

# Initializing the MSC8101 Communications Processor Module (CPM) I/O and Using Pinmux8101

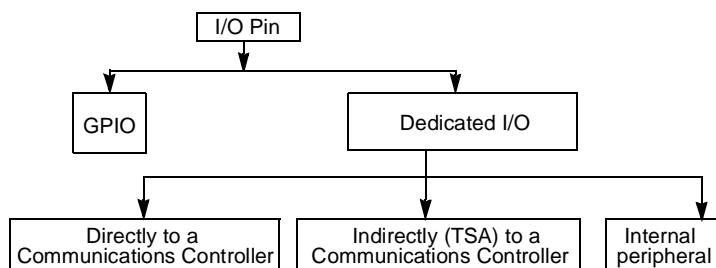
By Donald Simon

The MSC8101 supports four GPIO ports A–D in the CPM. Each port pin can be configured as GPIO or as a dedicated peripheral interface (dedicated I/O). A dedicated I/O configuration connects the I/O pin directly to a communications controller such as a fast communications controller (FCC) or serial communications controller (SCC), indirectly through the time-slot assigner (TSA), or to an internal CPM peripheral such as a baud-rate generator (BRG) or timer (see **Figure 1**). For a dedicated I/O configuration, further programming is required—for example, setting up a communications controller in the form of parameter RAM or buffer descriptors, TSA, or an internal peripheral. Information on this additional programming is beyond the scope of this document. For details, consult the *MSC8101 Reference Manual*.

Before you program the I/O pins, you must understand the mechanics of the CPM ports, so this application note touches upon the CPM Communications Processor (CP), TDM/NMSI interfaces, communications controllers and channels/interfaces/protocols. It describes how to use a very powerful tool called *Pin\_mux8101* that generates parallel I/O (PIO) initialization code to initialize the I/O pins as GPIO and/or dedicated I/O.

## CONTENTS

<b>1</b>	CPM Overview .....	2
<b>1.1</b>	TDM and NMSI Interfaces .....	2
<b>1.2</b>	Communications Controllers .....	4
<b>1.3</b>	Channels, Interfaces and Protocols .....	4
<b>2</b>	GPIO Versus Dedicated I/O.....	8
<b>3</b>	Steps in Assigning I/O Pins .....	11
<b>3.1</b>	Configuring I/O Pins .....	11
<b>3.2</b>	Clocking .....	13
<b>3.3</b>	Serial Channels by Function .....	15
<b>4</b>	Pin_mux8101 Application .....	22
<b>4.1</b>	Strengths and Weaknesses .....	23
<b>4.2</b>	Menus .....	24
<b>4.3</b>	Procedure .....	24
<b>4.4</b>	Application Example .....	25
<b>4.5</b>	Handling Duplicate I/O .....	26
<b>4.6</b>	Reporting .....	26
<b>5</b>	References .....	33



**Figure 1.** I/O Pin Types

# 1 CPM Overview

The CPM is one of three internal modules in the MSC8101: CPM, system interface unit (SIU), and the SC140 extended core. The CPM consists of much more than PIO ports (see **Figure 2**). The brain of the CPM is the communications processor (CP), a 32-bit RISC microcontroller that resides on a bus separate from the SC140 core and performs tasks independent of the SC140 core. The CP handles lower-layer communications tasks and DMA control, freeing the SC140 core to handle higher-layer activities. The CP works with the peripheral controllers, Serial DMA (SDMA) channels, timers, BRGs, and PIO ports.

**Note:** The MSC8101 device is the same as the MSC8103, except that the MSC8101 includes an enhanced filter coprocessor (EFCOP).

## 1.1 TDM and NMSI Interfaces

In general, the PIO ports are divided between two interface classifications (see **Figure 3**):

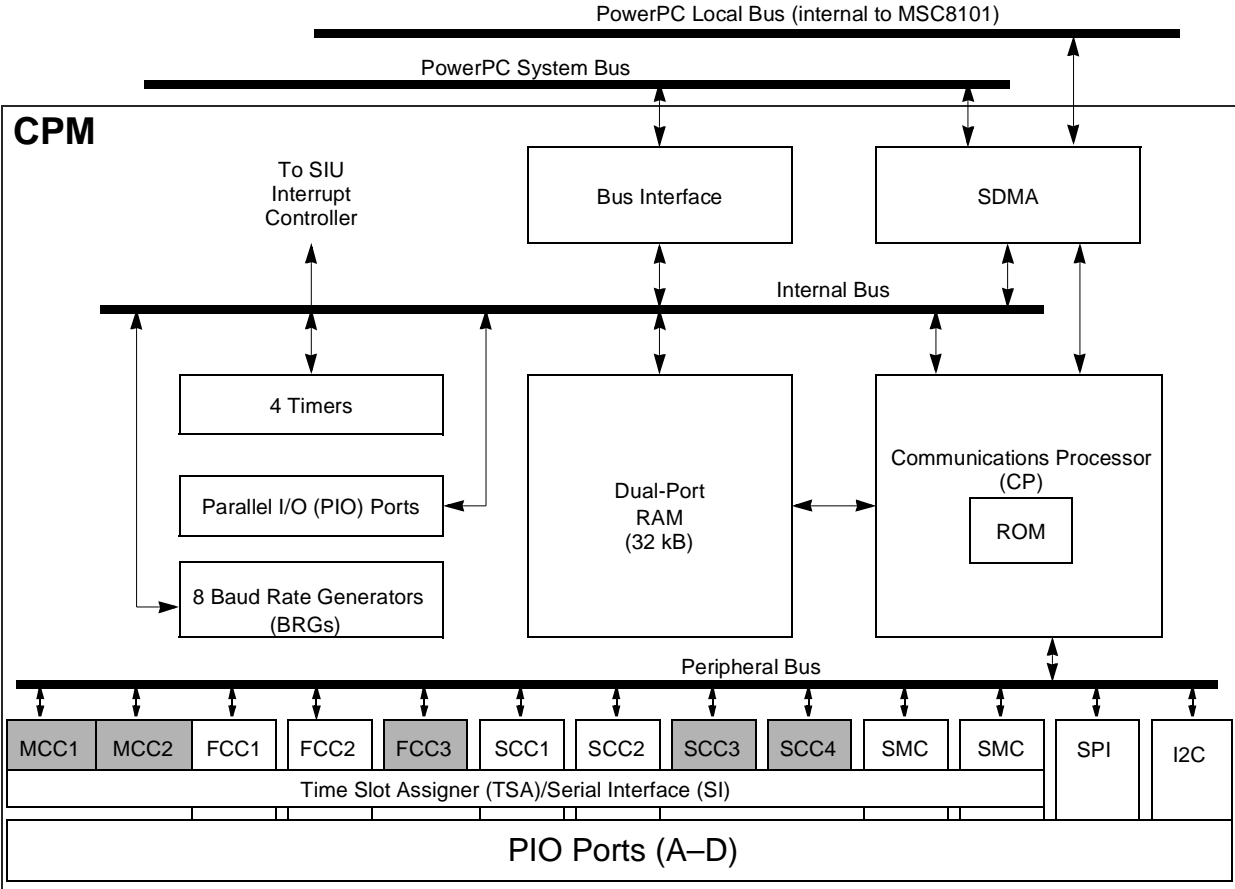
- TDM: time-division multiplexing
- NMSI: non-multiplexed serial interface

The TDM interfaces to the TSA, a functional block within the CPM that connects the TDM interfaces to selected communication controllers inside the MSC8101. The MSC8101 supports four TDM interfaces: TDMA1, TDMB2, TDMC2, and TDMD2.

The TSA and TDM interfaces are part of the serial interface within the CPM. There are actually two Serial Interfaces (SI1 and SI2). The “1” and “2” at the end of the TDM names indicate which Serial Interface (SI) they belong to: SI1 or SI2. The MSC8101 has one TDM from SI1 (TDMA1) and three from SI2 (TDMB2, TDMC2, and TDMD2). The TDM interfaces derive from the MPC8260, which supports eight TDM interfaces. The MSC8101 TDM interfaces are split up between SI1 and SI2 to allow for higher system performance. TDMA1 is special because it supports both bit and nibble data modes. TDM applications include: T1, E1, PCM, and ISDN. The other interface, Non-Multiplexed Serial Interface (NMSI), is a legacy term carried over to the MSC8101. *Non-multiplexed* refers to the fact that data is not multiplexed from communications controllers over time, as it is in the TDM interfaces.

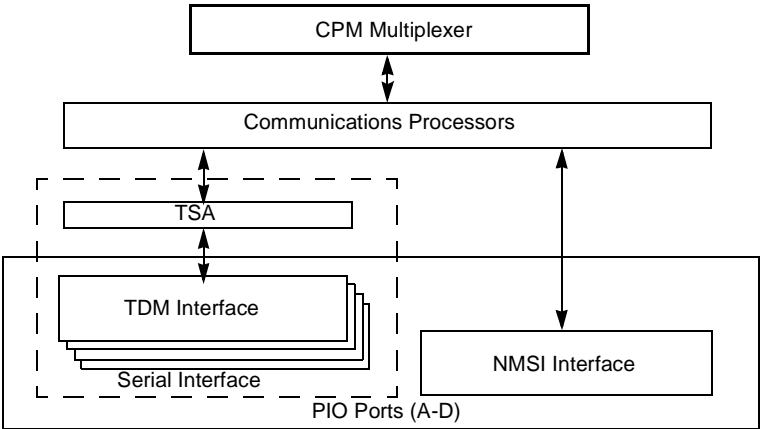
### Acronyms and Abbreviations

ANSI	American National Standards Institute
ATM	Asynchronous Transfer Mode
BRG	Baud Rate Generator
CP	Communications Processor
CPM	Communications Processor Module
$\overline{\text{DACK}}$	DMA Data Acknowledge Signal
DMA	Direct Memory Access
E1	European equivalent of North American T1 except that E1 carries information at the rate of 2.048 Mbps
FCC	Fast Communications Controller
GPIO	General-Purpose Input/Output
GSM	Groupe Speciale Mobile, now known as Global System for Mobile Communications
HDLC	High-level Data Link Control
I <sup>2</sup> C	Inter-Integrated Circuit
I/O	Input/Output
IP	Internet Protocol
ISDN	Integrated Services Digital Network
MCC	Multi-Channel Controller
MII	Media Independent Interface
NMSI	Non-Multiplexed Serial Interface
PCM	Pulse Code Modulation
PDAT	PIO Data register
PDIR	PIO Direction register
PIO	Parallel I/O
PODR	PIO Open-Drain Register
PPAR	PIO Port Pin Assignment Register
PSOR	PIO Special Options Register
RxD/TxD	Receive/Transmit Data
SCC	Serial Communications Controller
SDMA	Serial DMA
SI	Serial Interface
SIU	System Interface Unit
SMC	Serial Management Controller
SPI	Serial Peripheral Interface
T1	Digital transmission link with a capacity of 1.544 Mbps
TDM	Time-Division Multiplexing
TSA	Time-Slot Assigner
UART	Universal Asynchronous Receiver Transmitter
UTOPIA	Universal Test and Operation Interface for ATM



- Notes:**
1. The shaded areas are available only through one of four common TDM interfaces. The TDM buses are part of the PIO ports.
  2. There are 13 communications controllers: two Multi-Channel Communications Controllers (MCC1 and MCC2), three Fast Communications Controllers (FCC1–FCC3), four Serial Communications Controllers (SCC1–SCC4), two Serial Management Controllers (SMCs), a Serial Peripheral Interface (SPI), and an Inter-Integrated Circuit (I<sup>2</sup>C).

