EE2016 - Microprocessor (Theory + Lab)
Experiment-8: ARM-based microcontroller I/O Ports and LED Interfacing

In this experiment, we will write assembly language programs for interfacing light emitting diodes to an ARM-based microcontroller from NXP semiconductors. In particular, we will use the MCB2300 Keil-ARM board that is equipped with an LPC2378 microcontroller from NXP. We will write programs to send signals from the microcontroller to the LEDs on the board. The block diagram of the LPC2378 is shown in Figure 1.

1 Brief Description of the Software Environment

In addition to the selections made in Keil microvision for the ARM assembly language programming experiment, you need to do the following for this hardware experiment with the Keil-ARM board.

1. Choose 'Yes' when prompted to copy LPC2300.s Startup file.
2. Right click on Target 1, choose "Options for Target 'Target 1' ".
3. Under the Target tab, select 'Use Microlib'.
4. Choose Project $--> $ Build Target (assuming you have already added the assembly language program file to the project).
5. Then select the Debug tab and click on "Settings " in the ULINK2/ME ARM Debugger.
6. Set 'Max JTAG Clock' as 200kHz.
7. Connect the ULINK2 debugger and select "Flash $--> $ Download".

2 Ports and Register Settings

In LPC2378, there are a few registers whose settings determine the configuration of a port for our experiments (with LEDs). These are (i) PINSEL10 (ii) FIO2DIR (iii) FIO2PIN (iv) PINSEL4. We will have an address associated with each of these. (Please see Remark 1 for additional explanation regarding addresses for port registers.) There are many other registers as far as the LPC2378 is concerned (but we will not be using them for the experiment). The purpose of each port register is explained below.

- In general, port pins have multiple functions (input/output, serial receive, serial transmit etc.) and hence we use PINSELx registers to *specify the function*. For example, when specific register bits are 00, a certain function is performed by the corresponding port pin while the same pins will perform a different function when the register bits are 01.

  In the MCB2300 board, eight LEDs are connected to the lowest 8 pins of Fast Input/Output Port 2 (FIO2). The pins of FIO2 are shared between I/O and the Embedded Trace Module (ETM) interface. So in order to use the LEDs, the ETM interface has to be disabled. This is done by setting bit 3 of PINSEL10 register to 0. Further, we have to select FIO2 lower 8-bits as I/O. Each pair of bits in PINSEL4 sets the function of one pin of FIO2. Setting the pair to 00 selects the pin for I/O function. To make all the eight pins as I/O, we have to set 8 pairs of bits in PINSEL4 to 0.

