Chapter 3

GENERAL PURPOSE INPUT/OUTPUT (GPIO) FUNCTIONALITY

3.1 Pin Multiplexing (MUX) and General Purpose I/O Overview

Due to the limited number of physical pins on the LF2407 DSP, it is necessary to multiplex two functions onto most of the pins. That is, each pin can be programmed for either a primary or secondary (GPI/O) function (see Fig. 3.1). Once the pins on the LF2407 are multiplexed, the effective pin-out of the device is doubled. This provides enough effective pin-out for six General Purpose Input Output (GPIO) ports to be configured as the secondary function on most pins. Each Input/Output Port (IOP) consists of eight pins when they are configured to their secondary function.

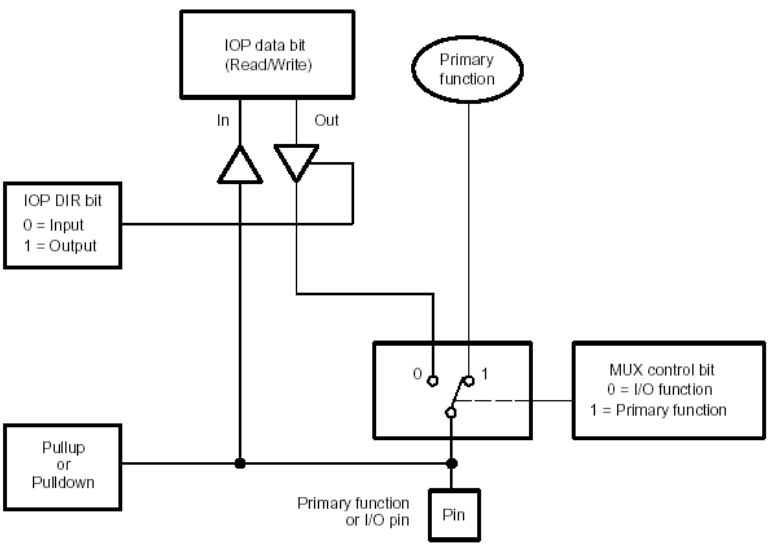

Figure 3.1 Block diagram of the multiplexing of a single pin. (Courtesy of Texas Instruments)

GPIO pins are grouped in sets of eight pins called ports. There are six ports total, ports A through F. Even though the pins are grouped in ports, each pin can be individually configured as primary or secondary (GPIO) functionality; and if GPIO, then either input or output. The multiplexing of primary pin functions with secondary GPIO functions provides a flexible method of controlling both the dedicated and secondary pin functions.

Each multiplexed pin's primary/secondary functionality is controlled by a corresponding bit in the appropriate MUX control register. Additionally, when the pin is in GPIO mode, there are port data and direction (PxDATDIR) control

registers which control the direction (input or output) and data of the port/pin. If the pin is configured as an output, then the data (voltage) on the pin is determined by what value is written to the pin's data bit. Inversely, if the pin is configured as an input, then the voltage level applied to the pin determines the value of the pin's corresponding data bit.

If the pin is configured as an output pin, it can either be set to a logic high "1" (3.3 Volts) or a logic low "0" (0 Volts) by writing to its corresponding data bit in the corresponding PxDATDIR register. If the pin is configured as an input, the pin's corresponding bit in the appropriate PxDATDIR register will be "1" if 3.3 Volts or "0" if 0 Volts is applied to the pin. The data bits in the PxDATDIR can then be read by the user code and the values used in the program. The input and output ports provide a convenient way to input or output binary data (each pin = 1 bit). For example, a seven-segment display could be controlled by a GPIO port configured as output.

***Note**: There is no relationship between the GPIO pins and the I/O space of the LF2407.*

3.2 Multiplexing and General Purpose I/O Control Registers

The three MUX control registers and six data/direction control registers are all mapped to data memory (see Table 3.1). They control all dedicated and shared pin functions:

- I/O MUX Control Registers (MCRA, MCRB, MCRC): These 16-bit registers determine whether a pin will operate in its primary function or secondary GPIO function. Two ports are assigned to each MUX control register. For example, the MCRA register controls ports A and B.