**Figure 2.** CPM Functional Diagram



**Figure 3.** TDM and NMSI Interfaces (General)

## 1.2 Communications Controllers

A communications controller handles the operation and protocol(s) between the physical interface and the communications processor (CP). As **Figure 2** shows, the MSC8101 supports thirteen communications controllers (MCC1–2, FCC1–3, SCC1–4, SMC1–2, SPI, and I<sup>2</sup>C). All communications controllers have dedicated external pins except MCC1–2, FCC3, and SCC3–4 (available only through the TSA/TDM interfaces). All communications controllers except SPI and I<sup>2</sup>C have access to the TDM interface. A communications controller can support more than one interface. For example, FCC1 supports UTOPIA 8 (ATM), MII (Ethernet and Fast Ethernet), HDLC (serial and nibble modes), and transparent interfaces. The SPI and I<sup>2</sup>C interfaces have the same name as the SPI and I<sup>2</sup>C communications controller.

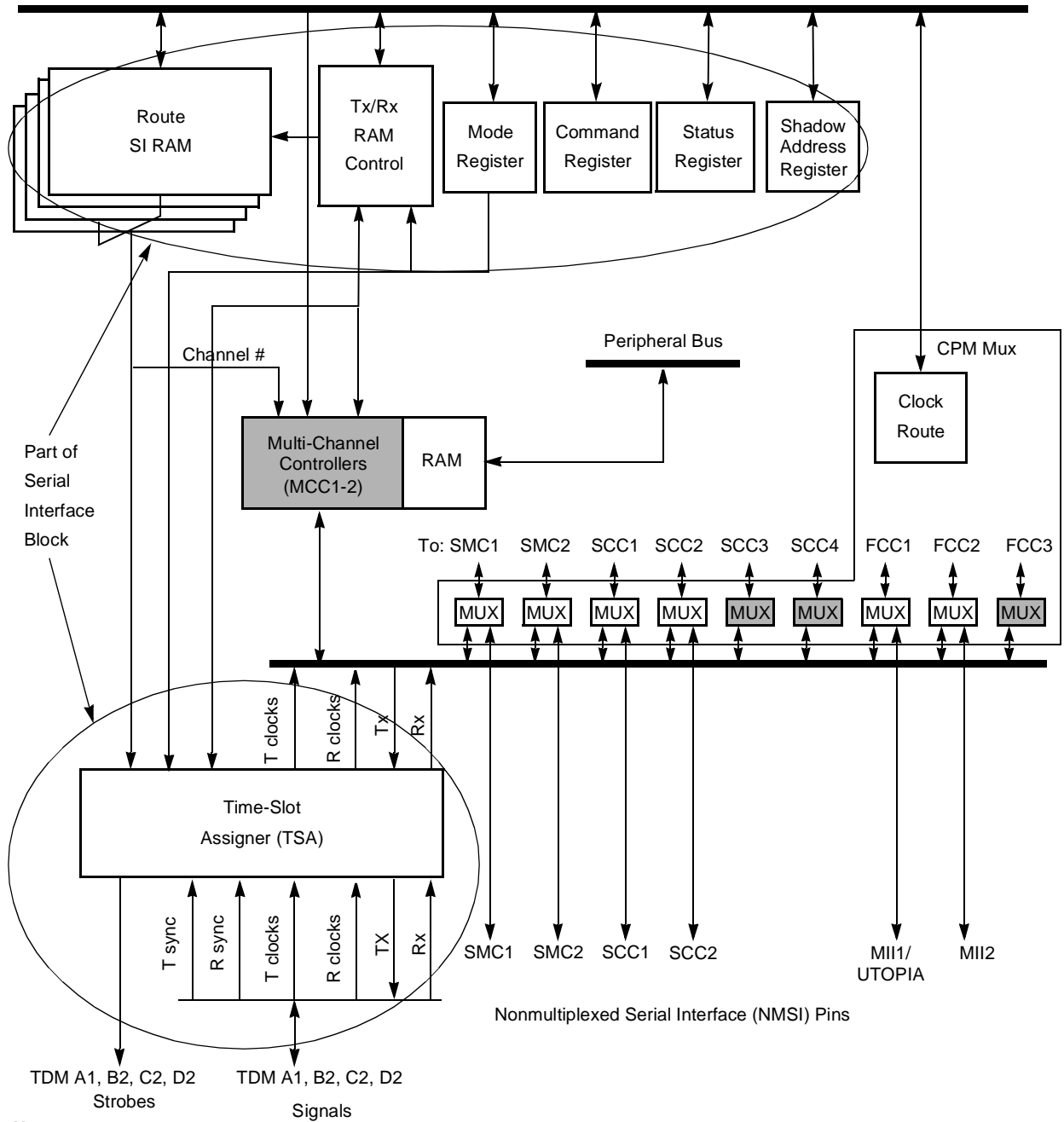
**Figure 4** shows a more detailed diagram of the TDM and NMSI interfaces. Each SI comprises one or more TDM interfaces, TSA, SI RAM, and several registers. The CPM multiplexer and MCCs are not part of the SIs.

## 1.3 Channels, Interfaces and Protocols

This section defines three complementary terms to show what each communications controller supports:

- *Channel*. A transmission path between two points independent of the interface used—for example, FCC1 supports Ethernet and ATM, among others. See **Table 4** and **Table 9**.
- *Interface*. A mechanical or electrical link connecting two or more pieces of equipment. All the interfaces discussed here refer to the electrical link, that is, to the signals required to support a given channel. FCC1 supports MII and UTOPIA interfaces, among others. See **Table 2**.
- *Protocol*. A specific set of rules, procedures, or conventions pertaining to format and timing of data transmission between two devices. Data communication protocols may include framing, error handling, and line control. There are three types of protocols: character-oriented, byte-oriented, and bit-oriented. FCC1 supports ATM and Ethernet protocols, among others. See **Table 4**.

Names that fall under one of these categories are not unique, so a similar name may also fall under another category. For example, ATM can be considered both a channel and a protocol.



**Notes:**

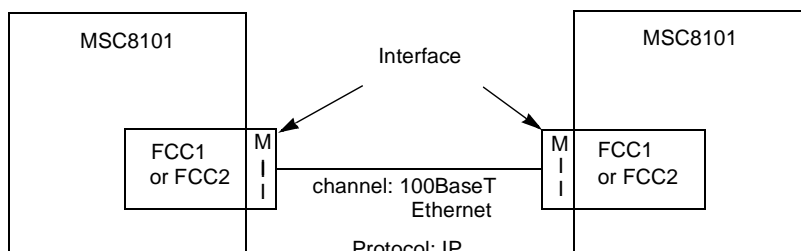
1. The shaded areas are available only through the four common TDM ports.
2. Any TDM strobe can be used on any TDM port for the associated SI.
3. The CPM multiplexer and the MCCs are not part of the SI.

**Figure 4.** TDM and NMSI Interfaces (Detailed)

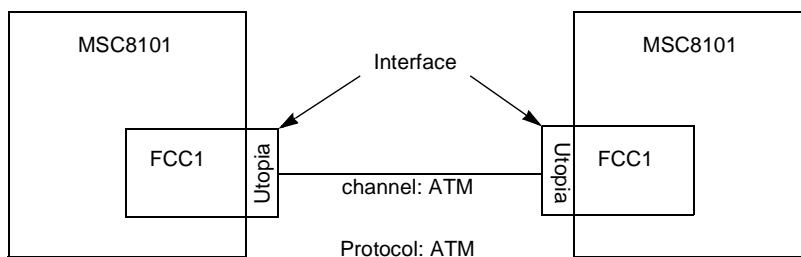
**Table 1** shows two examples defining the communications controller, channel, interface, and protocol for 100BaseT Ethernet and ATM. **Figure 5** and **Figure 6** depict these examples graphically.

**Table 1.** Channel, Interface, and Protocol Example

Examples	Communications Controllers	Channel	Interface	Prot
100BaseT Ethernet	FCC1, FCC2	100BaseT Ethernet	MII	IP
ATM	FCC1	ATM	UTOPIA	ATM



**Figure 5.** 100BaseT Ethernet Example



**Figure 6.** ATM Example

### 1.3.1 Interfaces

**Table 2** shows supported interface(s) for each communications controller. **Table 3** defines these interfaces.

**Table 2.** MSC8101 Interfaces

Communication Controllers	Interfaces Supported
FCC1	UTOPIA 8; MII (Ethernet:100BaseT/10BaseT); TDM
FCC2	MII (Ethernet: 100BaseT/10BaseT); TDM
FCC3	TDM
MCC1–2	TDM
SCC1–2	Ethernet (10BaseT); RS-232 with and without modem controls; TDM
SCC3–4	TDM
SMC1–2	RS-232; TDM
SPI	SPI
I <sup>2</sup> C	I <sup>2</sup> C

**Table 3.** MSC8101 CPM Interface Definitions

Interface	Description
I <sup>2</sup> C	<b>Inter-integrated circuit.</b> A Simple low-cost mechanism for multiple devices. I <sup>2</sup> C is more flexible than SPI for multi-masters in handling collisions. It features a synchronous 2-wire interface (clock and data); bidirectional operation; and support for master/slave, multi-master modes. Clock rates run up to 520 kHz@25 MHz system clock. Developed by Philips.
MII	<b>Media Independent Interface.</b> Part of the Fast Ethernet Specification, the MII replaces 10BaseT AUI (Attachment Unit Interface) and connects the MAC layer to the physical layer. The MII is the standard for all three 100Base-T specifications: 100Base-TX, 100Base-T4, 100Base-Fx). The MII interface can be used for both 100Base-T and 10Base-T. MSC8101 MII support includes four bits of data for transmit and four bits of data for receive. If data is transmitted/received at 100 MBPS, the clock rate must be at least 25 MHz (100 MBPS / 4 bits).
RS-232 with modem controls	An ANSI standard specifying three interfaces: electrical, functional, and mechanical. The RS-232 standard is typically used to communicate between computers, terminals and modems. All PCs support RS-232, usually through a DB9 connector as specified by ANSI. The MSC8101 supports TXD, RXD, RTS, CTS, and CD signals.
RS-232 with no modem controls	The MSC8101 SMC1–2 interface supports TXD, RXD, and SMSYN signals. The SMC interface does not support modem control signals, and the data rates are not as fast as the SCC interface.
SPI	<b>Serial Peripheral Interface.</b> A full duplex, synchronous, character-oriented channel that supports a 4-wire interface (receive, transmit, clock and slave select). SPI supports master/slave and multi-master modes. Character size is programmable.
TDM	<b>Time-Division Multiplexed.</b> External interface to the MSC8101 and the MSC8101 time-slot assigner (TSA) block.
UTOPIA 8	<b>Universal Test and Operations Interface for ATM.</b> UTOPIA is the interface to an ATM network. UTOPIA is defined by the ATM Forum in the <i>UTOPIA Specification Level 1</i> and <i>UTOPIA Specification Level 2</i> documents. <i>UTOPIA Specification Level 2</i> is a continuation of the <i>UTOPIA Specification Level 1</i> document. The MSC8101 is level 1- and level 2-compliant.

