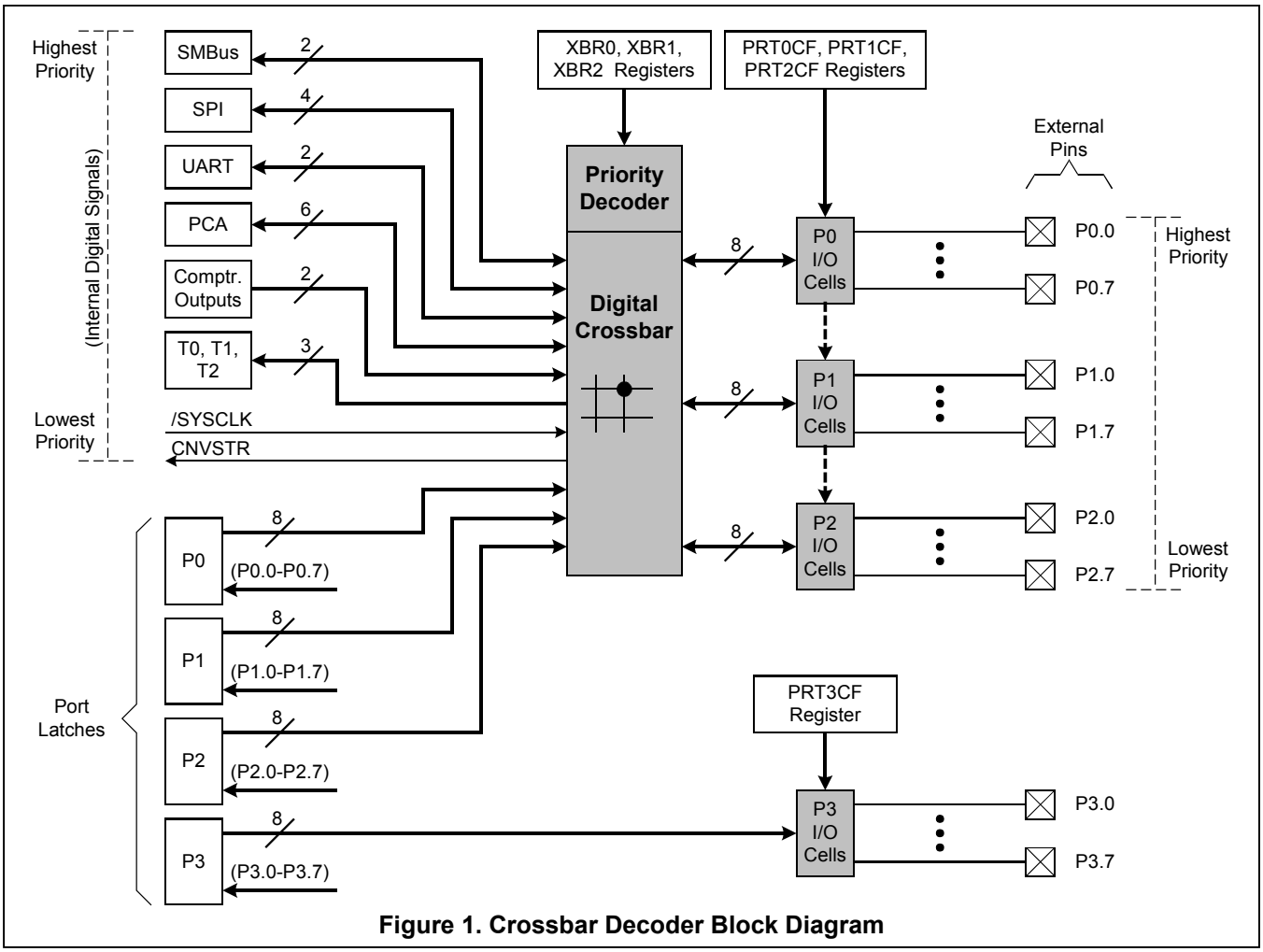

Figure 1. Crossbar Decoder Block Diagram

Introduction

The purpose of this application note is to describe how to configure and use the Port I/O Crossbar Decoder. Software and examples are provided.

The Crossbar, shown in Figure 1, is a multiplexer that maps the Port I/O pins to internal hardware peripherals on the device. For example, it determines to which port pins RXD and TXD of the UART get mapped.

The Crossbar governs the pin mapping of the SMBus, the SPI, the UART, the Timer Capture Modules, the External PCA Input, the comparator outputs, the Timer external inputs, /SYSCLK, and the A/D conversion start input. **The Crossbar must be configured and enabled before the I/O of any of these peripherals can be accessed**.

Unassigned port pins operate as normal GPIO.