- Data and Direction Control registers (PxDATDIR): Once a pin is configured in I/O mode by the appropriate MUX control register, the appropriate PxDATDIR register is used to configure each pin as input or output; and if output, whether the pin is high (3.3 Volts) or low (0 Volts).

Table 3.1 GPIO Control Register Summary

Data Memory Address	Register Name	Description
7090h	MCRA	I/O MUX Control Register A
7092h	MCRB	I/O MUX Control Register B
7094h	MCRC	I/O MUX Control Register C
7098h	PADATDIR	I/O Port A Data and Direction Register
709Ah	PBDATDIR	I/O Port B Data and Direction Register
709Ch	PCDATDIR	I/O Port C Data and Direction Register
709Eh	PDDATDIR	I/O Port D Data and Direction Register
7095h	PEDATDIR	I/O Port E Data and Direction Register
7096h	PFDATDIR	I/O Port F Data and Direction Register

3.2.1 I/O Multiplexing (MUX) Control Registers

I/O MUX Control Register A (MCRA) Configuration

15	14	13	12	11	10	9	8
MCRA.15	MCRA.14	MCRA.13	MCRA.12	MCRA.11	MCRA.10	MCRA.9	MCRA.8
RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0
7	6	5	4	3	2	1	0
MCRA.7	MCRA.6	MCRA.5	MCRA.4	MCRA.3	MCRA.2	MCRA.1	MCRA.0
RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0

Note: *R = read access, W = write access, -0 = value after reset.*

		Pin Function Selected	
Bit #	**Name.bit #**	**(MCA.n = 1) (Primary)**	**(MCA.n = 0) (Secondary)**
0	MCRA.0	SCITXD	IOPA0
1	MCRA.1	SCIRXD	IOPA1
2	MCRA.2	XINT1	IOPA2
3	MCRA.3	CAP1/QEP1	IOPA3
4	MCRA.4	CAP2/QEP2	IOPA4
5	MCRA.5	CAP3	IOPA5
6	MCRA.6	PWM1	IOPA6
7	MCRA.7	PWM2	IOPA7
8	MCRA.8	PWM3	IOPB0
9	MCRA.9	PWM4	IOPB1
10	MCRA.10	PWM5	IOPB2
11	MCRA.11	PWM6	IOPB3
12	MCRA.12	T1PWM/T1CMP	IOPB4
13	MCRA.13	T2PWM/T2CMP	IOPB5
14	MCRA.14	TDIRA	IOPB6
15	MCRA.15	TCLKINA	IOPB7

I/O MUX Control Register B (MCRB) Configuration

15	14	13	12	11	10	9	8
MCRB.15	MCRB.14	MCRB.13	MCRB.12	MCRB.11	MCRB.10	MCRB.9	MCRB.8
RW-1	RW-1	RW-1	RW-1	RW-1	RW-1	RW-1	RW-0
7	6	5	4	3	2	1	0
MCRB.7	MCRB.6	MCRB.5	MCRB.4	MCRB.3	MCRB.2	MCRB.1	MCRB.0
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-1	RW-1

Note: *R = read access, W = write access, -0 = value after reset.*

Bit #	Name.bit #	Pin Function Selected (MCB.n = 1) (Primary)	Pin Function Selected (MCB.n = 0) (Secondary)
0	MCRB.0	W/R	IOPC0
1	MCRB.1	BIO	IOPC1
2	MCRB.2	SPISIMO	IOPC2
3	MCRB.3	SPISOMI	IOPC3
4	MCRB.4	SPICLK	IOPC4
5	MCRB.5	SPISTE	IOPC5
6	MCRB.6	CANTX	IOPC6
7	MCRB.7	CANRX	IOPC7
8	MCRB.8	XINT2/ADCSOC	IOPD0
9	MCRB.9	EMU0	Reserved
10	MCRB.10	EMU1	Reserved
11	MCRB.11	TCK	Reserved
12	MCRB.12	TDI	Reserved
13	MCRB.13	TDO	Reserved
14	MCRB.14	TMS	Reserved
15	MCRB.15	TMS2	Reserved