### 1.3.2 Protocols

**Table 4** summarizes the protocols available for each communications controller.

**Table 4.** MSC8101 Communications Controllers Versus Protocols

Protocol	Communications Controllers								
	FCC1	FCC2	FCC3	SCC1–2	SCC3–4	MCC1–2	SMC1–2	SPI	I <sup>2</sup> C
ATM Utopia 8:									
• Multi-PHY Master multiplexed polling	+								
• Multi-PHY Master direct polling	+								
• Slave	+								
Fast Ethernet (100BaseT)	+	+							
Ethernet (10BaseT)	+	+		+					
HDLC nibble/serial	+	+		+	+	+			
Transparent	+	+		+		+	+		
UART				+			+		
BISYNC				+	+				
SPI								+	
I <sup>2</sup> C									+

### 1.3.3 MSC8101 Applications

Table 5 shows some possible MSC8101 configurations.

Table 5. MSC8101 Applications

Application	MCC1	MCC2	FCC1	FCC2	SCC1	SCC2	SMC1	SMC2
ISDN router	E1s	E1s	FEnet or ATM	FEnet	UART	UART		
ATM switch			ATM	FEnet	UART			
ATM access	E3 or E1s	E3 or E1s	ATM	FEnet	UART			
GSM mobile switching center	E1s		FEnet or ATM Backbone	10 M HDLC				

## 2 GPIO Versus Dedicated I/O

As Table 6 shows, the MSC8101 supports 66 I/O pins organized within ports A–D.

Table 6. MSC8101 CPM I/O Pins

Port	Number of MSC8101 CPM I/O Pins
Port A	26
Port B	14
Port C	18
Port D	8
<b>Total PIO Pins</b>	<b>66</b>

These ports are the external interface to the CPM. As depicted in Figure 1, each I/O pin can function as either a GPIO or as a dedicated peripheral pin. In addition, each I/O pin can be configured as either an input, output, or open-drain output. An open-drain configuration drives a pin low for a low and to high impedance (tri-state) for a high. Figure 7 shows how the CPM pins are configured. For a detailed description of each port pin and its multiplexed functions, refer to the *External Signals* chapter in the *MSC8101 Reference Manual* or the *MSC8101 Technical Data* sheet.

FCC1		ATM/UTOPIA		FCC1		GPIO
MPHY Master mux poll or Slave	MPHY Master dir. poll	Ether net MII	HDL C/trans p.	HDL C	Serial Nibble	
TXENB		COL				PA31
TXCLA V	TXCLA V0	CRS	RTS			PA30
TXSOC		TX_E R				PA29
RXENB		TX_E N				PA28

Figure 7. CPM Port A–D Pin Multiplexed Functionality

RXSOC		RX_D											PA2
RXCL	RXCLA	V											7
AV	V0	R											PA2
TXD0													6
TXD1													PA2
TXD2													5
TXD3													PA2
TXD4		TXD3	TXD									4	
TXD5		TXD2	3									PA2	
TXD6		TXD1	TXD									3	
TXD7		TXD0	2									PA2	
RXD7		RXD0	TXD									2	
RXD6		RXD1	TXD									PA1	
RXD5		RXD2	0									9	
RXD4		RXD3	RXD									PA1	
RXD3			0									8	
RXD2			0									PA1	
RXD1			1									7	
RXD0			2									PA1	
			3									6	
												5	
												PA1	
												4	
												PA1	
												3	
												PA1	
												2	
												PA1	
												1	
												PA1	
												0	
												PA1	
												9	
												PA8	
												8	
												PA7	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	
												5	
												PA6	
												4	
												PA6	
												3	
												PA6	
												2	
												PA6	
												1	
												PA6	
												0	
												PA6	
												9	
												PA6	
												8	
												PA6	
												7	
												PA6	
												6	
												PA6	



		CD		SMR XD	LIST4			PC4
RXAD DR3	RXCLA V2		RXD			DRACK1/D ONE1		PD3 1
TXAD DR4	TXCLA V3		TXD			DRACK2/D ONE2		PD3 0
RXAD DR4	RXCLA V3		RTS/TE NA			SPI		PD2 9
	RXPRTY					SPISEL	BRG 10	PD1 9
	TXPRTY					SPICLK		PD1 8
						SPI MOSI	BRG 20	PD1 7
TXAD DR3	TXCLA V2			SMS YN		SPI MISO		PD1 6
								PD7

Figure 7. CPM Port A–D Pin Multiplexed Functionality

### 3 Steps in Assigning I/O Pins

**Table 7** lists general steps in assigning I/O pins. You do not necessarily have to perform the steps in the order listed in the table. Typically, selecting pins for an application is an iterative process.

**Table 7.** Steps in Assigning PIO pins

Step	Procedure	Comments
1	Start with your system requirements.	Make a check list of all your I/O requirements.
2	As much as possible, start from an existing design.	Refer to the <i>MSC8101 Reference Manual</i> and web sites listed in <b>Section 5</b> .
3	Determine your performance needs.	The tool is called the <i>MPC8260 CPM Performance Evaluator</i> . You can obtain this tool in a .zip file as a companion to this application note on the Freescale web site listed on the back cover of this document. The name of the file is AN1854SW.zip. You must decompress the software in the .zip file before you can use it. The MSC8101 CPM is similar to the MPC8260 CPM, except that not all the MPC8260 pins are available externally to the device.
4	Select your I/O pin configuration.	This step can be accomplished manually by following the procedures in <b>Section 3.1</b> or by an automated tool called <i>Pin_mux8101</i> described in <b>Section 4</b> , <i>Pin_mux8101 Application</i> , on page 22. <i>Pin_mux8101</i> groups the pins according to functionality, keeping track of which pins are available. <i>Pin_mux8101</i> can generate code to initialize the PIO ports.

#### 3.1 Configuring I/O Pins

In general, each CPM pin can be configured as a GPIO or dedicated I/O. By default, all CPM I/O pins are inputs. The I/O pins do not have internal pull-up or pull-down resistors. Unused I/O pins can be configured as outputs after reset and left unconnected. However, unused I/O pins configured as inputs should be tied high or low and not left unconnected. Some dedicated I/O signals can serve as an output on two different pins. This gives you a choice as to which pin can be selected when another function is using the first pin. **Table 8** shows which pins are duplicated. You can freely configure such functions to be output on two pins at once. However, there is typically no advantage in doing so unless there is a large fanout that makes it advantageous to share the load between the two pins.

**Table 8.** Duplicate I/O Pins

Duplicate I/O Pin	I/O Pins
SMC2: SMTXD	PA9, PC15
SCC2: $\overline{\text{CTS}}/\text{CLSN}$	PC28, PC12
SCC1: $\overline{\text{CTS}}/\text{CLSN}$	PC29, PC15
BRG1O	PC31, PD19
BRD2O	PC30, PD17

As **Table 9** shows, each PIO port has five memory-mapped, read/write, 32-bit control registers. Since each PIO port has less than 32 input/output pins, not all bits in each of the 32-bit control registers are used. In general, bit 0 for each register corresponds to Port (A/B/C/D) 0, or P<sub>x</sub>0, where *x* is A, B, C, or D. Refer to the chapter on parallel ports in the *MSC8101 Reference Manual* for bit assignments and descriptions.

**Table 9.** PIO Port Register Names

Register	Name	Description
PPAR <sub>A/B/C/D</sub>	Port Pin Assignment Registers	Set up a pin as a GPIO or a dedicated peripheral pin.
PSOR <sub>A/B/C/D</sub>	Port Special Options Registers	Set up the peripheral function as either dedicated 1 or 2 if the pin is configured for a dedicated peripheral pin (PPAR <sub>x</sub> ).
PDIR <sub>A/B/C/D</sub>	Port Data Direction Registers	Set up a pin as an input or output.
PODR <sub>A/B/C/D</sub>	Port Open-Drain Registers	Set up a pin as an output or open-drain driver.
PDATA <sub>A/B/C/D</sub>	Port Data Registers	Return the data at the pin.

**Table 10** describes the functions of the five PIO control registers. PPAR<sub>x</sub> defines each pin in the PIO port as either a GPIO pin or a pin dedicated to one of the communications controllers or communications processor peripherals. If a dedicated pin in the PPAR<sub>x</sub> register is chosen, it can either be of type dedicated 1 or dedicated 2, as set in the PSOR<sub>x</sub> register. Dedicated 1 and 2 refers to which multiplexed function on the pin is selected. Dedicated 1 and 2 functionality is defined in the “Dedicated Pin Assignment” tables in the chapter on parallel ports in the *MSC8101 Reference Manual*. PDIR<sub>x</sub> configures the pin as either an input or output. If the pin is an output, it can be either a regular or open-drain output, as set by PODR<sub>x</sub>. Data is read/written from/to the pin through the PDAT<sub>x</sub> register.

**Table 10.** PIO Port Register Functional Operation

Register Name	0	1	Description
PPAR <sub>x</sub>	GPIO	Dedicated	PIO Port Pin Assignment Register
PSOR <sub>x</sub>	Dedicated 1	Dedicated 2	PIO Special Options Register
PDIR <sub>x</sub>	Input	Output	PIO Direction Register (bidirectional signals must be programmed as input (PDIR=0))
PODR <sub>x</sub>	Regular	Open-Drain	PIO Open-Drain Register
PDAT <sub>x</sub>	0	1	PIO Data Register

Following are the procedures for programming a PIO port pin as a GPIO or a dedicated I/O function, respectively. An alternative method is to use a tool called *Pin\_mux8101*, which is described in **Section 4, *Pin\_mux8101 Application***, on page 22.

If a GPIO is required, perform the following steps:

1. Set the appropriate bit(s) in PPAR $x$  as zero. For example, if PA31 is a GPIO, clear bit 31 in PPARA.
2. If the PIO port pin is an output, set the corresponding pin as either actively driven by the MSC8101 or an open-drain driver. For example, if PA31 is actively driven, clear bit 31 in PODRA.
3. If the PIO port pin is an output, write the initial value in PDAT $x$ . For example, if PA31 needs to output as a one, set bit 31 in PDATA.
4. Set the PIO port pin data direction (input or output) in PDIR $x$ . For example, if PA31 is an output, set bit 31 in PDIRA.

