To initialize an I/O port for general use we perform seven steps. Steps two through four are needed only for the LM4F/TM4C microcontrollers. First, we activate the clock for the port. Second, we unlock the port; unlocking is needed only for pins PC3-0, PD7, PF0 on the LM4F and TM4C. Third, we disable the analog function of the pin, because we will be using the pin for digital I/O. Fourth, we clear bits in the **PCTL** (Table 6.1) to select regular digital function. Fifth, we set its direction register. Sixth, we clear bits in the alternate function register, and lastly, we enable the digital port. We need to add a short delay between activating the clock and accessing the port registers. The direction register specifies bit for bit whether the corresponding pins are input or output. A **DIR** bit of 0 means input and 1 means output.

Common Error: You will get a bus fault if you access a port without enabling its clock.

Help

In this first example we will make PF4 and PF0 input, and we will make PF3 PF2 and PF1 output, as shown in Program 6.1. To use a port we first must activate its clock in the **SYSCTL_RCGC2_R** register. The second step is to unlock the port, by writing a special value to the **LOCK** register, followed by setting bits in the **CR** register. Only PC3-0, PD7, and PF0 on the TM4C need to be unlocked. All the other bits on the TM4C are always unlocked. The third step is to disable the analog functionality, by clearing bits in the **AMSEL** register. The fourth step is to select GPIO functionality, by clearing bits in the **PCTL** register, as described in Table 6.1. The fifth step is to specify whether the pin is an input or an output by clearing or setting bits in the **DIR** register. Because we are using the pins as regular digital I/O, the sixth step is to clear the corresponding bits in the **AFSEL** register. The last step is to enable the corresponding I/O pins by writing ones to the **DEN** register. To run this example on the LaunchPad, we also set bits in the **PUR** register for the two switch inputs (Figure 6.3) to have an internal pull-up resistor.

When the software reads from location 0x400253FC, the bottom 8 bits are returned with the current values on Port F. The top 24 bits are returned zero. As shown in Figure 6.7, when reading an I/O port, the input pins show the current digital state, and the output pins show the value last written to the port. The function **PortF_Input** will read from the five input pins, and return a value depending on the current status of the inputs. As shown in Figure 6.7, when writing to an I/O port, the input pins are not affected, and the output pins are changed to the value written to the port. The function **PortF_Output** will write new values to the three output pins. The **#include** will define symbolic names for all the I/O ports for that microcontroller. The header file **tm4c123ge6pm.h** can be found in the **inc** folder.

```
#include "tm4c123ge6pm.h"

unsigned long In;  // input from PF4
unsigned long Out; // output to PF2 (blue LED)
//   Function Prototypes
void PortF_Init(void);
// 3. Subroutines Section
// MAIN: Mandatory for a C Program to be executable
```

```
int main(void){    // initialize PF0 and PF4 and make them inputs
  PortF_Init();    // make PF3-1 out (PF3-1 built-in LEDs)
  while(1){
    In = GPIO_PORTF_DATA_R&0x10;   // read PF4 into Sw1
    In = In>>2;                    // shift into position PF2
    Out = GPIO_PORTF_DATA_R;
    Out = Out&0xFB;
    Out = Out|In;
    GPIO_PORTF_DATA_R = Out;        // output
  }
}
// Subroutine to initialize port F pins for input and output
// PF4 is input SW1 and PF2 is output Blue LED
// Inputs: None
// Outputs: None
// Notes: ...
void PortF_Init(void){ volatile unsigned long delay;
  SYSCTL_RCGC2_R |= 0x00000020;     // 1) activate clock for Port F
  delay = SYSCTL_RCGC2_R;           // allow time for clock to start
  GPIO_PORTF_LOCK_R = 0x4C4F434B;   // 2) unlock GPIO Port F
  GPIO_PORTF_CR_R = 0x1F;           // allow changes to PF4-0
  // only PF0 needs to be unlocked, other bits can't be locked
  GPIO_PORTF_AMSEL_R = 0x00;        // 3) disable analog on PF
  GPIO_PORTF_PCTL_R = 0x00000000;   // 4) PCTL GPIO on PF4-0
  GPIO_PORTF_DIR_R = 0x0E;          // 5) PF4,PF0 in, PF3-1 out
  GPIO_PORTF_AFSEL_R = 0x00;        // 6) disable alt funct on PF7-0
  GPIO_PORTF_PUR_R = 0x11;          // enable pull-up on PF0 and PF4
  GPIO_PORTF_DEN_R = 0x1F;          // 7) enable digital I/O on PF4-0
}
```

Program 6.1. A set of functions using PF4,PF0 as inputs and PF3-1 as outputs (C6_InputOutputxxx.zip).

CHECKPOINT 6.3

Does the entire port need to be defined as input or output, or can some pins be input while others are output?

Hide Answer

Since there are as many bits in a port as there are bits in the direction register, each bit can be individually programmed as input or output.

CHECKPOINT 6.4

How do we change Program 6.1 to run using Port A?

Hide Answer

First, write a 0x00000001 to the clock register to activate Port A. Second, change all the labels PORTF to PORTA. If it is an LM3S microcontroller, remove steps 2, 3, and 4.

EdX is a non-profit created by founding partners Harvard and MIT whose mission is to bring the best of higher education to students of all ages anywhere in the world, wherever there is Internet access. EdX's free online MOOCs are interactive and subjects include computer science, public health, and artificial intelligence.

Help