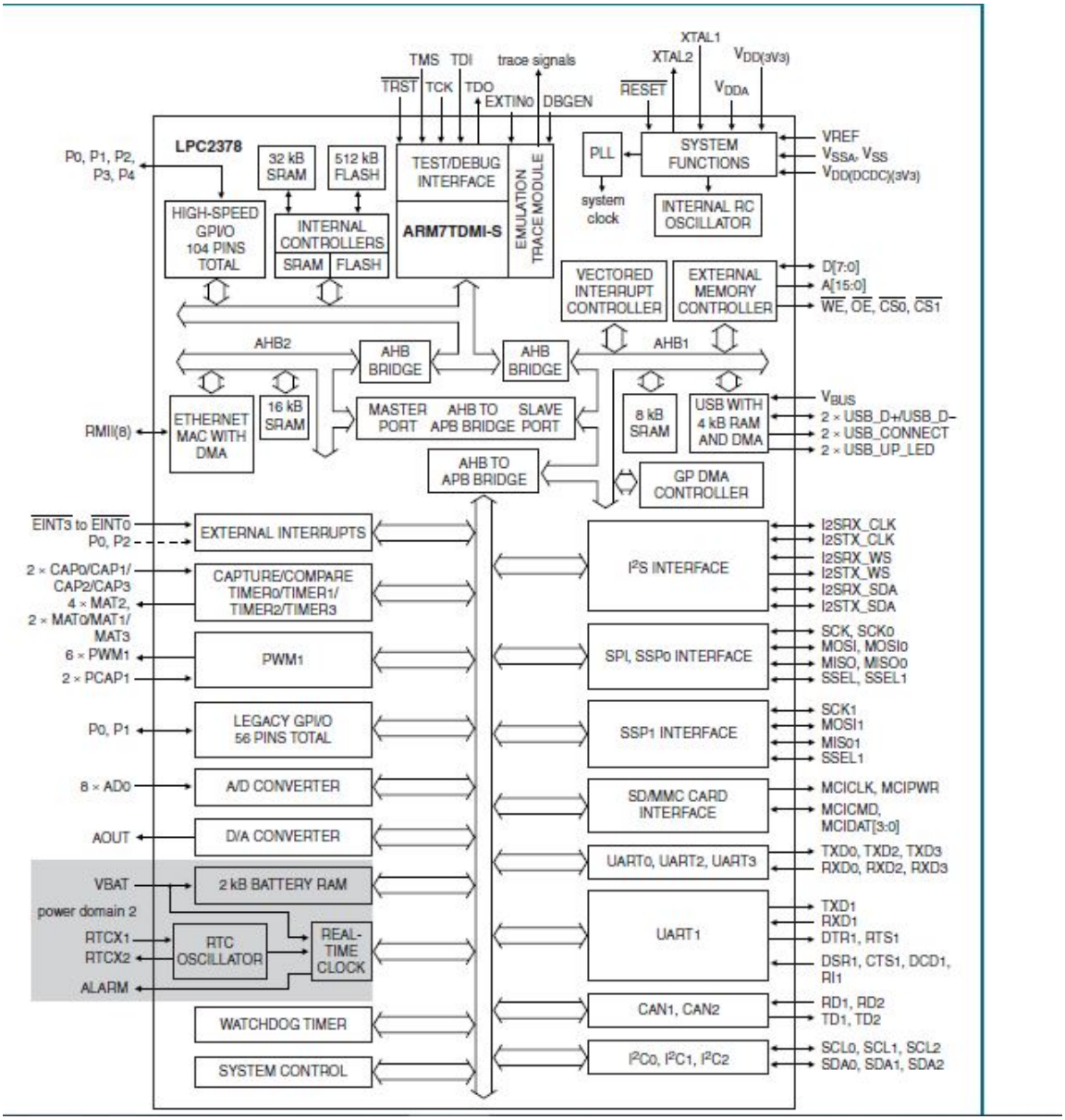

Figure 1: LPC2378

- Every port has a direction register (to set the direction of each pin of the port). A setting of 1 in the direction register sets the corresponding pin of the port as output while a setting of 0 makes the pin as input. The specific register of interest to us is FIO2DIR.
- Once the port pins are selected and the direction is set, we have to specify the port register which is directly related to the (numerical) value to be displayed. This is accomplished with the FIO2PIN register. FIO2PIN is chosen (instead of FIO1PIN or FIO3PIN for instance) since we have chosen to use fast input/output port 2 above for the LED interfacing task.

In your program, you need to specify the address for the port registers as well as do certain operations on some of them (to configure) before writing the actual code for LED display/blinking. Additional details about ports and registers are available in Chapters 9 and 10 in the LPC23XX User Manual uploaded on Moodle.

Remark 1 *Just as we have addresses for external memory, there are also addresses for registers associated with GPIO ports. In other words, we have what is called memory mapped input-output. Registers such as r0, r1 etc. that we have used in assembly language programs are referred (to) by their names (no address is typically associated with them).*

3 Tasks for the Experiment

The various tasks for which you need to complete/write assembly language programs are described below. The first major objective is to display a number on the LEDs on the board. Thereafter, we want to cause the LEDs to blink.

Task 1: Complete the following code to specify addresses for the various registers. Use the LPC2378 user manual (chapters 9 and 10) for this purpose.

```
    AREA LED,CODE,READONLY
    ENTRY
    EXPORT SystemInit
    EXPORT __main

; Fill the port register addresses from the LPC2378 manual
PINSEL10 EQU ..........
FIO2DIR EQU  ..........
PINSEL4 EQU ..........
FIO2PIN EQU ............
```

Task 2: Complete the code below for setting the PINSEL10 register. Open *Peripherals −− > Pin Connect Block* and see the initial setting for PINSEL10 (and justify the same using the description/tables in the LPC2378 user manual). Then fill in the blanks suitably.

```
SystemInit

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

Task 3: Complete the code below to set PINSEL4 suitably (for the same task of lighting LEDs).

```
    LDR R0,.......; load the address of appropriate PINSELx
```

```
    MOV R2,.......; load appropriate constant
    STR R2,[R0]
```

Task 4: Fill in the code below to set FIO2DIR suitably.

```
LDR R0, .....
MOV R2, ......
STR ......
```

Task 5: Complete the program below to display a number on the LEDs on the MCB2300 board. Use FIO2PIN suitably. Demonstrate the working of the code for different choices for the number to be displayed.

```
; Add code designed  above (that sets  various registers)

__main

; Code to display a number on LEDs begins
    LDR R0, .......; use the appropriate register to set values
    MOV R2, .......
    STR R1,[R0]
forever  b forever
    END
```

Task 6: In the previous task, we focussed on display of a fixed number on the LEDs. Modify the program for task 5 suitably to make the LEDs blink. In particular, add code for delay.

4 Report

Submit a report on the experiment on Moodle (within a week of this experiment). One report per group (with the names of the group members) is sufficient. The report should contain details of the solution (including the code) and your observations and experience (in programming, debugging etc.). Please note that reports that closely match those of other groups will be penalized.