If a dedicated I/O function is required, perform the following steps:

1. The CPM pin functions are multiplexed. Refer to the chapter on parallel I/O ports in the *MSC8101 Reference Manual*, under the tables titled “Port xxx--Dedicated Pin Assignment (PPAR $_{xxx}$  = 1),” and find the required I/O pin function. For example, if an SPI interface is required, find the pins defining the SPI interface ( $\overline{\text{SPISEL}}$ , SPICLK, SPIMOSI, SPIMISO). These are signals PD[16–19]. Note the columns the pins fall under (that is, PSOR $_{xxx}$  = 0 or 1, PIR $_{xxx}$  = 0 or 1). For example, the SPI pins all fall under the column to set PSORD = 1 and PDIRD = 0.
2. Set the bits in the PSOR $x$  as described in step one. For the SPI example, set bits 16–19 in PSORD.
3. If the PIO port pin is an output, set the corresponding pin as either actively driven by the MSC8101 or as an open-drain driver in PODRD. An open-drain driver drives a pin active low. Otherwise, the pin is three stated. For example, if the SPI is the master,  $\overline{\text{SPISEL}}$ , SPICLK, SPIMOSI are outputs. If these pins are to be driven actively, clear bits 17–19 in PODRD.
4. If the PIO port pin is an output, write the initial value in PDAT $x$ . For example, for an SPI master,  $\overline{\text{SPISEL}}$ , SPICLK, and SPIMOSI are outputs. If these pins are to be initially output as a one, set bits 17–19 in PDATA.
5. Set the PIO port pin data direction (input or output) in PDIR $x$ . For example, if the SPI interface is for a master,  $\overline{\text{SPISEL}}$ , SPICLK, and SPIMOSI are all outputs and SPIMISO is an input. Therefore, set bits 17–19 and clear bit 16 in PDIRD.
6. Set the corresponding PIO port pin assignment for dedicated in the PPAR $_{xxx}$  register. All pins other than the GPIO pins are dedicated CPM pins. For example, all the SPI interface pins ( $\overline{\text{SPISEL}}$ , SPICLK, SPIMOSI, and SPIMISO) are dedicated CPM pins. Therefore, set bits 16–19 in PPARD.

## 3.2 Clocking

The clocks for the communications controllers derive from a bank of eight internal BRGs and ten external clock (CLK) pins. The flexibility in choosing from several external clocks or internal BRGs is as follows:

- A serial device is not required to choose a serial device clock from a predefined pin or BRG.
- A group of serial receivers and transmitters can share the same clock rate from the same pin.

**Table 11** shows the clock source options for the communications controllers and TDM channels.

**Table 11.** Clock and BRG Options for Communications Controllers/TDM Channels

Controller/ Channel	CL K1	CL K2	CL K3	CL K4	CL K5	CL K6	CL K7	CL K8	CL K9	CLK 10	BR G1	BR G2	BR G3	BR G4	BR G5	BR G6	BR G7	BR G8
SCC1 RCLKx	3	3	3	3							3	3	3	3				
SCC1 TCLKx	3	3	3	3							3	3	3	3				
SCC2 RCLKx	3	3	3	3							3	3	3	3				
SCC2 TCLKx	3	3	3	3							3	3	3	3				
FCC1 RCLKx	3	3							3	3					3	3	3	3
FCC1 TCLKx	3	3							3	3					3	3	3	3
FCC2 RCLKx			3	3	3	3									3	3	3	3
FCC2 TCLKx			3	3	3	3									3	3	3	3
TDMA1 Rx	3								3									
TDMA1 Tx		3								3								
TDMB2 Rx					3		3											
TDMB2 Tx						3		3										
TDMC2 Rx			3				3											
TDMC2 Tx				3				3										
TDMD2 Rx	3								3									
TDMD2 Tx		3								3								
SMC1 SMCLKx							3		3		3						3	
SMC2 SMCLKx									3	3		3						3
TIN1				3														
TIN2			3															
TIN3								3										
TIN4							3											
BRG1			3		3													
BRG2			3		3													
BRG3					3				3									
BRG4					3				3									
BRG5			3		3													
BRG6			3		3													
BRG7					3				3									
BRG8					3				3									

There are two restrictions on the bank-of-clocks mapping:

- Only four of the ten sources can connect to any given FCC or SCC receiver or transmitter.
- The SMC transmitter and receiver share the same clock source when connected to the NMSI.

As **Table 11** shows, the internal BRGs (BRG[1–8]) and timers (TIN[1–4]) can be directed externally to the clock pins. For further information, refer to the chapters on CPM multiplexing and BRGs in the *MSC8101 Reference Manual*.

### 3.3 Serial Channels by Function

**Table 12** groups the I/O pin signals by function. Notice that a signal may not be available because of pin multiplexing. For example, if FCC1 MII is required, the FCC1 UTOPIA 8 Master and Slave, HDLC, and Transparent signal groupings are not available. The *Clock Signal Names* column indicates the names of the clock source for the channel. The clock source can derive from the internal BRG or an external clock.

**Table 12.** Signal Names Grouped by Function

Function	Signal Names	Clock Signal Names	Channel-Group Functions Not Available Because of Pin Muxing
FCC1: MII Fast Ethernet	COL, CRS, TX_EN, TX_ER, RX_DV, RX_ER, TXD[0–3], RXD[0–3]	TCLK, RCLK	<b>FCC1:</b> UTOPIA 8 Master / Slave, HDLC serial/nibble, and Transparent operations <b>SDMA:</b> MSNUM[0–1] <b>GPIO:</b> PA[14–31]
FCC1: UTOPIA 8 Multi-PHY Master multiplexed polling	TXD[0–7], RXD[0–7], TXADDR[0–4], RXADDR[0–4], TXENB, RXENB, TXCLAV, RXCLAV, TXSOC, RXSOC, TXPRTY, RXPRTY	TCLK, RCLK	<b>FCC1:</b> MII Ethernet, UTOPIA 8 Slave/Master direct polling, HDLC serial/nibble, Transparent <b>FCC2:</b> HDLC serial/nibble, Transparent <b>SCC1:</b> Ethernet, HDLC serial, UART, Transparent <b>SCC2:</b> Ethernet, HDLC serial, UART, Transparent <b>SDMA</b> <b>SMC1</b> <b>SPI</b> <b>SI1:</b> TDMA1 <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PA[10–31], PC [6–7, 12–15], PD[7, 16–19, 29]
FCC1: UTOPIA 8 Multi-PHY Master direct polling	TXD[0–7], RXD[0–7], TXADDR[0–2], RXADDR[0–2], TXENB, RXENB, TXCLAV[0–3], RXCLAV[0–3], TXSOC, RXSOC, TXPRTY, RXPRTY  <b>Note:</b> TXCLAV1 is multiplexed with TXADDR2 and RXCLAV1 is multiplexed with RXADDR2	TCLK, RCLK	<b>FCC1:</b> MII Ethernet, UTOPIA 8 Slave/Master direct polling, HDLC serial/nibble, Transparent <b>FCC2:</b> HDLC serial/nibble, Transparent <b>SCC1:</b> Ethernet, HDLC serial, UART, Transparent <b>SCC2:</b> Ethernet, HDLC serial, UART, Transparent <b>SDMA</b> <b>SMC1</b> <b>SPI</b> <b>SI1:</b> TDMA1 <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PA[10–31], PC [6–7, 12–15], PD[7, 16–19, 29]

**Table 12.** Signal Names Grouped by Function (Continued)

Function	Signal Names	Clock Signal Names	Channel-Group Functions Not Available Because of Pin Muxing
FCC1: UTOPIA 8 Slave	TXD[0–7], RXD[0–7], TXADDR[0–4], RXADDR[0–4], TXENB, RXENB, TXCLAV, RXCLAV, TXSOC, RXSOC, TXPRTY, RXPRTY	TCLK, RCLK	<b>FCC1:</b> MII Ethernet, UTOPIA 8 Master, HDLC serial/nibble, Transparent <b>FCC2:</b> HDLC serial/nibble, Transparent <b>SCC1:</b> Ethernet, HDLC serial, UART, Transparent <b>SCC2:</b> Ethernet, HDLC serial, UART, Transparent <b>SDMA</b> <b>SMC1</b> <b>SPI</b> <b>SI1:</b> TDMA1 <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PA[10–31], PC [6–7, 12–15], PD[7, 16–19, 29]
FCC1: HDLC serial mode	TXD, RXD, $\overline{\text{RTS}}$ , $\overline{\text{CTS}}$ , $\overline{\text{CD}}$	TCLK, RCLK	<b>FCC1:</b> MII Ethernet, UTOPIA 8 Master/Slave, HDLC nibble, Transparent <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PA[17–18, 30], PB[21–22, 28], PC[6–7]
FCC1: HDLC nibble mode	TXD[0–3], RXD[0–3], $\overline{\text{RTS}}$ , $\overline{\text{CTS}}$ , $\overline{\text{CD}}$	TCLK, RCLK	<b>FCC1:</b> MII Ethernet, UTOPIA 8 Master/Slave, HDLC serial, Transparent <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PA[14–21, 30], PB[21–22, 28], PC[6–7]
FCC1: Transparent	TXD, RXD, $\overline{\text{RTS}}$ , $\overline{\text{CTS}}$ , $\overline{\text{CD}}$	TCLK, RCLK	<b>FCC1:</b> MII Ethernet, UTOPIA 8 Master/Slave, HDLC serial/nibble <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PA[17–18, 30], PB[21–22, 28], PC[6–7]
FCC2: MII Ethernet	COL, CRS, TX_EN, TX_ER, RX_DV, RX_ER, TXD[0–3], RXD[0–3]	TCLK, RCLK	<b>FCC2:</b> HDLC serial/nibble, Transparent <b>SCC2:</b> Ethernet, HDLC serial, UART, Transparent <b>I2C</b> <b>SI1:</b> TDMA1 nibble <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PB[18–31]
FCC2: HDLC serial mode	TXD, RXD, $\overline{\text{RTS}}$ , $\overline{\text{CTS}}$ , $\overline{\text{CD}}$	TCLK, RCLK	<b>FCC2:</b> HDLC nibble, Transparent, MII Ethernet <b>SCC2:</b> Ethernet, HDLC serial, UART, Transparent <b>SMC1</b> <b>SI1:</b> TDMA1 serial/nibble <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PB[21–22, 28], PC[4–5]
FCC2: HDLC nibble mode	TXD[0–3], RXD[0–3], $\overline{\text{RTS}}$ , $\overline{\text{CTS}}$ , $\overline{\text{CD}}$	TCLK, RCLK	<b>FCC2:</b> HDLC serial, Transparent, MII Ethernet <b>SCC2:</b> HDLC serial, UART, Transparent <b>SMC1</b> <b>I2C</b> <b>SI1:</b> TDMA1 serial/nibble <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PB[18–25, 28], PC[4–5]

