MXT3020

reference manual

version 4.0

Order Number: 100107-04
Revision C of the MXT3020

July 1999

Printed in the United States of America.

Contractor/manufacturer is:
Maker Communications, Inc.
73 Mount Wayte Avenue, Framingham, MA 01702

This manual supercedes and obsoletes the following Maker Communications publication:

100107-03 - MXT3020 Reference Manual, dated July 1998
100350-01 - MXT3020 Reference Manual, errata sheet, Sept/Oct 1998

CONTENTS

List of Figures

List of Tables

Preface

Maker Communications designs, develops, and markets a complete line of programmable, High-Intensity Communications Processors™, software solutions, and development tools. These products are designed to be adapted quickly and effectively to an extensive range of networking applications.

Maker Products

Maker Communications's product line includes:

- Traffic Stream Processing Solutions
- ATM Cell Processing Solutions

Traffic Stream Processing Solutions

Integrated Circuit

The **MXT4400 Traffic Stream Processor** is Maker Communications's very high-performance communications processing engine. Equally adept at supporting packets and ATM cells, this device resides in the data path of networking systems and provides wire-speed, programmable internetworking and traffic management. It is the first of a product family that supports OC-12 and OC-48 traffic loads. This programmable chip is adapted for a number of different applications using either Maker's firmware packages or customer developed applications.

Software Solutions

Executing on the MXT4400, **PortMaker™** employs a modular architecture built on a common kernel to deliver standard inter-domain services. These services include ATM Adaptation Layer 5 (AAL5) Segmentation and Reassembly (SAR) at 622 Mb/s and support for Packet over SONET.

Development Tools

Maker Communications offers a full suite of development tools for the MXT4400 Traffic Stream Processor. These include a Verilog model of the chip, the **TSP Assembler**, the MXT4400 simulated test bench (**ESIM**), and the graphical version thereof (**GESIM)**. The TSP Assembler assembles, links, and creates a program image that is downloaded to the MXT4400 during device initialization. ESIM is a Verilog-based simulator that provides a tightly controlled and fully observable environment to test and debug both processor applications and external host programs before running them on the target hardware. ESIM is complemented with a graphical post processor, GESIM.

ATM Cell Processing Solutions

Integrated Circuits

Maker Communications delivers a wide range of ATM solutions based on the **MXT3010** cell processing engine and the **MXT3020** circuit interface coprocessor. The MXT3010 is a high-performance programmable cell processor engine specifically designed to handle ATM cell manipulation and transmission at data rates up to 622 Mb/s. The MXT3020 is an ATM circuit interface coprocessor for the MXT3010 cell processor. It provides flexible interworking between Time Division Multiplexed (TDM) links and the ATM network.

Software Solutions

The MXT3010 and MXT3020 are complemented with a series of software applications that provide standard cell processing functionality. **CellMaker®-155** and **CellMaker®-622** execute on an MXT3010 and provide ATM Adaptation Layer 5 (AAL5) Segmentation and Reassembly (SAR) at data rates of 155 Mb/s and 622 Mb/s, respectively. **AccessMaker™** executes on an MXT3010 with up to four attached MXT3020 coprocessors. It provides cell processing functions for both packet and circuit interworking to support multiple services concurrently including AAL1, AAL5, IMA, and cell relay.

Development Tools

Maker Communications offers a full suite of development tools for the MXT3010 Cell Processor including Verilog models of the chips, the **AS3010** assembler, CellMaker Simulator (**CSIM**), and Graphical CellMaker Simulator (**GCSIM**). The AS3010 assembles, links, and creates a program image that is downloaded to the MXT3010 during device initialization. CSIM is a Verilog-based simulator that provides a tightly controlled and fully observable environment to execute and debug both processor applications and external host programs before running them on the target hardware. Maker also provides two development boards. CSIM is complemented with a graphical post processor, GCSIM. The **MXT3016** is a 32-bit, PCI bus-based development board used to test 622Mb/s applications. The **MXT3025** is a 32-bit, PCI bus-based evaluation board used to test OC-3 ATM (MXT3010) and T1 (MXT3020) applications.

