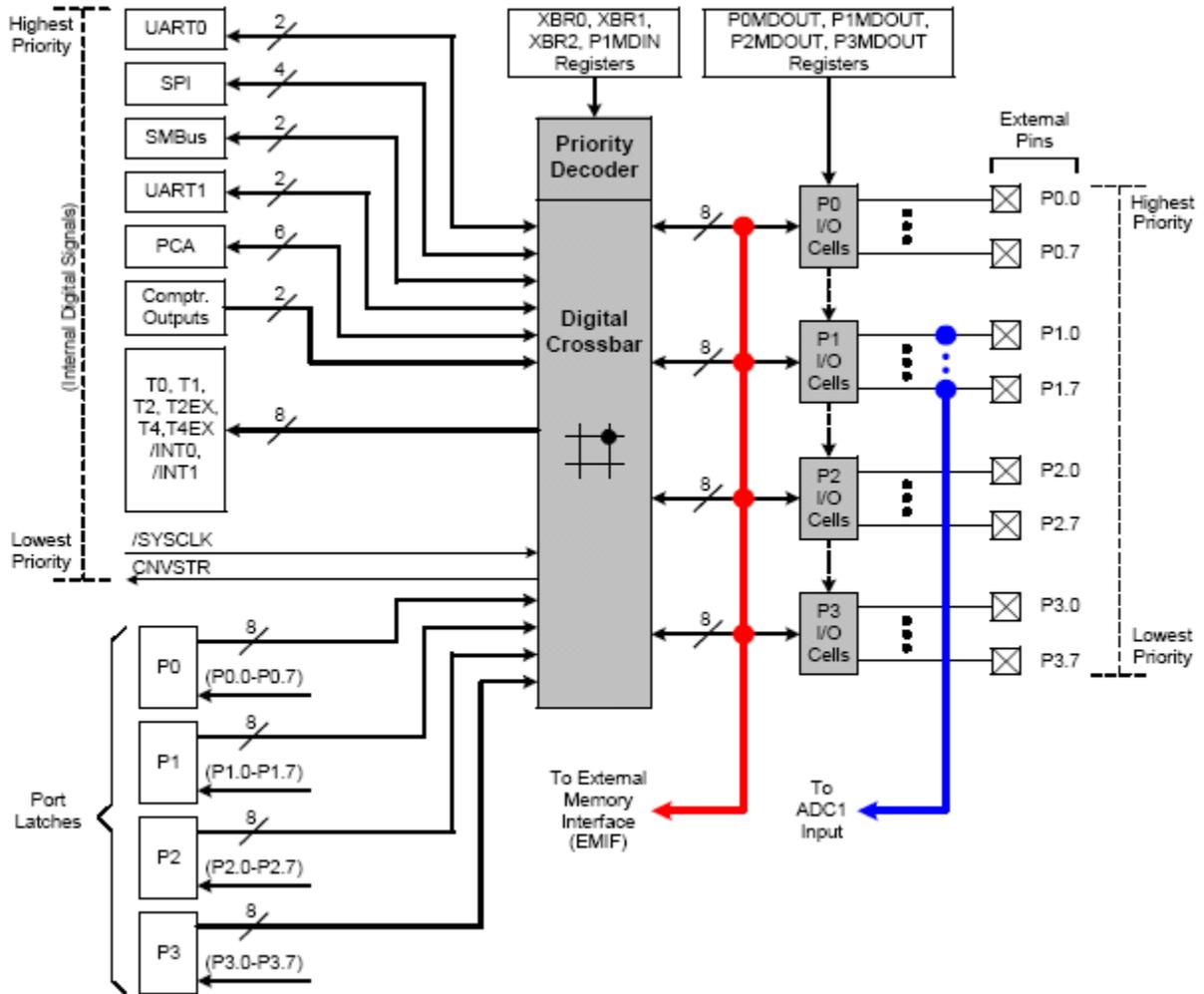


Input and Output Port Configuration

The C8051F020 has an "enhanced" set of I/O ports P0-P3. Below is the block diagram of this system:



Note the Digital Crossbar switch. This switch can be configured to connect the port latches to the I/O cells. It can also be configured to connect other systems, including UART1, SPI, SMBus, PCA, and other things, to the I/O cells.