**Table 12.** Signal Names Grouped by Function (Continued)

Function	Signal Names	Clock Signal Names	Channel-Group Functions Not Available Because of Pin Muxing
FCC2: Transparent	TXD, RXD, $\overline{\text{RTS}}$ , $\overline{\text{CTS}}$ , $\overline{\text{CD}}$	TCLK, RCLK	<b>FCC2:</b> HDLC serial/nibble, MII Ethernet <b>SCC2:</b> HDLC serial, UART, Transparent <b>SMC1</b> <b>SI1:</b> TDMA1 serial/nibble <b>SI2:</b> TDMB2, TDMC2, TDMD2 <b>GPIO:</b> PB[21–22, 28], PC[4–5]
SCC1: Ethernet	TXD, RXD, $\overline{\text{RTS}}$ /TENA, $\overline{\text{CTS}}$ /CLSN, $\overline{\text{CD}}$ /RENA  <b>Note:</b> $\overline{\text{CTS}}$ /CLSN has two possible pin connections.	TCLK, RCLK	<b>FCC1:</b> UTOPIA8 Master/Slave <b>SCC1:</b> HDLC/UART/Transparent <b>SI1:</b> TDMA1 serial/nibble <b>SMC2</b> (if $\overline{\text{CTS}}$ /CLSN uses PC15) <b>DMA1/DMA2</b> (DRACK/DONE protocol) <b>Timers:</b> Timer 2 (if $\overline{\text{CTS}}$ /CLSN uses PC29) <b>BRGs:</b> BRG30 (if $\overline{\text{CTS}}$ /CLSN uses PC29) <b>Clocks:</b> CLK3 (if $\overline{\text{CTS}}$ /CLSN uses PC29) <b>GPIO:</b> PC14, PC15 (or PC29), PD[29–31]
SCC1: HDLC serial/UART/Transparent	TXD, RXD, $\overline{\text{RTS}}$ , $\overline{\text{CTS}}$ , $\overline{\text{CD}}$  <b>Note:</b> $\overline{\text{CTS}}$ has two possible pin connections.	TCLK, RCLK  <b>Note:</b> UART does not use a clock	<b>FCC1:</b> UTOPIA8 Master/Slave <b>SCC1:</b> Ethernet <b>SI1:</b> TDMA1 serial/nibble <b>SMC2</b> (if CTS uses PC15) <b>DMA1/DMA2</b> (DRACK/DONE protocol) <b>Timers:</b> Timer 2 (if $\overline{\text{CTS}}$ uses PC29) <b>BRGs:</b> BRG30 (if $\overline{\text{CTS}}$ uses PC29) <b>Clocks:</b> CLK3 (if $\overline{\text{CTS}}$ uses PC29) <b>GPIO:</b> PC14, PC15 (or PC29), PD[29–31]
SCC2: Ethernet	TXD, RXD, $\overline{\text{RTS}}$ /TENA, $\overline{\text{CTS}}$ /CLSN, $\overline{\text{CD}}$ /RENA  <b>Note:</b> $\overline{\text{CTS}}$ /CLSN has two possible pin connections.	TCLK, RCLK	<b>FCC1:</b> UTOPIA 8 Master/Slave <b>FCC2:</b> MII Ethernet, HDLC, Transparent <b>SCC2:</b> HDLC serial, UART, Transparent <b>SI1:</b> LIST[3–4] <b>SI2:</b> TDMB2 <b>Timers:</b> Timer 1/2 (if $\overline{\text{CTS}}$ /CLSN uses PC28) <b>BRGs:</b> BRG40 (if $\overline{\text{CTS}}$ /CLSN uses PC28) <b>Clocks:</b> CLK4 (if $\overline{\text{CTS}}$ /CLSN uses PC28) <b>GPIO:</b> PB[28, 30–31], PC12, PC13 (or PC28)
SCC2: HDLC serial/UART/Transparent	TXD, RXD, $\overline{\text{RTS}}$ , $\overline{\text{CTS}}$ , $\overline{\text{CD}}$  <b>Note:</b> $\overline{\text{CTS}}$ has two possible pin connections.	TCLK, RCLK  <b>Note:</b> UART does not use a clock	<b>FCC1:</b> UTOPIA 8 Master/Slave <b>FCC2:</b> MII Ethernet/HDLC/ Transparent <b>SCC2:</b> Ethernet <b>SI1:</b> LIST[3–4] <b>SI2:</b> TDMB2 <b>Timers:</b> Timer 1/2 (if $\overline{\text{CTS}}$ uses PC28) <b>BRGs:</b> BRG40 (if $\overline{\text{CTS}}$ uses PC28) <b>Clocks:</b> CLK4 (if $\overline{\text{CTS}}$ uses PC28) <b>GPIO:</b> PB[28, 30–31], PC12, PC13 (or PC28)
SDMA	MSNUM[0–5]		<b>FCC1:</b> UTOPIA 8 Master/Slave <b>GPIO:</b> PA[10–13, 24–25]
SMC1	SMTXD, SMRXD, $\overline{\text{SMSYN}}$		<b>FCC1:</b> UTOPIA 8 Master/Slave, <b>FCC2:</b> HDLC, Transparent <b>SI2:</b> LIST[3–4] <b>GPIO:</b> PD7, PC[4–5]

**Table 12.** Signal Names Grouped by Function (Continued)

Function	Signal Names	Clock Signal Names	Channel-Group Functions Not Available Because of Pin Muxing
SMC2	SMTXD, SMRXD, SMSYN  <b>Note:</b> SMTXD has two possible pin connections.		<b>FCC1:</b> UTOPIA 8 Master/Slave (if PC15 is used as SMTXD) <b>TDMA1:</b> Serial/Nibble <b>GPIO:</b> PC15 (or PA9), PA[7–8]
SPI	SPISEL, SPICLK, SPIMOSI, SPIMISO		<b>FCC1:</b> UTOPIA 8 Master/Slave <b>BRG:</b> BRG10 (PD19), BRG20 (PD17) <b>GPIO:</b> PD[16–19]
I <sup>2</sup> C	SDA, SCL		<b>FCC2:</b> MII Ethernet, HDLC, Transparent <b>GPIO:</b> [PB18–19]
TDMA1 Serial	L1TXD, L1RXD, L1TSYNC/GRANT, plus one of the LIST[1–4] strobes	L1TCLK, L1RCLK	<b>SMC2</b> <b>TDMA1:</b> Nibble <b>GPIO:</b> PA[6–9] <i>If LIST1 used:</i> <b>DMA1</b> <b>Clocks:</b> CLK10 <b>GPIO:</b> PC22 <i>If LIST2 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave <b>SCC1:</b> all modes <b>GPIO:</b> PC14 <i>If LIST3 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave <b>SCC2:</b> all modes <b>GPIO:</b> PC12 <i>If LIST4 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave <b>SCC2:</b> all modes <b>GPIO:</b> PC13
TDMA1 Nibble	L1TXD[0–3], L1RXD[0–3], L1TSYNC/GRANT, L1RSYNC, plus one of the LIST[1–4] strobes	L1TCLK, L1RCLK	<b>SMC2</b> <b>TDMA1:</b> Serial <b>FCC2:</b> MII Ethernet, HDLC, Transparent <b>TDMC2</b> <b>TDMD2</b> <b>GPIO:</b> PA[6–9], PB[20–25] <i>If LIST1 used:</i> <b>GPIO:</b> PC22 <i>If LIST2 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave <b>SCC1:</b> all modes <b>GPIO:</b> PC14 <i>If LIST3 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave <b>SCC2:</b> all modes <b>GPIO:</b> PC12 <i>If LIST4 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave <b>SCC2:</b> all modes <b>GPIO:</b> PC13

**Table 12.** Signal Names Grouped by Function (Continued)

Function	Signal Names	Clock Signal Names	Channel-Group Functions Not Available Because of Pin Muxing
TDMB2	L1TXD, L1RXD, L1TSYNC, L1RSYNC, plus one of the LIST[1–4] strobes	L1TCLK, L1RCLK	<b>FCC2:</b> MII Ethernet, HDLC, Transparent <b>SCC2:</b> Ethernet, HDLC serial, UART, Transparent <b>GPIO:</b> PB[28–31] <i>If LIST1 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave, HDLC, Transparent <b>GPIO:</b> PC7 <i>If LIST2 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave, HDLC, Transparent <b>GPIO:</b> PC6 <i>If LIST3 used:</i> <b>FCC2:</b> HDLC, Transparent <b>SMC1</b> <b>GPIO:</b> PC5 <i>If LIST4 used:</i> <b>FCC2:</b> HDLC, Transparent <b>SMC1</b> <b>GPIO:</b> PC4
TDMC2	L1TXD, L1RXD, L1TSYNC, L1RSYNC, plus one of the LIST[1–4] strobes	L1TCLK, L1RCLK	<b>FCC2:</b> MII Ethernet, HDLC nibble <b>TDMA1:</b> Nibble <b>GPIO:</b> PB[24–27] <i>If LIST1 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave, HDLC, Transparent <b>GPIO:</b> PC7 <i>If LIST2 used:</i> <b>FCC1:</b> UTOPIA 8 Master/Slave, HDLC, Transparent <b>GPIO:</b> PC6 <i>If LIST3 used:</i> <b>FCC2:</b> HDLC, Transparent <b>SMC1</b> <b>GPIO:</b> PC5 <i>If LIST4 used:</i> <b>FCC2:</b> HDLC, Transparent <b>SMC1</b> <b>GPIO:</b> PC4

**Table 12.** Signal Names Grouped by Function (Continued)

Function	Signal Names	Clock Signal Names	Channel-Group Functions Not Available Because of Pin Muxing
TDMD2	L1TXD, L1RXD, L1TSYNC, L1RSYNC, plus one of the LIST[1–4] strobes	L1TCLK, L1RCLK	<p><b>FCC2:</b> MII Ethernet, HDLC serial/nibble, Transparent</p> <p><b>TDMA1:</b> Nibble</p> <p><b>GPIO:</b> PB[20–23]</p> <p><i>If LIST1 used:</i></p> <p><b>FCC1:</b> UTOPIA 8 Master/Slave, HDLC, Transparent</p> <p><b>GPIO:</b> PC7</p> <p><i>If LIST2 used:</i></p> <p><b>FCC1:</b> UTOPIA 8 Master/Slave, HDLC, Transparent</p> <p><b>GPIO:</b> PC6</p> <p><i>If LIST3 used:</i></p> <p><b>FCC2:</b> HDLC, Transparent</p> <p><b>SMC1</b></p> <p><b>GPIO:</b> PC5</p> <p><i>If LIST4 used:</i></p> <p><b>FCC2:</b> HDLC, Transparent</p> <p><b>SMC1</b></p> <p><b>GPIO:</b> PC4</p>
DMA1	DREQ1, $\overline{\text{DACK1}}$ , $\overline{\text{DRACK1/DONE1}}$		<p><b>SCC1:</b> All modes (if DMA uses DRACK/DONE protocol)</p> <p><b>S11:</b> LIST1</p> <p><b>EXT2</b></p> <p><b>Clocks:</b> CLK9, CLK10</p> <p><b>GPIO:</b> PC[22–23], PD31</p>
DMA2	DREQ2, $\overline{\text{DACK2}}$ , $\overline{\text{DRACK2/DONE2}}$		<p><b>SCC1:</b> All modes (if DMA uses <math>\overline{\text{DRACK/DONE}}</math> protocol)</p> <p><b>BRG:</b> BRG7, BRG8</p> <p><b>Clocks:</b> CLK7, CLK8</p> <p><b>Timers:</b> Timer 3, Timer 4</p> <p><b>GPIO:</b> PC[22–23], PD31</p>
BRG1	BRG10 <b>Note:</b> BRG10 has two possible pin connections.		<p><b>Clocks:</b> CLK1 (for PC31)</p> <p><b>Timers:</b> Timer 1 (for PC31)</p> <p><b>FCC1:</b> UTOPIA 8 Master/Slave (for PD19)</p> <p><b>SPI</b> (for PD19)</p> <p><b>GPIO:</b> PC31 or PD19</p>
BRG2	BRG20 <b>Note:</b> BRG20 has two possible pin connections.		<p><b>EXT1</b></p> <p><b>Clocks:</b> CLK2 (for PC30)</p> <p><b>Timers:</b> Timer 1 (for PC30)</p> <p><b>FCC1:</b> UTOPIA 8 Master/Slave (for PD17)</p> <p><b>SPI</b> (for PD17)</p> <p><b>GPIO:</b> PC30 or PD17</p>
BRG3	BRG30		<p><b>SCC1</b> (if <math>\overline{\text{CTS/CLSN}}</math> uses PC29)</p> <p><b>Clocks:</b> CLK3</p> <p><b>Timers:</b> Timer 2</p> <p><b>GPIO:</b> PC29</p>
BRG4	BRG40		<p><b>Clocks:</b> CLK4</p> <p><b>Timers:</b> Timer 1, Timer 2</p> <p><b>GPIO:</b> PC28</p>