Using this Manual

This manual is a reference document intended to assist in the design of communications systems. This manual is product specific and should be used in conjunction with other Maker product documentation to develop systems. For a complete listing of Maker documentation refer to the web site at http://www.maker.com/support.

Additional Document References

The information provided in Maker documentation assumes some reader familiarity with the following documents:

- ATM Forum's *UTOPIA Specification, Level 2, Version 1.0, af-phy-0039.000, dated June 1995*
- PCI Specification

Terminology

The glossary contained in this manual defines Maker terminology as well as networking/communications terminology applicable to Maker products.

Notes and Cautions

The manual uses these conventions:

Notes contain information that is incidental to the current subject matter.

Cautions warn of the risk of unfortunate system damage or loss of data.

Typographical Conventions

This document uses the following typographical conventions:

- API commands appear in mixed case, for example Open_Channel.
- Simulator commands appear in lower case, for example cdl_dump.
- Instruction mnemonics and registers appear in uppercase, for example the R_TSSCFG register.
- User input appears in **`bold monospace font`**.
- System output and code examples appear in `monospace font`.
- Variables that accompany commands or configuration switches appear in *<italics and in brackets>*, for example sim *<14000>*
- Text references to locations off of the current page appear in italics. For example, see *Maker Products on page xi*.
- For on-line viewers using ADOBE Acrobat PDF readers, all hypertext links appear in blue.

Change bars

Change bars are provided to indicate revisions made since the publication of the manual.

Contacting Support Services

Maker Communications has the following forums for getting information, communicating ideas, and reporting problems:

- Sales and customer support: 508-628-0622
- Customer support: support@maker.com
- Product inquiries: info@maker.com
- Facsimile: 508-628-0256
- Web: www.maker.com

Organization of this manual

This reference manual includes three sections: *Subsystems*, *Register Reference*, and *Signal Descriptions and Electrical Characteristics*. Also included are Appendix A, *Acronyms*, and Appendix B, *Registered Decoder PAL Source Code.*

The *Subsystems* section is a general tutorial on the organization and features of the MXT3020. Intended for readers unfamiliar with the chip, this section includes information on the:

- Circuit Interface
- Data Mover (Scatter/Gather) Units with List RAMs and Task Buffer RAMs
- Scatter and Gather Memory Interfaces
- Port 2 Interface

Section 2, *Register Reference*, is intended for experienced users of the MXT3020. This section condenses the material provided in the *Subsystems* section into tabular form, and provides the bit assignments and functions for all MXT3020 registers, which are listed in alphabetical order.

Section 3, *Signal Descriptions and Electrical Characteristics*, section includes information on:

- Timing information
- Pin out and pin listing
- Signal descriptions
- Electrical parameters
- Thermal characteristics
- Mechanical information

Section 1 Subsystems

This section is composed of five chapters. It provides an overview of the MXT3020 ATM circuit interface coprocessor and its major functional subsystems.

CHAPTER 1 *Device Overview*

The MXT3020 is a TDM circuit interface coprocessor for building systems that integrate ATM, packets, and TDM – including voice-over-ATM products. Together with the MXT3010, the MXT3020 delivers the performance and features to support multiple services concurrently, including structured and unstructured AAL1 circuit emulation, ATM User-Network Interface (UNI), Inverse Multiplexing for ATM (IMA), AAL2, and TM4.0 compliant AAL5. With up to four MXT3020s connected to a single MXT3010, systems can be developed that support any combination of up to 32 T1, E1, or J2 links through standard framers, or up to 64 bidirectional links (up to 8 Mb/s each) supporting SCSA and MVIP computer telephony buses.

MXT3020 applications include:

- Circuit emulation service for converting TDM data into AAL1 cells in both Structured Data Transfer (SDT) and Unstructured Data Transfer (UDT) modes.
- ATM User-Network Interface support and inverse multiplexing over TDM circuits including T1, E1, and J2.

Key features of the MXT3020 include:

- A glueless connection to the MXT3010. The MXT3010's SWAN processor programs and controls the MXT3020.
- Support for structured and unstructured data transfer modes on 16 bidirectional TDM wires that directly support SCSA and MVIP telephony busses, or 8 unidirectional TDM wire pairs that connect to T1, E1, or J2 framers.