I/O MUX Control Register C (MCRC) Configuration

15	14	13	12	11	10	9	8
Reserved	Reserved	MCRC.13	MCRC.12	MCRC.11	MCRC.10	MCRC.9	MCRC.8
		RW–0	RW–0	RW–0	RW–0	RW–0	RW–0
7	6	5	4	3	2	1	0
MCRC.7	MCRC.6	MCRC.5	MCRC.4	MCRC.3	MCRC.2	MCRC.1	MCRC.0
RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–1

Note: *R = read access, W = write access, -0 = value after reset.*

		Pin Function Selected	
it #	**Name.bit #**	**(MCC.n = 1) (Primary)**	**(MCC.n = 0) (Secondary)**
0	MCRC.0	CLKOUT	IOPE0
1	MCRC.1	PWM7	IOPE1
2	MCRC.2	PWM8	IOPE2
3	MCRC.3	PWM9	IOPE3
4	MCRC.4	PWM10	IOPE4
5	MCRC.5	PWM11	IOPE5
6	MCRC.6	PWM12	IOPE6
7	MCRC.7	CAP4/QEP3	IOPE7
8	MCRC.8	CAP5/QEP4	IOPF0
9	MCRC.9	CAP6	IOPF1
10	MCRC.10	T3PWM/T3CMP	IOPF2
11	MCRC.11	T4PWM/T4CMP	IOPF3
12	MCRC.12	TDIRB	IOPF4
13	MCRC.13	TCLKINB	IOPF5
14	MCRC.14	Reserved	IOPF6
15	MCRC.15	Reserved	Reserved

3.2.2 Port Data and Direction Control Registers

Port A Data and Direction Control Register (PADATDIR)

15	14	13	12	11	10	9	8
A7DIR	A6DIR	A5DIR	A4DIR	A3DIR	A2DIR	A1DIR	A0DIR
RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0
7	6	5	4	3	2	1	0
IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0
RW–†	RW–†	RW–†	RW–†	RW–†	RW–†	RW–†	RW–†

† The reset value of these bits depends upon the state of the respective pins.

Note: *R = read access, W = write access, -0 = value after reset.*

Bits 15–8 AnDIR – Direction Bits

0 Configure corresponding pin as an input
1 Configure corresponding pin as an output

Bits 7–0 IOPAn – Data Bits

If AnDIR = 0, then:

0 Corresponding I/O pin is read as a low
1 Corresponding I/O pin is read as a high

If AnDIR = 1, then:

0 Set corresponding I/O pin low
1 Set corresponding I/O pin high

Port B Data and Direction Control Register (PADATDIR)

15	14	13	12	11	10	9	8
B7DIR	B6DIR	B5DIR	B4DIR	B3DIR	B2DIR	B1DIR	B0DIR
RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0
7	6	5	4	3	2	1	0
IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0
RW–†	RW–†	RW–†	RW–†	RW–†	RW–†	RW–†	RW–†

† The reset value of these bits depends upon the state of the respective pins.

Note: *R = read access, W = write access, -0 = value after reset.*

Bits 15–8 BnDIR – Direction Bits

0 Configure corresponding pin as an input
1 Configure corresponding pin as an output

Bits 7–0 IOPBn – Data Bits
If BnDIR = 0, then:
0 Corresponding I/O pin is read as a low
1 Corresponding I/O pin is read as a high
If BnDIR = 1, then:
0 Set corresponding I/O pin low
1 Set corresponding I/O pin high

Port C Data and Direction Control Register (PCDATDIR)

15	14	13	12	11	10	9	8
C7DIR	C6DIR	C5DIR	C4DIR	C3DIR	C2DIR	C1DIR	C0DIR
RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0
7	6	5	4	3	2	1	0
IOPC7	IOPC6	IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0
RW–†	RW–†	RW–†	RW–†	RW–†	RW–†	RW–†	RW–x

† The reset value of these bits depends upon the state of the respective pins.