**Table 12.** Signal Names Grouped by Function (Continued)

Function	Signal Names	Clock Signal Names	Channel-Group Functions Not Available Because of Pin Muxing
BRG5	BRG50		<b>Clocks:</b> CLK5 <b>Timers:</b> Timer 2 <b>GPIO:</b> PC27
BRG6	BRG60		<b>Clocks:</b> CLK6 <b>Timers:</b> Timer 3 <b>TMCLK</b> (input to SIU) <b>GPIO:</b> PC26
BRG7	BRG70		<b>DMA2</b> <b>Clocks:</b> CLK7 <b>Timers:</b> Timer 4 <b>GPIO:</b> PC25
BRG8	BRG80		<b>DMA2</b> <b>Clocks:</b> CLK8 <b>Timers:</b> Timer 3, Timer 4 <b>GPIO:</b> PC24
CLK1		CLK1	<b>BRG:</b> BRG1 (PC31 only) <b>Timers:</b> Timer 1 <b>GPIO:</b> PC31
CLK2		CLK2	<b>BRG:</b> BRG2 (PC30 only) <b>Timers:</b> Timer 1, Timer 2 <b>EXT1</b> <b>GPIO:</b> PC30
CLK3		CLK3	<b>BRG:</b> BRG3 <b>Timers:</b> Timer 2 <b>SCC1</b> (if CTS/CLSN uses PC29) <b>GPIO:</b> PC29
CLK4		CLK4	<b>BRG:</b> BRG4 <b>Timers:</b> Timer 1, Timer 2 <b>SCC2</b> (if CTS/CLSN uses PC28) <b>GPIO:</b> PC28
CLK5		CLK5	<b>BRG:</b> BRG5 <b>Timers:</b> Timer 3, Timer 4 <b>GPIO:</b> PC27
CLK6		CLK6	<b>BRG:</b> BRG6 <b>Timers:</b> Timer 3 <b>TMCLK</b> (input to SIU) <b>GPIO:</b> PC26
CLK7		CLK7	<b>DMA2</b> <b>BRG:</b> BRG7 <b>Timers:</b> Timer 4 <b>GPIO:</b> PC25
CLK8		CLK8	<b>DMA2</b> <b>BRG:</b> BRG8 <b>Timers:</b> Timer 3, Timer 4 <b>GPIO:</b> PC24
CLK9		CLK9	<b>DMA1</b> <b>EXT2</b> <b>GPIO:</b> PC23

**Table 12.** Signal Names Grouped by Function (Continued)

Function	Signal Names	Clock Signal Names	Channel-Group Functions Not Available Because of Pin Muxing
CLK10		CLK10	<b>DMA1</b> <b>SI1:</b> LIST1 <b>GPIO:</b> PC22
Timer 1	TIN1, $\overline{\text{TGATE1}}$ , $\overline{\text{TOUT1}}$		<b>SCC2</b> (if $\overline{\text{CTS/CLSN}}$ uses PC28) <b>BRG:</b> BRG1, BRG2, BRG4 <b>Clocks:</b> CLK1, CLK2, CLK4 <b>Timers:</b> TIMER 2 <b>GPIO:</b> PC[28, 30–31]
Timer 2	TIN2, $\overline{\text{TGATE1}}$ , $\overline{\text{TOUT2}}$		<b>SCC1</b> (if $\overline{\text{CTS/CLSN}}$ uses PC29) <b>SCC2</b> (if $\overline{\text{CTS/CLSN}}$ uses PC28) <b>BRG:</b> BRG1, BRG3, BRG4 <b>Clocks:</b> CLK1, CLK3, CLK4 <b>Timers:</b> TIMER 1 <b>GPIO:</b> PC[28–29, 31]
Timer 3	TIN3, $\overline{\text{TGATE2}}$ , $\overline{\text{TOUT3}}$		<b>DMA2</b> <b>BRG:</b> BRG5, BRG6, BRG8 <b>Clocks:</b> CLK5, CLK6, CLK8 <b>Timers:</b> TIMER 4 <b>TMCLK</b> <b>GPIO:</b> PC[24, 26–27]
Timer 4	TIN4, $\overline{\text{TGATE2}}$ , $\overline{\text{TOUT4}}$		<b>DMA2</b> <b>BRG:</b> BRG5, BRG7, BRG8 <b>Clocks:</b> CLK5, CLK7, CLK8 <b>Timers:</b> TIMER 3 <b>GPIO:</b> PC[24–25, 27]
SIU Clock input	TMCLK		<b>BRG:</b> BRG6 <b>Clocks:</b> CLK6 <b>Timers:</b> Timer 3 <b>GPIO:</b> PC26
External Input 1	EXT1		<b>BRG:</b> BRG2 <b>Clocks:</b> CLK2 <b>Timers:</b> Timer 1 <b>GPIO:</b> PC30
External Input 2	EXT2		<b>DMA2</b> <b>BRG:</b> BRG6 <b>Clocks:</b> CLK9 <b>GPIO:</b> PC23

## 4 Pin\_mux8101 Application

*Pin\_mux8101* is a PC Windows™ utility that defines configurations and generates initialization code for the MSC8101 PIO ports. *Pin\_mux8101* is available in a .zip file named AN1854SW.zip at the Freescale web site listed on the back cover of this document. *Pin\_mux8101* supports the following:

- I/O pins can be selected on the basis of signal grouping.
- Signal groups can be customized to remove a signal.
- Signal groups are shown that are not available because of pin multiplexing.
- GPIO pins can be configured as inputs, outputs, or output open-drain.

- Dedicated output pins can be configured as outputs or output open-drain.
- A detailed report can be generated with C initialization code.

*Pin\_mux8101* can save you considerable time in selecting the I/O pins and generating initialization code. However, it can be of real value only if you understand its strengths and weaknesses. In addition, it is recommended that you become thoroughly familiar with the application I/O requirements and the signals for each interface the application implements. Following are two examples highlighting the importance of understanding the signals in the interface and not relying completely on the tool (*Pin\_mux8101*).

Example 1:

The SPI interface supports one or more slaves. For multiple slaves, multiple GPIOs, independent of the SPI signals lines (SPISEL, SPICLK, SPIMOSI, SPIMISO), may be required to select the SPI slaves.

Example 2:

The signals defined for Ethernet (TXD, RXD,  $\overline{\text{RTS}}/\text{RENA}$ ,  $\overline{\text{CTS}}/\text{CLSN}$ ,  $\overline{\text{CD}}/\text{RENA}$ , TCLK, RCLK) do not include any additional signals to control and read status back from the Ethernet driver. This requires extra PIO and/or address/data lines. *Pin\_mux8101* can configure GPIO lines.

Do not rely on a tool unless you know its strengths and weaknesses, as well as the particular requirements. The *MSC8101 Reference Manual* is the best place to start in understanding the MSC8101 PIO.

## 4.1 Strengths and Weaknesses

*Pin\_mux8101* strengths are as follows.

- Manages all PIO pins.
- Provides a report on all of the MSC8101 pins.
- Generates code in C to initialize the PIO.
- Saves considerable work in setting up the PIO ports.
- Provides a check if you hand code the PIO ports.

*Pin\_mux8101* weaknesses are as follows:

- Clocks and BRGs are not part of the pin groupings in *Pin\_mux8101* and must be selected manually (<Allocate>, <Groups of pins> menu). Refer to **Section 3.2** for details on which clocks and BRGs are available for each communications controller and TDM channel. The only CPM pin groupings that do not require additional clocks or BRGs are the SPI and I<sup>2</sup>C communications controllers. Use the *Clock Source Options* table in **Section 3.2** alongside *Pin\_mux8101*.
- *Pin\_mux8101* initializes only the physical PIO ports. Other register(s) may need to be programmed. For example, *Pin\_mux8101* supports the SPI interface, but programming for master/ slave, number of bits, polarity of bits, and so on must still be programmed unless the defaults are used.
- Knowledge of the pins being programmed is still required. For example, most but not all of the *Pin\_mux8101* signal groupings require a clock source to be selected manually. SPI and I<sup>2</sup>C are examples of signal groupings that do not require a clock source to be selected outside the signal group.
- A function can be supported on several pins. However, *Pin\_mux8101* supports only one of the pins with that function. If another pin with the same function is required, you must manually change the code generated by *Pin\_mux8101*.

## 4.2 Menus

Table 13 shows *Pin\_mux8101* menus and their descriptions.

**Table 13.** Pin\_mux8101 Menus

Menu	Submenu	Description
File	Initialize	Reset all parameters. This is the default state when <i>Pin_mux8101</i> is started.
	Load Configuration	Load parameter file.
	Save Configuration	Save all parameters to a file.
	Exit	Quit <i>Pin_mux8101</i> .
Allocate	Pins in groups	Select which pins in each group to use. For example, it can remove the CD signal from <i>SCC1-HDLC/UART/TRANSPARENT</i> grouping.
	Groups of pins	Select the CPM functionality required. For example, <i>SCC1-ETHERNET</i> . Note: signals can be removed from a grouping through the <i>Pins in groups</i> menu.
	Parallel ports	Modify all unused PIO pins from INPUT to either <i>INPUT</i> , <i>OUTPUT</i> , or <i>OUTPUT-OPEN DRAIN</i> .
	Open-drain outputs	Modify all output pins selected in a group to <i>OUTPUT</i> or <i>OUTPUT-OPEN DRAIN</i> .
Report	Save report to file	Generate a report and save it to a file.
	Display report file	Display a report file. Note that a report can be opened with any text editor.
Help	Display help file	Display on-line help.
	Program version	Display the version number of <i>Pin_mux8101</i> and copyright notice.

## 4.3 Procedure

Table 14 shows a suggested procedure for using *Pin\_mux8101*.