- Data Mover Units that support fully arbitrary mapping of DS0s to cell payloads including cross link mapping for use in trunking applications. The Data Mover Units contain cell delineation, Header Error Control (HEC) generation and checking, plus cell payload scrambling and descrambling hardware to support T1 and E1 User-Network Interfaces.
- Support for Nx64 mode with up to 2048 DS0's of traffic with up to 1024 virtual circuits.
- Internal clock recovery using Synchronous Residual Time Stamp (SRTS) or Adaptive techniques.

Figure 1 shows a block diagram of a possible application using the MXT3020 as a coprocessor for the MXT3010 to provide an interface between ATM and T1/E1 facilities.

FIGURE 1. T1/E1/ATM Interface Using the MXT3010 and MXT3020

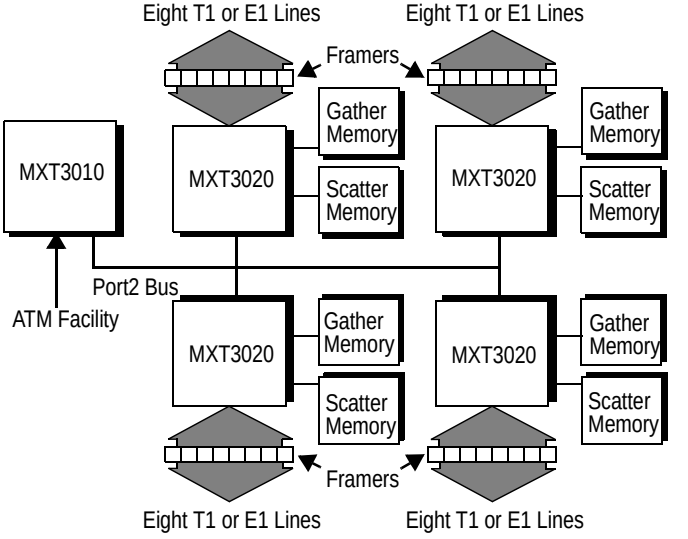
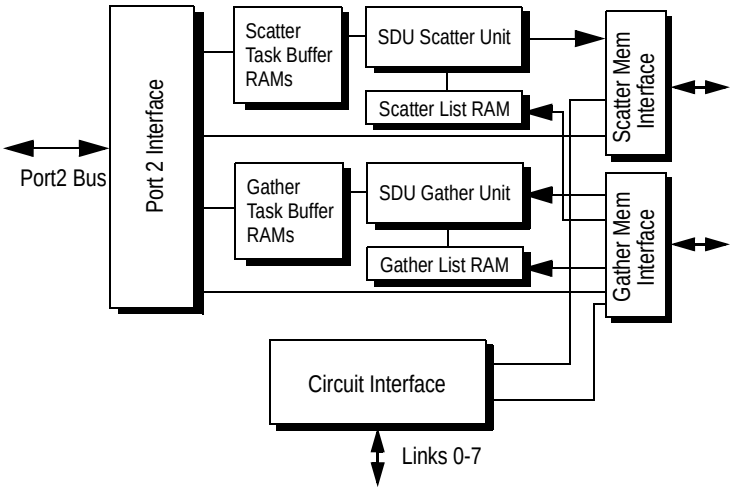


Figure 2 shows a block diagram of the MXT3020.

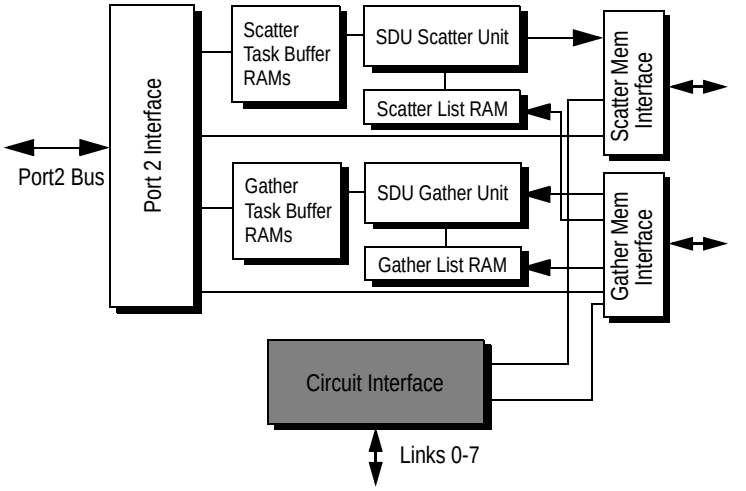
FIGURE 2. Block diagram of the MXT3020



The major sections of the MXT3020 are the:

