; load appropriate constant
    STR R2,[R0]

    ;PINSEL4 is used is used to set the what the pins are used for here as they are used as GPIO pins
    LDR R0,=PINSEL4; load the address of appropriate PINSELx
    MOV R2,#0x00000000; load appropriate constant
    STR R2,[R0]

    ;This is used to set the Direction of the pins by setting the last 8 bits.I have set all the bits
    LDR R0,=FIO2DIR
    MOV R2,#0x000000FF
    STR R2,[R0]

; Add code designed above (that sets various registers)
__main
; Code to display a number on LEDs begins


forever
    LDR R0,=FIO2PIN; use the appropriate register to set values
;Set those bits high which you want the LED to turn on.
    MOV R2, #0x00000003 ;Will display number 3
    STR R2,[R0]

;Setting 2560*2560 clock cycle delay
    MOV R4,#2560
    MOV R3,#2560
outer_delay
    ADD R4,R4,#-1
inner_delay
        ADD R3,R3,#-1
        CMP R3,#0
        BPL inner_delay
    BPL outer_delay

MOV R2, #0x00000000
STR R2,[R0]

;Setting 2560*2560 clock cycle delay
```

```
    MOV R4,#2560
    MOV R3,#2560
outer_delay2
    ADD R4,R4,#-1
inner_delay2
        ADD R3,R3,#-1
        CMP R3,#0
        BPL inner_delay2
    BPL outer_delay2

b forever
END
```

3 Comments

1. It took some time to figure out how each register works and their addresses.Required sufficient time to read the manual.
2. Setting the frequency of the blinking LED was a challenge we had to try many numbers in order to get a perfect frequency which neither too slow nor too fast.
3. Writing the perfect ARM code took some time.