**Table 14.** Pin\_mux8101 Procedure

Step	Menus	Description	Example
1	Start <i>Pin_mux8101</i>	Read the <i>Help</i> window that appears. This window can be left open or closed.	
2	Allocate, Pins in groups	Select the pins you want to be allowed in each group.	<p>UART using SCC1 with only TXD and RXD signals (remove RTS, CTS, and CD).</p> <p>Select <i>SCC1-HDLC/UART/TRANSPARENT</i>.</p> <p>In the <i>Pin used</i> group, select <i>RTS</i>, <i>CTS</i>, and <i>CD</i> to move these pins into the <i>Pins not used</i> group and make them available for other functions.</p> <p><i>Close</i>. Closes the window.</p>
3	Allocate, Group of pins	Select the group of pins to be used.	<p>UTOPIA 8 master using FCC1.</p> <p>Select <i>FCC1-UTOPIA8/MASTER</i>.</p> <p><i>Close</i>. Closes the window.</p>
4	Parallel ports	The available pins and the default state of the pins.	<p>pa0 is an input and pa1 is an output open drain.</p> <p>pa0 is already an input, so it is left alone.</p> <p>Select <i>pa1</i> twice to toggle <i>pa1</i> to an <i>OUTPUT-OPEN DRAIN</i>.</p> <p><i>Close</i>. Closes the window.</p>

**Table 14.** Pin\_mux8101 Procedure (Continued)

Step	Menus	Description	Example
5	Open-drain outputs	This window shows all the output pins generated from the selection made in <i>Groups of pins</i> and <i>Pins in groups</i> menus. The outputs can then be toggled between <i>OUTPUT</i> and <i>OPEN-DRAIN</i> outputs.	TXD from step 1 is being changed from an output to an output open drain. Select <i>PD30 SCC1</i> to toggle the output to an <i>OPEN-DRAIN</i> . <i>Close</i> . Closes the window.
6	(optional) Save configuration	Saves all the settings.	<i>File, Save configuration, &lt;filename&gt;, Open</i> .
7	Report, Save report to file	Generate report with all settings with C code. The report is saved in ASCII.	<i>Report, Save report to file, &lt;filename&gt;, Open</i> .
8	Report, Display report file	Display a report including all settings with C code.	<i>Report, Display report file, &lt;filename&gt;, Open</i> .

## 4.4 Application Example

**Table 15** shows an example using *Pin\_mux8101*. **Table 15** contains two columns for the communications controller (*initial* and *revised* pick). The Communications Controller (*initial* pick) column is the anticipated selection before *Pin\_mux8101* is used. The Communications Controller (*revised*) column is the revised column after *Pin\_mux8101* is used. A pin can be removed from a functional group through the *Pins in groups* menu. A pin should be removed before the signal groupings are selected. Then the signal groupings are selected from the following menus: *<Allocate>*, *<Groups of pins>*.

**Table 15.** Pin\_mux8101 Example (Channels)

Signal Groupings	Communications Controller (initial pick)	Communications Controller (revised)	Description
UTOPIA 8 Master	FCC1	FCC1	All lines used.
Ethernet 10BaseT	SCC1	FCC2 MII	All lines used.
UART 1	SCC1	SMC2 no SMSYN signal	TxD, RxD with no modem controls ( $\overline{\text{RTS}}$ , $\overline{\text{CTS}}$ , $\overline{\text{CD}}$ ). SMC1–2 supports a signal. For this example, SMSYN is removed from the SMC2 grouping. This needs to be done in the <i>Pins in groups</i> menu prior to selecting the channel groups in the <i>Groups of pins</i> menu.
PIO port: input	PA6	PA6	All unused pins default to an input.
PIO port: output open drain	PA7	PC4	Two output types: output, output open-drain.

*Pin\_mux8101* does not include the required clocks for each functional grouping. All functional groupings supported by *Pin\_mux8101* except SPI, I<sup>2</sup>C, and SMC1–2 require a clock source. The clock source can be either an external clock (CLKx) or an MSC8101 internally-generated clock (BRGx). These clock sources are selected from the same menu as the channel selections from within *Pin\_mux8101*: *<Allocate>*, *<Groups of pins>*. Not every communications controller supports every clock source. The available clock options are shown in **Table 11**. **Table 16** shows the initial pick and any necessary refinements for the example from **Table 15**.

**Table 16.** Pin\_mux8101 Example (Clocks)

Function	Clock (initial pick)	Clock (revised)	Description
UTOPIA 8 Master	CLK1	CLK1	none
Ethernet 10BaseT	CLK2	CLK2	none
UART 1	CLK3	CLK9	Clock source changed because went from SCC1 to SMC2
PIO port: input	no clock	no clock	No clock required
PIO port: output open-drain	no clock	no clock	No clock required

## 4.5 Handling Duplicate I/O

**Table 8**, *Duplicate I/O Pins*, on page 12 shows the dedicated I/O pins that have a similar function on another pin. **Table 17** expands upon this information, showing the pin that *Pin\_mux8101* selects. You cannot change this selection.

**Table 17.** Pin\_mux8101 Selection of Duplicate I/O Pins

Duplicate I/O Pin Function	I/O Pin	I/O Pin Supported by <i>Pin_mux8101</i>
SMC2: SMTXD	PA9, PC15	PA9
SCC2: $\overline{\text{CTS}}$ /CLSN	PC28, PC12	PC28
SCC1: $\overline{\text{CTS}}$ /CLSN	PC29, PC15	PC29
BRG10	PC31, PD19	PC31
BRD20	PC30, PD17	PC30

## 4.6 Reporting

*Pin\_mux8101* generates a report with the following information:

- List of allocated group of pins (used and unused pins).
- List of pins and allocated functions (how all pins are set up: input, output, bidirectional, not implemented).
- PIO port register programming.

The PIO port register programming consists of C code to configure the PIO ports. The code is broken down into six areas:

- *Code to clear the Port Pin Assignment Registers (PPARA–D)*. PPARx configures the PIO ports as either GPIO or dedicated I/O. In this case, the PIO ports are cleared to GPIO pins. It is good programming practice to start from a known state. Note that this is the same state as from a power-on reset.
- *Code to clear the Port Data Direction Registers (PDIRA–D)*. PDIRx configure the PIO port pins as either input or output. In this case, the PIO ports are cleared to inputs. This also includes support for bidirectional signals where applicable. It is good programming practice to start from a known state. Note that this is the same state as from a reset.

- *Code to configure the Port Special Options Registers (PSORA–D).* POSRx registers configure the PIO port pins as either dedicated 1 or 2 special operation; they are used only if the bits in PPARx are programmed to dedicated. If used, the PIO ports are cleared to dedicated 1 ports. It is good programming practice to start from a known state. Note that this is the same state as from a power-on reset.
- *Code to configure the Port Data Direction Registers (PDIRA–D).* PDIRx configures the PIO port pins to either input or output.
- *Code to configure the Port Open-Drain Registers (PODRA–D).* PODRx applies only if the bits in PDIRx are set to output(s). PODRx configures the PIO port pins to be actively driven by the MSC8101 or makes them open-drain output(s).
- *Code to configure the Port Pin Assignment Registers (PPARA–D).* PPARx configures the PIO port pins to either GPIO or dedicated I/O.

*Pin\_mux8101* does not configure the Port Data Registers (PDATA–D), which set the output value GPIOs either to 0 or 1. The outputs of the dedicated special operation pins are automatically set according to the protocol they support.

Following is the output *Pin\_mux8101* generated with the example in **Table 15**. The C code generated is located at the end of the listing. This code should be verified using the “Dedicated Pin Assignment” tables in the *CPM Parallel I/O* chapter in the *MSC8101 Reference Manual*.

```
List of allocated groups of pins
=====
Group of pins: FCC1-UTOPIA8/MASTER
-----
Used: TxADDR0
      TxADDR1
      TxADDR2
      TxADDR3
      TxADDR4
      RxADDR0
      RxADDR1
      RxADDR2
      RxADDR3
      RxADDR4
      TxEnb
      RxEnb
      TxClav
      RxClav
      TxSOC
      RxSOC
      TxPrty
      RxPrty
      TxD0
      TxD1
      TxD2
      TxD3
      TxD4
      TxD5
      TxD6
      TxD7
      RxD0
```

RxD1  
 RxD2  
 RxD3  
 RxD4  
 RxD5  
 RxD6  
 RxD7

Not used:

Group of pins: FCC2-MII

-----

Used: COL

CRS  
 RX\_DV  
 TX\_EN  
 RX\_ER  
 TX\_ER  
 TxD0  
 RxD0  
 TxD1  
 TxD2  
 TxD3  
 RxD1  
 RxD2  
 RxD3

Not used:

Group of pins: SMC2

-----

Used: SMTXD

SMRXD

Not used:SMSYN

Group of pins: CLK1

-----

Used: CLK1

Not used:

Group of pins: CLK2

-----

Used: CLK2

Not used:

Group of pins: CLK9

-----

Used: CLK9

Not used:

List of pins and allocated functions

=====

Pin Group - Function

-----

PA0 not implemented  
 PA1 not implemented  
 PA2 not implemented  
 PA3 not implemented  
 PA4 not implemented  
 PA5 not implemented  
 PA6 PARALLEL PORT - INPUT  
 PA7 PARALLEL PORT - INPUT