- Circuit Interface
- Data Mover (Scatter/Gather) Units with List RAMs and Task Buffer RAMs
- Scatter and Gather Memory Interfaces
- Port 2 Interface

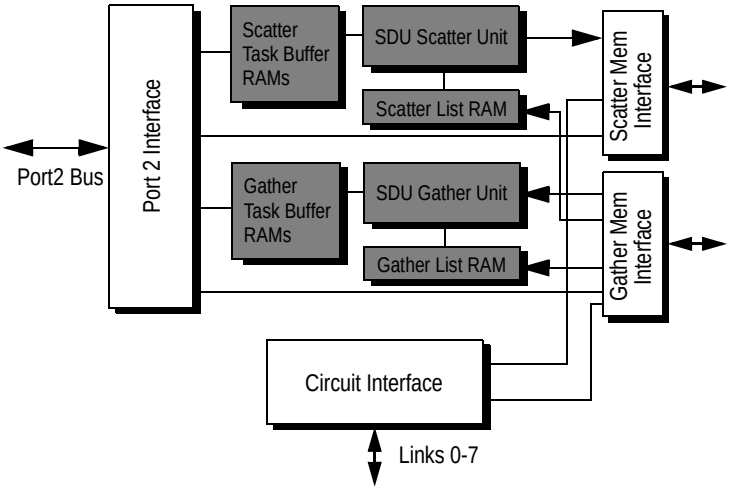
CIRCUIT INTERFACE



The Circuit Interface receives and transmits TDM serial data over eight serial link pairs.

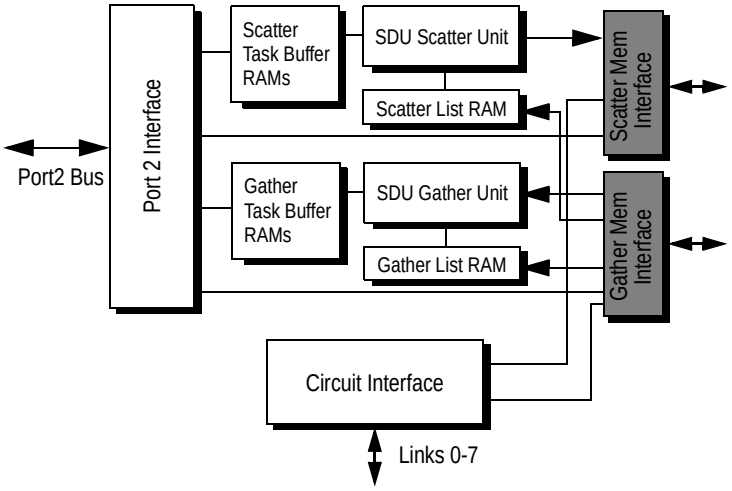
- Each link pair supports structured and unstructured data transfer (SDT and UDT) modes, and is independent with its own clocks and unidirectional mode frame synchronization.
- Each link pair can be configured in a unidirectional mode suitable for use with framers or in a bidirectional mode compatible with telephony busses. When configured in unidirectional mode, each link pair implements one receive line and one transmit line. When configured in bidirectional mode, each link of the pair implements an independent bidirectional line.
- For each link, a tri-state control map is provided. The MXT3020 steps through the tri-state control map as each DS0 occurs, enabling or disabling the tri-state output enable on the link as directed by the control bits in the map.
- Received TDM frames are written into buffers in Gather Memory. Transmit TDM frames are read from buffers in Scatter memory.