Note: *R = read access, W = write access, -0 = value after reset, x = undefined.*

Bits 15–8 CnDIR – Direction Bits
0 Configure corresponding pin as an input
1 Configure corresponding pin as an output

Bits 7–0 IOPCn – Data Bits
If CnDIR = 0, then:
0 Corresponding I/O pin is read as a low
1 Corresponding I/O pin is read as a high
If CnDIR = 1, then:
0 Set corresponding I/O pin low
1 Set corresponding I/O pin high

Port D Data and Direction Control Register (PDDATDIR)

15-9	8
Reserved	D0DIR
	RW–0
7-1	0
Reserved	IOPD0
	RW–†

† The reset value of this bit depends upon the state of the respective pins.

Note: *R = read access, W = write access, -0 = value after reset.*

Bits 15–9 Reserved

Bit 8 D0DIR – Direction Bits

0 Configure corresponding pin as an input
1 Configure corresponding pin as an output

Bits 7–1 Reserved

Bit 0 IOPD0 – Data Bit

If D0DIR = 0, then:

0 Corresponding I/O pin is read as a low
1 Corresponding I/O pin is read as a high

If D0DIR = 1, then:

0 Set corresponding I/O pin low
1 Set corresponding I/O pin high

Port E Data and Direction Control Register (PEDATDIR)

15	14	13	12	11	10	9	8
E7DIR	E6DIR	E5DIR	E4DIR	E3DIR	E2DIR	E1DIR	E0DIR
RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0
7	6	5	4	3	2	1	0
IOPE7	IOPE6	IOPE5	IOPE4	IOPE3	IOPE2	IOPE1	IOPE0
RW–†	RW–†	RW–†	RW–†	RW–†	RW–†	RW–†	RW–x

† The reset value of these bits depends upon the state of the respective pins.

Note: *R = read access, W = write access, -0 = value after reset, x = undefined.*

Bits 15–8 EnDIR – Direction Bits

0 Configure corresponding pin as an input
1 Configure corresponding pin as an output

Bits 7–0 IOPEn – Data Bits

If EnDIR = 0, then:

0 Corresponding I/O pin is read as a low
1 Corresponding I/O pin is read as a high

If EnDIR = 1, then:

0 Set corresponding I/O pin low
1 Set corresponding I/O pin high

Port F Data and Direction Control Register (PFDATDIR)

15	14	13	12	11	10	9	8
Reserved	F6DIR	F5DIR	F4DIR	F3DIR	F2DIR	F1DIR	F0DIR
	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0	RW–0
7	6	5	4	3	2	1	0
Reserved	IOPF6	IOPF5	IOPF4	IOPF3	IOPF2	IOPF1	IOPF0
	RW–†	RW–†	RW–†	RW–†	RW–†	RW–†	RW–†

† The reset value of these bits depends upon the state of the respective pins.

Note: *R = read access, W = write access, -0 = value after reset.*

Bit 15 Reserved

Bits 14–8 FnDIR – Direction Bits

0 Configure corresponding pin as an input
1 Configure corresponding pin as an output

Bit 7 Reserved

Bits 6–0 IOPFn – Data Bits

If FnDIR = 0, then:

0 Corresponding I/O pin is read as a low
1 Corresponding I/O pin is read as a high

If FnDIR = 1, then:

0 Set corresponding I/O pin low
1 Set corresponding I/O pin high

3.3 Using the General Purpose I/O Ports

The GPIO functionality is relatively simple to use and provides a valuable way of imputing and outputting data to and from the DSP. To use the GPIO functionality of a particular pin or groups of pins, the following steps must be followed to configure the DSP:

1. Set the bits in the appropriate MUX control register to configure the desired pins for GPIO function. This can be done by writing a "0" to the corresponding bits in the appropriate MUX. It may not be absolutely necessary to do this due to the fact that upon a reset (power on) the pins in the LF2407 are by default in their GPIO functionally. However, configuring the MUX register anyway is good programming practice.