The Crossbar provides two key system benefits:

- All unassigned GPIO port pins on Port 0, Port 1, and Port 2 are grouped contiguously.
- It provides flexibility in peripheral selection for reduced pin-count devices where some I/O ports may not be available. Peripheral selection is limited only by the **number** of port pins available, not by **which** port pins are available. This allows the system designer to choose which digital peripherals are available at the digital I/O pins on a pin-limited device.

Key Points

- The Crossbar MUST be enabled in order to use any Port 0, Port 1, or Port 2 port pins as outputs.
- The Crossbar registers should be configured before any of the digital peripherals are enabled.
- The Crossbar registers are typically configured once at reset, near the top of the Reset handler, and then left alone.
- The Crossbar settings change the pinout of the device.
- Each Crossbar setting results in a device pinout that is unique. If you enable or disable peripherals in the Crossbar, the pinout WILL change.
- The **output** mode of the port pins (open-drain or push-pull) must be explicitly set for the output port pins, even those assigned by the Crossbar. Exceptions to this are the SDA and SCL pins on the SMBus and the RX pin on the UART, which are automatically configured as open-drain.
- The open-drain or push-pull mode of Crossbar-assigned **input** pins (like NSS or /INT0, for example) is not important. These pins are configured as inputs regardless of the corresponding Port Configuration Register setting.
- To configure a GPIO pin as an input, the Port Configuration Register bit associated with that pin must be cleared, which selects that pin to have an open-drain output. Additionally, the Port bit associated with that pin must be set to a '1' which tri-states the pin or loosely pulls it high if WEAKPUD in XBR2 is set to '0'. This is the reset configuration of the port pins.
- The value at the port pins can be read at any time by reading the associated port SFR, regardless of the Crossbar register setting or whether the pin is configured as an input or an output.
- The Enable bits in the Crossbar registers are unique and distinct from the enable bits in the digital peripherals themselves.
    - Peripherals do not need to be enabled in the Crossbar in order to be used (for example, a PCA module can generate interrupts even if its output is not routed to a pin).
    - Peripherals that are enabled in the Crossbar, but disabled in their own SFRs still control the port pins. That is, the port pins can be read at any time, but the outputs are controlled exclusively by the owning peripheral and cannot be accessed as general-purpose outputs.
- The four external interrupts on Port 1 (P1.[4..7]) are triggered by a falling edge at the pin, regardless of the source of the falling edge, the Crossbar setting, or the output mode of the port pin.
- Unlike the standard 8051, true push-pull outputs are provided. If the 'strong-then-weak' pull-up function of the 8051 is required, it can be emulated in software by configuring the associated port output as 'push-pull' followed by a configuration to 'open-drain'.

Determining Device Pinout

This section describes how to use the Priority Crossbar Decode Table, Table 4, to determine the device pinout based on peripheral selection in the Crossbar registers, which are listed in Figures 2 through 4.

To determine the pinout, first configure the Crossbar registers based on the peripherals needed. Then starting at the top of the Priority Crossbar Decode table, scan down until you reach the first enabled device. This device will use P0.0, and if more pins are required, they will be assigned in order from P0.1 up. For example, if the SPI is the first peripheral enabled, then SCK MISO MOSI and NSS will be mapped to P0.0, P0.1, P0.2, and P0.3 respectively. The next enabled device will be assigned P0.4. All unassigned pins behave as GPIO.

Example 1

Assume that the application calls for:

- SPI
- UART
- 2 capture modules
- /INT0
- T2

Referencing the Port I/O Crossbar Register descriptions, which are listed in Figures 2 through 4 for convenience, the Crossbar registers are configured as follows:

```
XBR0 = 00010110b ; enable UART, 2 capture
                 ; modules, and SPI
XBR1 = 00100100b ; enable T2 and /INT0
XBR2 = 01000000b ; enable Crossbar
```

Starting from the top of Table 4, we find that the SPI pins will occupy P0.[0..3] (since the SMBus is not used), the UART pins P0.[4..5], CEX0 pin P0.6, CEX1 pin P0.7, /INT0 pin P1.0, and T2 pin P1.1. The other port pins P1.[2..7] and P2.[0..7] are available for GPIO. The pinout is listed in Table 1 below.

Example 2

Assume that the application calls for:

- UART
- /INT1
- /SYSCLK
- CNVSTR

The Crossbar registers are configured as follows:

```
XBR0 = 00000100b ; enable UART
XBR1 = 10010000b ; enable /INT1 and
                 ; /SYSCLK
XBR2 = 01000001b ; enable CNVSTR and
                 ; Crossbar
```

The pinout is listed in Table 2.