Configuring Ports to be INPUTS

1. Set output mode to "Open-Drain"
2. Write a logic 1 to the associated bit in the Port Data register

Figure 17.11. P0MDOUT: Port0 Output Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xA4
<p>Bits7-0: P0MDOUT.[7:0]: Port0 Output Mode Bits. 0: Port Pin output mode is configured as Open-Drain. 1: Port Pin output mode is configured as Push-Pull.</p> <p>Note: SDA, SCL, and RX0 (when UART0 is in Mode 0) and RX1 (when UART1 is in Mode 0) are always configured as Open-Drain when they appear on Port pins.</p>								

To set all bits of Port 1 to be inputs:

```
mov P1MDOUT, #0 ; Set open-drain mode
mov P1, #0FFh ;
```

Configuring Ports to be OUTPUTS

1. Set the desired mode in the Port Output Mode Register.

```
push-pull: 0 = gnd, 1 = Vdd
open drain: 0 = gnd, 1 = high impedance (Z)
```

Enabling the Crossbar Switch

The crossbar switch must be enabled to use Ports 0-3. We do this by setting Bit6 of XBR2 high.

```
mov XBR2, #40h
```

Figure 17.9. XBR2: Port I/O Crossbar Register 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
WEAKPUD	XBARE	-	T4EXE	T4E	UART1E	EMIFLE	CNVSTE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE3
Bit7:	WEAKPUD: Weak Pull-Up Disable Bit. 0: Weak pull-ups globally enabled. 1: Weak pull-ups globally disabled.							
Bit6:	XBARE: Crossbar Enable Bit. 0: Crossbar disabled. All pins on Ports 0, 1, 2, and 3, are forced to Input mode. 1: Crossbar enabled.							
Bit5:	UNUSED. Read = 0, Write = don't care.							
Bit4:	T4EXE: T4EX Input Enable Bit. 0: T4EX unavailable at Port pin. 1: T4EX routed to Port pin.							
Bit3:	T4E: T4 Input Enable Bit. 0: T4 unavailable at Port pin. 1: T4 routed to Port pin.							
Bit2:	UART1E: UART1 I/O Enable Bit. 0: UART1 I/O unavailable at Port pins. 1: UART1 TX and RX routed to 2 Port pins.							
Bit1:	EMIFLE: External Memory Interface Low-Port Enable Bit. 0: P0.7, P0.6, and P0.5 functions are determined by the Crossbar or the Port latches. 1: If EMIOCF.4 = '0' (External Memory Interface is in Multiplexed mode) P0.7 (/WR), P0.6 (/RD), and P0.5 (ALE) are 'skipped' by the Crossbar and their output states are determined by the Port latches and the External Memory Interface. 1: If EMIOCF.4 = '1' (External Memory Interface is in Non-multiplexed mode) P0.7 (/WR) and P0.6 (/RD) are 'skipped' by the Crossbar and their output states are determined by the Port latches and the External Memory Interface.							
Bit0:	CNVSTE: External Convert Start Input Enable Bit. 0: CNVSTR unavailable at Port pin. 1: CNVSTR routed to Port pin.							

Exercise:

- 1) Write the instructions to set bit 6 of port 1 to be an output, and bit 7 of port 3 to be an input. Enable the crossbar switch as well. Set the mode of port 1 as push-pull.
- 2) Add bit move instructions to read bit 7 of port 3 and write it to bit 6 of port 1.

Insert these instructions into the following program shell.

```

#include (c8051f020.inc)      ; Include register definition
file.                        ;
                               ; Reset Vector
    cseg AT 0
    ljmp    Main             ; Locate a jump to the start

```

of code at

; the reset vector.

```
; Disable watchdog timer
```

```
Main:      mov WDTCN, #0DEh
```

```
           mov WDTCN, #0ADh
```

```
*****put your 1) code here *****
```

```
loop:
```

```
*****put your 2) code here *****
```

```
sjmp loop
```

```
END
```

Save the file as **ports.asm**. Assemble, build, download, and step through the program. Try executing the instruction that reads bit 7 of port 3 with the P3.7 switch pressed and unpressed. Do you read different values? Now run the program continuously and try pressing the pushbutton on the board. Do you understand the behavior?

Is the pushbutton active high or active low?

Is the LED active high or active low?

Change the mode of port 1 to open drain and note the behavior and explain it.