2. Now that the desired pins are configured as GPIO, set the Port Data and Direction (PxDATDIR) register(s) that corresponds to the desired pins. When configuring the PxDATDIR, the most significant bits control the direction (input or output) and the lower bits determine (output) or display

(input) the pin data. If an input pin is desired, only the direction bit needs to be set since when the direction bit is set to input, writing to the data bit has no effect. The corresponding data bit will be used to display the logic value applied to that pin. If an output pin is required, both the direction and data bits need to be configured because the data bit will determine what logic value the pin will be set to.

3. The selected pins are now configured. The input data on pins can be obtained by reading the entire PxDATDIR register and obtaining the data for desired bits. For output, new values can be written to the pins by writing to the corresponding entire PxDATDIR register.

***Note**: When a pin is configured as input, it is important to note whether the pin has either a pull-up or pull-down resistor. If the input pin is not connected to anything, the pin's data bit will read "1" if a pull-up or "0" if a pull-down resistor exists. The pullup/down resistor comes into play only when the pin is an input and not connected. When the input pin is connected to either a logic "1" or logic "0" voltage, the pull-up/down resistor is overridden and has no effect. The reason behind the pullup/down resistor is that a digital input pin should never be completely floating.*

Example 3.1 illustrates configuring all the pins in a port for output and writing "1" to each of the eight pins in the port.

Example 3.1 Display the binary number "00100010"b with the eight pins on port A:

1. Configure the bits in MCRA so port A is I/O ("XXXXXXXX00000000"). The most significant bits in MCRA control port B; therefore, in this example, we do not care what we write to them.

2. Set pins to output in PADATDIR by setting bits 15-8 as "11111111", with "00100010"b as the data (bits 7-0). "X" designates "don't care" bits.

3.4 General Purpose I/O Exercise

This exercise allows the reader to become familiar with using the GPIO functionality on the LF2407 DSP controller. Practical applications using the GPIO functionality are very similar to the algorithms presented in this exercise. In addition, this exercise helps the reader practice writing assembly programs.

The XF pin is introduced during this exercise. The XF pin on the LF2407 is a general purpose output pin which is controlled by the XF bit in the C2xx DSP core. Because it is core controlled, the XF bit can be set and cleared without having to write to a register. The XF is easily set (made 3.3V) by the "SETC XF" command or cleared (made 0V) by the CLRC XF command. This pin can be useful in testing

code to see if your code ran to a certain point, where the code would set or clear the XF bit. The level of the XF pin may be checked by either an oscilloscope or volt meter. In order to assure a correct signal is read, be sure the ground lead of the tester is connected to the digital ground of the LF2407 EVM.

1. Startup CCS and create a new project titled "lab3" in the same manner as the exercise in the previous chapter. Name the source file "lab3.asm" and include the same header files as before.

2. Write a program that first stores a certain set of values into data memory locations starting at memory address 300h. The values should be such that they control the "up counting" of a seven-segment display from "0" to "F". The program should then read the memory locations and send the values out on port A; displaying each value for a second or so.

3. Place a seven-segment LED display on a breadboard and connect the port A pins to the display in the appropriate positions.

4. Run the program and watch as the seven-segment display "counts" from 0 to F.

5. When the display has finished counting a 1000-hertz square wave should be produced by toggling the XF pin.

6. Connect the oscilloscope to the XF pin and digital ground of the LF2407 EVM. Use the oscilloscope to view the 1000-hertz waveform.

This exercise is now concluded.