Example 3

Assume that the application calls for everything EXCEPT:

- Timer capture modules
- SPI port
- T0, T1, and T2 inputs
- Comparator outputs
- /SYSCLK

Which is another way of saying that we need the following:

- SMBus (I2C port)
- UART
- ECI

Table 1. Pinout for Example 1

Pin	Function
P0.0	SCK
P0.1	MISO
P0.2	MOSI
P0.3	NSS
P0.4	TX
P0.5	RX
P0.6	CEX0
P0.7	CEX1
P1.0	/INT0
P1.1	T2
P1.[2..7]	GPIO
P2.[0..7]	GPIO

Table 2. Pinout for Example 2

Pin	Function
P0.0	TX
P0.1	RX
P0.2	/INT1
P0.3	/SYSCLK
P0.4	CNVSTR
P0.[5..7]	GPIO
P1.[0..7]	GPIO
P2.[0..7]	GPIO

- /INT0
- /INT1
- T2EX
- CNVSTR

To enable the above, we configure the Crossbar registers as follows:

```
XBR0 = 01000101b ; enable ECI, UART, and
                 ; SMBus
XBR1 = 01010100b ; enable /INT0, /INT1, and
                 ; T2EX
XBR2 = 01000001b ; enable CNVSTR and
                 ; Crossbar
```

The pinout is listed in Table 3.

Table 3. Pinout for Example 3

Pin	Function
P0.0	SDA
P0.1	SCL
P0.2	TX
P0.3	RX
P0.4	ECI
P0.5	/INT0
P0.6	/INT1
P0.7	T2EX
P1.0	CNVSTR
P1.[1..7]	GPIO
P2.[0..7]	GPIO

Software Example

The following code segment illustrates how to configure the Crossbar registers:

```
;-----------------------------------------------------------------------------
;     CYGNAL INTEGRATED PRODUCTS, INC.
;
;
;     FILE NAME   : cross1.ASM
;     TARGET MCU  : C8051F000
;     DESCRIPTION : Example source code for configuring the crossbar.
;
; IMPLEMENTATION NOTES:
;
;
;-----------------------------------------------------------------------------

;-----------------------------------------------------------------------------
; EQUATES
;-----------------------------------------------------------------------------

$MOD8F000

;-----------------------------------------------------------------------------
; GLOBAL VARIABLES AND ASSIGNMENTS
;-----------------------------------------------------------------------------

; Crossbar port pin definitions (optional)

SCK         EQU   P0.0
MISO        EQU   P0.1
MOSI        EQU   P0.2
NSS         EQU   P0.3
TX          EQU   P0.4
RX          EQU   P0.5
CEX0        EQU   P0.6
CEX1        EQU   P0.7
INT0        EQU   P1.0
T2          EQU   P1.1

; other variables and assignments

;-----------------------------------------------------------------------------
; INTERRUPT VECTOR TABLE
;-----------------------------------------------------------------------------
; Reset Vector
            org   00h
            ljmp  Reset

; other interrupt vectors here…

;-----------------------------------------------------------------------------
; INTERRUPT VECTOR CODE
```

```
;-------------------------------------------------------------------------
            org   0B3h

Reset:
            mov   XBR0, 00010110b         ; enable SPI, UART, and 2 external
                                          ;  captures
            mov   XBR1, 00100100b         ; enable /INT0 and T2
            mov   XBR2, 01000000b         ; enable Crossbar and weak pull-ups
                                          ; Port pinout is as follows:
                                          ; P0.0 = SCK - example: master mode
                                          ; P0.1 = MISO
                                          ; P0.2 = MOSI
                                          ; P0.3 = NSS
                                          ; P0.4 = TX
                                          ; P0.5 = RX
                                          ; P0.6 = CEX0 - example: output
                                          ; P0.7 = CEX1 - example: input
                                          ; P1.0 = /INT0
                                          ; P1.1 = T2
                                          ; P1.[2..7], P2.[0..7] are GPIO
            mov   PRT0CF, #01010101b      ; CEX0, TX, MOSI, and SCK are push-pull
                                          ;  outputs
            mov   PRT1CF, #00000000b      ; T2 and /INT0 are inputs assigned by the
                                          ;  Crossbar, so their Port Configuration
                                          ;  value is irrelevant
            mov   PRT2CF, #00000000b      ; configuring all of Port 2 as inputs
            mov   P2,     #11111111b      ; To configure unassigned GPIO pins as
                                          ;  inputs, the Output type must be Open-
                                          ;  Drain, and the Port bits MUST be set
                                          ;  to '1'.

            ; other configuration code here

            ljmp  Main

;-------------------------------------------------------------------------
; MAIN PROGRAM CODE
;-------------------------------------------------------------------------

Main:
            sjmp  $                       ; spin here forever
            ; main code routines here…
;-------------------------------------------------------------------------
; END
;-------------------------------------------------------------------------

END
```

Table 4. Priority Crossbar Decode Table

	P0								P1								P2							
PIN I/O	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
SDA	●																							
SCL		●																						
SCK	●		●																					
MISO		●		●																				
MOSI			●		●																			
NSS				●		●																		
TX	●		●		●		●																	
RX		●		●		●		●																
CEX0	●		●		●		●		●															
CEX1		●		●		●		●		●														
CEX2			●		●		●		●		●													
CEX3				●		●		●		●		●												
CEX4					●		●		●		●		●											
ECI	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
CP0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									
CP1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
T0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							
/INT0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						
T1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					
/INT1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				
T2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			
T2EX	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		
/SYSCLK	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
CNVSTR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

In the Priority Decode Table, a dot (•) is used to show the external Port I/O pin (column) to which each signal (row) can be assigned by the user application code.