PA8 SMC2 - SMRXD - INPUT OR BIDIRECTIONAL  
 PA9 SMC2 - SMTXD - OUTPUT  
 PA10 FCC1-UTOPIA8/MASTER - RxD0 - INPUT OR BIDIRECTIONAL  
 PA11 FCC1-UTOPIA8/MASTER - RxD1 - INPUT OR BIDIRECTIONAL  
 PA12 FCC1-UTOPIA8/MASTER - RxD2 - INPUT OR BIDIRECTIONAL  
 PA13 FCC1-UTOPIA8/MASTER - RxD3 - INPUT OR BIDIRECTIONAL  
 PA14 FCC1-UTOPIA8/MASTER - RxD4 - INPUT OR BIDIRECTIONAL  
 PA15 FCC1-UTOPIA8/MASTER - RxD5 - INPUT OR BIDIRECTIONAL  
 PA16 FCC1-UTOPIA8/MASTER - RxD6 - INPUT OR BIDIRECTIONAL  
 PA17 FCC1-UTOPIA8/MASTER - RxD7 - INPUT OR BIDIRECTIONAL  
 PA18 FCC1-UTOPIA8/MASTER - TxD7 - OUTPUT  
 PA19 FCC1-UTOPIA8/MASTER - TxD6 - OUTPUT  
 PA20 FCC1-UTOPIA8/MASTER - TxD5 - OUTPUT  
 PA21 FCC1-UTOPIA8/MASTER - TxD4 - OUTPUT  
 PA22 FCC1-UTOPIA8/MASTER - TxD3 - OUTPUT  
 PA23 FCC1-UTOPIA8/MASTER - TxD2 - OUTPUT  
 PA24 FCC1-UTOPIA8/MASTER - TxD1 - OUTPUT  
 PA25 FCC1-UTOPIA8/MASTER - TxD0 - OUTPUT  
 PA26 FCC1-UTOPIA8/MASTER - RxClav - INPUT OR BIDIRECTIONAL  
 PA27 FCC1-UTOPIA8/MASTER - RxSOC - INPUT OR BIDIRECTIONAL  
 PA28 FCC1-UTOPIA8/MASTER - RxEnb - OUTPUT  
 PA29 FCC1-UTOPIA8/MASTER - TxSOC - OUTPUT  
 PA30 FCC1-UTOPIA8/MASTER - TxClav - INPUT OR BIDIRECTIONAL  
 PA31 FCC1-UTOPIA8/MASTER - TxEnb - OUTPUT  
 PB0 not implemented  
 PB1 not implemented  
 PB2 not implemented  
 PB3 not implemented  
 PB4 not implemented  
 PB5 not implemented  
 PB6 not implemented  
 PB7 not implemented  
 PB8 not implemented  
 PB9 not implemented  
 PB10 not implemented  
 PB11 not implemented  
 PB12 not implemented  
 PB13 not implemented  
 PB14 not implemented  
 PB15 not implemented  
 PB16 not implemented  
 PB17 not implemented  
 PB18 FCC2-MII - RxD3 - INPUT OR BIDIRECTIONAL  
 PB19 FCC2-MII - RxD2 - INPUT OR BIDIRECTIONAL  
 PB20 FCC2-MII - RxD1 - INPUT OR BIDIRECTIONAL  
 PB21 FCC2-MII - RxD0 - INPUT OR BIDIRECTIONAL  
 PB22 FCC2-MII - TxD0 - OUTPUT  
 PB23 FCC2-MII - TxD1 - OUTPUT  
 PB24 FCC2-MII - TxD2 - OUTPUT  
 PB25 FCC2-MII - TxD3 - OUTPUT  
 PB26 FCC2-MII - CRS - INPUT OR BIDIRECTIONAL  
 PB27 FCC2-MII - COL - INPUT OR BIDIRECTIONAL  
 PB28 FCC2-MII - RX\_ER - INPUT OR BIDIRECTIONAL  
 PB29 FCC2-MII - TX\_EN - OUTPUT  
 PB30 FCC2-MII - RX\_DV - INPUT OR BIDIRECTIONAL

```

PB31 FCC2-MII - TX_ER - OUTPUT
PC0 not implemented
PC1 not implemented
PC2 not implemented
PC3 not implemented
PC4 PARALLEL PORT - OUTPUT-OPEN DRAIN
PC5 PARALLEL PORT - INPUT
PC6 FCC1-UTOPIA8/MASTER - RxADDR2 - OUTPUT
PC7 FCC1-UTOPIA8/MASTER - TxADDR2 - OUTPUT
PC8 not implemented
PC9 not implemented
PC10 not implemented
PC11 not implemented
PC12 FCC1-UTOPIA8/MASTER - RxADDR1 - OUTPUT
PC13 FCC1-UTOPIA8/MASTER - TxADDR1 - OUTPUT
PC14 FCC1-UTOPIA8/MASTER - RxADDR0 - OUTPUT
PC15 FCC1-UTOPIA8/MASTER - TxADDR0 - OUTPUT
PC16 not implemented
PC17 not implemented
PC18 not implemented
PC19 not implemented
PC20 not implemented
PC21 not implemented
PC22 PARALLEL PORT - INPUT
PC23 CLK9 - CLK9 - INPUT OR BIDIRECTIONAL
PC24 PARALLEL PORT - INPUT
PC25 PARALLEL PORT - INPUT
PC26 PARALLEL PORT - INPUT
PC27 PARALLEL PORT - INPUT
PC28 PARALLEL PORT - INPUT
PC29 PARALLEL PORT - INPUT
PC30 CLK2 - CLK2 - INPUT OR BIDIRECTIONAL
PC31 CLK1 - CLK1 - INPUT OR BIDIRECTIONAL
PD0 not implemented
PD1 not implemented
PD2 not implemented
PD3 not implemented
PD4 not implemented
PD5 not implemented
PD6 not implemented
PD7 FCC1-UTOPIA8/MASTER - TxADDR3 - OUTPUT
PD8 not implemented
PD9 not implemented
PD10 not implemented
PD11 not implemented
PD12 not implemented
PD13 not implemented
PD14 not implemented
PD15 not implemented
PD16 FCC1-UTOPIA8/MASTER - TxPrty - OUTPUT
PD17 FCC1-UTOPIA8/MASTER - RxPrty - INPUT OR BIDIRECTIONAL
PD18 FCC1-UTOPIA8/MASTER - RxADDR4 - OUTPUT
PD19 FCC1-UTOPIA8/MASTER - TxADDR4 - OUTPUT
PD20 not implemented
PD21 not implemented

```

```

PD22 not implemented
PD23 not implemented
PD24 not implemented
PD25 not implemented
PD26 not implemented
PD27 not implemented
PD28 not implemented
PD29 FCC1-UTOPIA8/MASTER - RxADDR3 - OUTPUT
PD30 PARALLEL PORT - INPUT
PD31 PARALLEL PORT - INPUT
Parallel port registers programming
=====
/* MSC8101 parallel ports configuration code */
void init_parallel_ports()
{
    /* Clear the Port Pin Assignment Registers */
    PPARA = 0x00000000;
    PPARB = 0x00000000;
    PPARC = 0x00000000;
    PPARD = 0x00000000;
    /* Clear the Port Data Direction Registers */
    PDIRA = 0x00000000;
    PDIRB = 0x00000000;
    PDIRC = 0x00000000;
    PDIRD = 0x00000000;
    /* Program the Port Special Options Registers */
    PSORA = 0x00000000;
    PSORB = 0x00000004;
    PSORC = 0x030f0000;
    PSORD = 0x01000004;
    /* Program the Port Data Direction Registers */
    PDIRA = 0x00403fcd;
    PDIRB = 0x000003c5;
    PDIRC = 0x0b0f0000;
    PDIRD = 0x0100b004;
    /* Program the Port Open-Drain Registers */
    PODRA = 0x00000000;
    PODRB = 0x00000000;
    PODRC = 0x08000000;
    PODRD = 0x00000000;
    /* Program the Port Pin Assignment Registers */
    PPARA = 0x00ffffff;
    PPARB = 0x00003fff;
    PPARC = 0x030f0000;
    PPARD = 0x0100f004;
}

*****
/* Following is an example using the above initialization code. */
#include "m8101.h" /* IMMR definitions and declarations */
/* Global Declarations */
t_8101IMM *IMM; /* t_8101IMM is a structure for MSC8101 */
/* main internal memory map (IMM) */
/* Function Prototype(s) */
void init_parallel_ports(); /* Initializes the parallel I/O pins */

```

```

void Main()
{
    IMM = (t_8101IMM *) (0x14700000); /* Internal memory map (IMM) pointer
*/
    init_parallel_ports(); /* Initialize the parallel I/O pins */
}

/*-----
* FUNCTION NAME: init_parallel_ports
*
* DESCRIPTION:
*
*   Initializes the parallel I/O pins using the example above. Uses
*   settings generated by the Pin_mux8101 utility.
*
* EXTERNAL EFFECTS: Initializes relevant PIO registers.
*
* PARAMETERS: None
*
* RETURNS: None
*-----*/
void init_parallel_ports()
{
    /* Clear the Port Pin Assignment Registers */
    IMM->io_regs[PORT_A].ppar = 0x00000000;
    IMM->io_regs[PORT_B].ppar = 0x00000000;
    IMM->io_regs[PORT_C].ppar = 0x00000000;
    IMM->io_regs[PORT_D].ppar = 0x00000000;
    /* Clear the Port Data Direction Registers */
    IMM->io_regs[PORT_A].pdir = 0x00000000;
    IMM->io_regs[PORT_B].pdir = 0x00000000;
    IMM->io_regs[PORT_C].pdir = 0x00000000;
    IMM->io_regs[PORT_D].pdir = 0x00000000;
    /* Program the Port Special Options Registers */
    IMM->io_regs[PORT_A].psor = 0x00000000;
    IMM->io_regs[PORT_B].psor = 0x00000004;
    IMM->io_regs[PORT_C].psor = 0x030f0000;
    IMM->io_regs[PORT_D].psor = 0x01000004;
    /* Program the Port Data Direction Registers */
    IMM->io_regs[PORT_A].pdir = 0x00000000;
    IMM->io_regs[PORT_B].pdir = 0x00000000;
    IMM->io_regs[PORT_C].pdir = 0x00000000;
    IMM->io_regs[PORT_D].pdir = 0x00000000;
    /* Program the Port Open-Drain Registers */
    IMM->io_regs[PORT_A].podr = 0x00000000;
    IMM->io_regs[PORT_B].podr = 0x00000000;
    IMM->io_regs[PORT_C].podr = 0x08000000;
    IMM->io_regs[PORT_D].podr = 0x00000000;
    /* Program the Port Pin Assignment Registers */
    IMM->io_regs[PORT_A].ppar = 0x00ffffff;
    IMM->io_regs[PORT_B].ppar = 0x00003fff;
    IMM->io_regs[PORT_C].ppar = 0x030f0000;
    IMM->io_regs[PORT_D].ppar = 0x0100f004;
} /* end init_parallel_ports() */

```

## 5 References

- [1] *MSC8101 Reference Manual (MSC8101RM)*.

**NOTES:**

**NOTES:**

## **How to Reach Us:**

### **Home Page:**

[www.freescale.com](http://www.freescale.com)

### **E-mail:**

[support@freescale.com](mailto:support@freescale.com)

### **USA/Europe or Locations not listed:**

Freescale Semiconductor  
Technical Information Center, CH370  
1300 N. Alma School Road  
Chandler, Arizona 85224  
+1-800-521-6274 or +1-480-768-2130  
[support@freescale.com](mailto:support@freescale.com)

### **Europe, Middle East, and Africa:**

Freescale Halbleiter Deutschland GMBH  
Technical Information Center  
Schatzbogen 7  
81829 München, Germany  
+44 1296 380 456 (English)  
+46 8 52200080 (English)  
+49 89 92103 559 (German)  
+33 1 69 35 48 48 (French)  
[support@freescale.com](mailto:support@freescale.com)

### **Japan:**

Freescale Semiconductor Japan Ltd.  
Headquarters  
ARCO Tower 15F  
1-8-1, Shimo-Meguro, Meguro-ku,  
Tokyo 153-0064, Japan  
0120 191014 or +81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

### **Asia/Pacific:**

Freescale Semiconductor Hong Kong Ltd.  
Technical Information Center  
2 Dai King Street  
Tai Po Industrial Estate  
Tai Po, N.T. Hong Kong  
+800 2666 8080

### **For Literature Requests Only:**

Freescale Semiconductor Literature Distribution Center  
P.O. Box 5405  
Denver, Colorado 80217  
1-800-441-2447 or 303-675-2140  
Fax: 303-675-2150  
[LDCForFreescaleSemiconductor@hibbertgroup.com](mailto:LDCForFreescaleSemiconductor@hibbertgroup.com)

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. StarCore is a licensed trademark of StarCore LLC. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2000, 2005.