General Purpose I/O on MCF52259

The general purpose I/O module of the MCF52259 allows you to use almost any I/O pin as either a digital input or output where you can read from or write the value of the I/O pin via software control. However, most I/O pins can be used for several alternative functions only one of which is GPIO (i.e. a pin may be used as an Analog to Digital converter, as a USB input, or as GPIO). Thus, some initialization code may be required before an I/O pin can be used for this purpose. I/O pins are grouped into "ports" based on common functionality. Refer to the Coldfire 5225x Reference Manual chapter 15 (and Figure 15-1 for alternative pin functions).These ports can be controlled in a single access or on a bit by bit basis.

The MCF52259on-board I/O functions (2 Pushbuttons, 4 DIP switches, 4 LED's, 3-axis accelerometer and a potentiometer) are mapped to the following ports.

I/O Function	GPIO Port	Notes
PushButton - SW1	PORTTA[0]	Produces '0' when Pressed 0
PushButton - SW3 (there is no SW2)	PORTTA[1]	
DIP Switches 4 downto 1	PORTDD[7:4]	Produces '0' when in the 'ON' position
LED4 downto LED1	PORTTC[3:0]	Light is ON when a '1' is written
Accelerometer X-Axis	PORTAN[0]	
Accelerometer Y-Axis	PORTAN[1]	
Accelerometer Z-Axis	PORTAN[2]	
Potentiometer	PORTAN[3]	

Table 1 – MCF52259OnBoard I/O Connections

To control an I/O pin we must follow the procedure outlined below:

1. Initialize the port's "Pin Assignment Register" (PxxPAR) to indicate that the pins should be used for GPIO vs. there alternate functions. Usually writing a value of all 0's indicates GPIO.
2. Initialize the port's "Data Direction Register" (DDRxx) to select whether the pins should be used as inputs (write a '0' to the appropriate bit in the PxDDR) or outputs (write a '1' to the appropriate bit in the PxxDDR)
3. We are now ready to use the I/O pins as outputs or inputs:
    a. If the pins of the port are to be used as outputs, we can simply write a value to the PORTxx register which will then drive the actual I/O pins with the given value.
    b. If the pins of the port are to be used as inputs, we can now read from the SETxx register address to sample the current value of I/O pin.

Register	Description
PxxPAR **(Port Assignment Register):**	The value in this register configures the I/O pin for GPIO or one of its alternate functions. Usually a PxPAR value of 0's indicate GPIO.
DDRxx **(Data Direction Register):**	The value in this register configures the I/O pin to be an input or output.
PORTxx	The value in this register stores the output value of the I/O pin if DDRx is enabled (logic '1').
SETxx **(Pin Data/Set Register)**	A read of this address returns the current state of the I/O pin.

Table 2 – GPIO Control Ports

The addresses for the pertinent I/O control registers are shown below along with the bit values and connections. By using simple MOVE instructions we can deposit the necessary initialization values to appropriate registers are read the value of a register/pin inputs by simply 'moving' a value from its address.

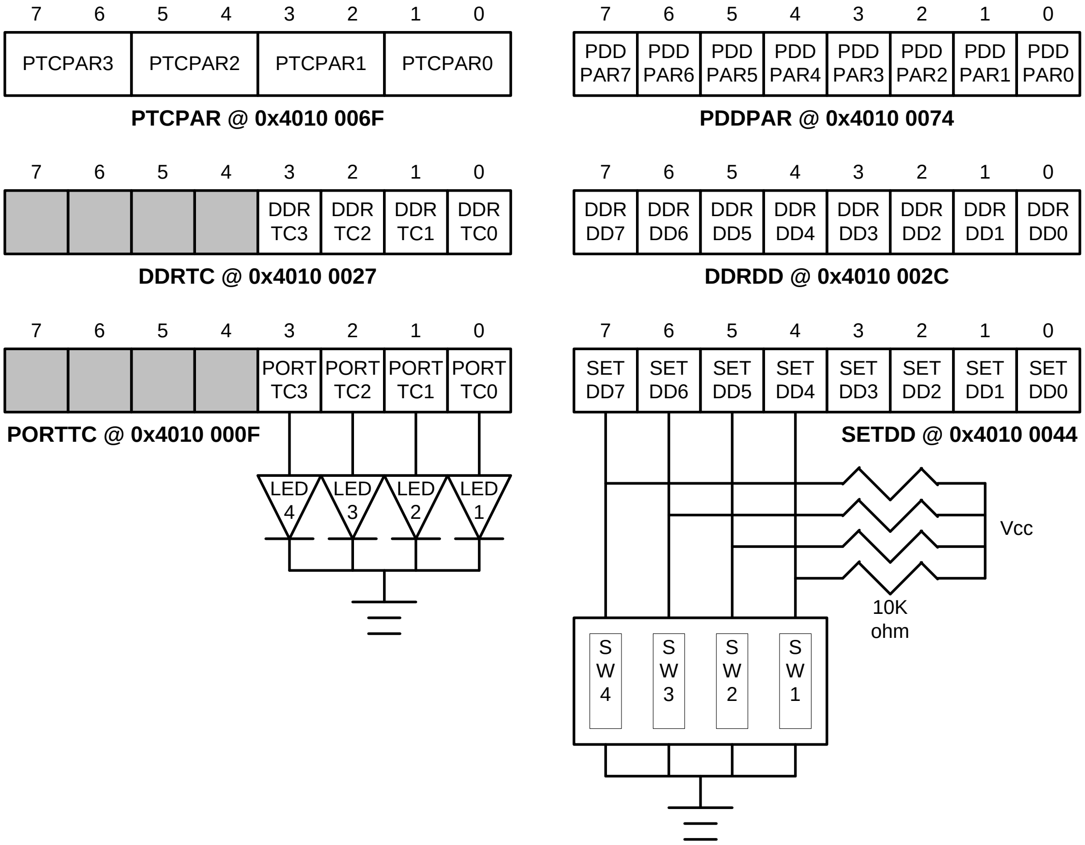

Figure 1 - I/O Control Registers and their addresses

Important: The LS 4-bits of PDDPAR and DDRDD are used for the debugger's operation and MUST REMAIN PROGRAMMED to 1's.

Blocks of Assembly in C: Most C compilers allow the programmer to embed assembly code sequences inside a C function by using an **`asm{`** … } block.

```
intmain(void)
{
      int counter = 0;

      asm {
            move.l #0,d0
            ...
      }
```