Figure 2. XBR0: Port I/O Crossbar Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0OEN	ECIE	PCA0ME			UARTEN	SPI0OEN	SMB0OEN	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE1

Bit7: CP0OEN: Comparator 0 Output Enable Bit
0: CP0 unavailable at Port pin.
1: CP0 routed to Port Pin.

Bit6: ECIE: PCA0 Counter Input Enable Bit
0: ECI unavailable at Port pin.
1: ECI routed to Port Pin.

Bits5-3: PCA0ME: PCA Module I/O Enable Bits
000: All PCA I/O unavailable at Port pins.
001: CEX0 routed to Port Pin.
010: CEX0, CEX1 routed to 2 Port Pins.
011: CEX0, CEX1, CEX2 routed to 3 Port Pins.
100: CEX0, CEX1, CEX2, CEX3 routed to 4 Port Pins.
101: CEX0, CEX1, CEX2, CEX3, CEX4 routed to 5 Port Pins.
110: RESERVED
111: RESERVED

Bit2: UARTEN: UART I/O Enable Bit
0: UART I/O unavailable at Port pins.
1: RX, TX routed to 2 Port Pins.

Bit1: SPI0OEN: SPI Bus I/O Enable Bit
0: SPI I/O unavailable at Port pins.
1: MISO, MOSI, SCK, and NSS routed to 4 Port Pins.

Bit0: SMB0OEN: SMBus Bus I/O Enable Bit
0: SMBus I/O unavailable at P0.0, P0.1.
1: SDA routed to P0.0, SCL routed to P0.1.

Figure 3. XBR1: Port I/O Crossbar Register 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SYSCKE	T2EXE	T2E	INT1E	T1E	INT0E	T0E	CP1OE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE2

Bit7: SYSCKE: SYSCLK Output Enable Bit
0: SYSCLK unavailable at Port pin.
1: SYSCLK output routed to Port Pin.

Bit6: T2EXE: T2EX Enable Bit
0: T2EX unavailable at Port pin.
1: T2EX routed to Port Pin.

Bit5: T2E: T2 Enable Bit
0: T2 unavailable at Port pin.
1: T2 routed to Port Pin.

Bit4: INT1E: /INT1 Enable Bit
0: /INT1 unavailable at Port pin.
1: /INT1 routed to Port Pin.

Bit3: T1E: T1 Enable Bit
0: T1 unavailable at Port pin.
1: T1 routed to Port Pin.

Bit2: INT0E: /INT0 Enable Bit
0: /INT0 unavailable at Port pin.
1: /INT0 routed to Port Pin.

Bit1: T0E: T0 Enable Bit
0: T0 unavailable at Port pin.
1: T0 routed to Port Pin.

Bit0: CP1OE: Comparator 1 Output Enable Bit
0: CP1 unavailable at Port pin.
1: CP1 routed to Port Pin.

Figure 4. 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	-	-	-	-	-	CNVSTE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE3

Bit7: WEAKPUD: Port I/O Weak Pull-up Disable Bit
0: Weak Pull-ups Enabled
1: Weak Pull-ups Disabled
Bit6: XBARE: Crossbar Enable Bit
0: Crossbar Disabled
1: Crossbar Enabled
Bits5-1: UNUSED. Read = 00000b, Write = don't care.
Bit0: CNVSTE: ADC Convert Start Input Enable Bit
0: CNVSTR unavailable at Port pin.
1: CNVSTR routed to Port Pin.

Example Usage of XBR0, XBR1, XBR2:
When selected, the digital resources fill the Port I/O pins in order (top to bottom as shown in the Priority Crossbar Decode Table) starting with P0.0 through P0.7, and then P1.0 through P1.7, and finally P2.0 through P2.7. If the digital resources are not mapped to the Port I/O pins, they default to their matching internal Port Register bits.

Example1: If XBR0 = 0x11, XBR1 = 0x00, and XBR2 = 0x40:
P0.0=SDA, P0.1=SCL, P0.2=CEX0, P0.3=CEX1, P0.4 … P2.7 map to corresponding Port I/O.

Example2: If XBR0 = 0x80, XBR1 = 0x04, and XBR2 = 0x41:
P0.0=CP0, P0.1=/INT0, P0.2 = CNVSTR, P0.3 … P2.7 map to corresponding Port I/O.