THE DATA MOVER (SCATTER/GATHER) UNITS



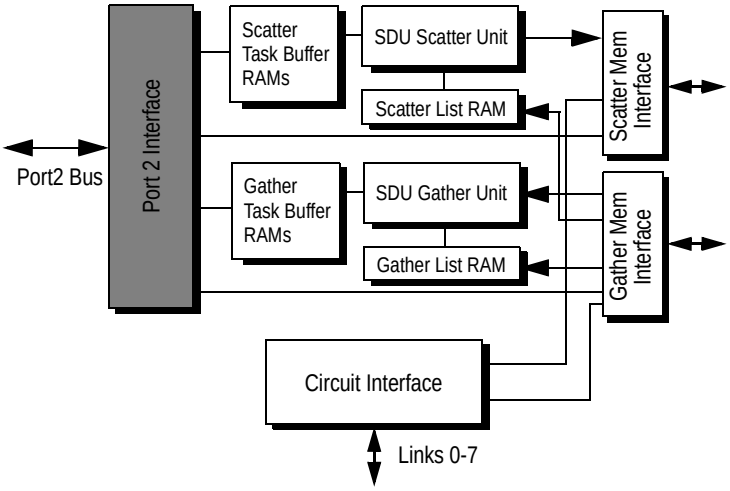
The Data Mover Units (DMUs) are specialized scatter/gather machines. One DMU disassembles SAR Service Data Units (SDUs) from ATM cells and prepares data for transmission over TDM links. Simultaneously, the other DMU receives data from TDM links and assembles it into ATM SAR SDUs. The Data Mover Units are programmed by the MXT3010, which loads lists of instructions for controlling the data transfer process into a designated area of Gather Memory. The MXT3010 then loads control information and data into the Task Buffer RAM of the scatter machine or control information into the Task Buffer RAM of the gather machine.

SCATTER AND GATHER MEMORY INTERFACES



The Scatter and Gather Memory interfaces are each 16-bits wide and support up to 512 Kbytes of pipelined synchronous SRAM. Each port can accept accesses from any of the on-chip masters (Circuit Interface, Data Mover Units, or Port2 Interface). Because the Scatter and Gather Memory ports are independent, operations to these ports can occur simultaneously.

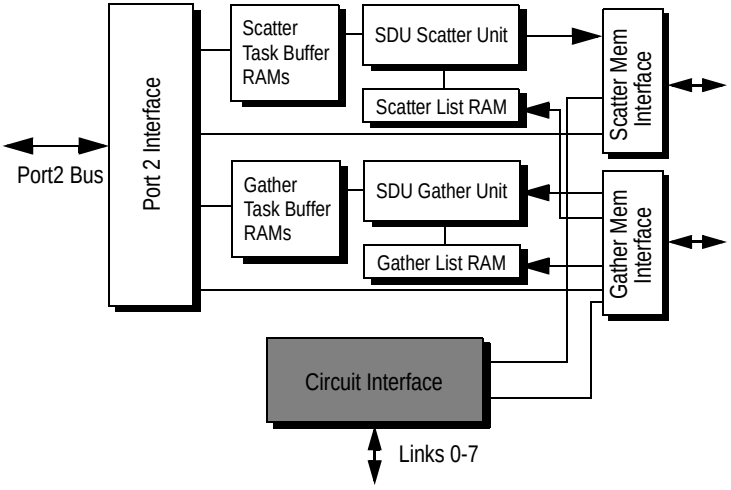
PORT2 INTERFACE



The Port2 Interface is the MXT3020's connection to the MXT3010 device. Through the Port2 Interface, the MXT3010 can read and write Scatter and Gather memory, the Task Buffer RAMs, List RAMs, and all internal registers. See "MXT3020 address space" on page 83. Once a Port2 transfer begins, the MXT3020's Port2 Interface demultiplexes the Port2 bus and decodes the address. If the address maps to this chip, the MXT3020 performs the read or write operation to the register or RAM selected by the address. The MXT3020 address space is accessible only in burst transfer mode.

An MXT3020 occupies 1 Mbyte of Port2 address space. Two MXT3020s can reside on a Port2 bus without the need for additional device selection logic. The P2A20 pin of one MXT3020 is strapped low to decode Port2 bus address space 0x000000 to 0x0FFFFF while the P2A20 pin of the other MXT3020 is strapped high to decode Port2 address space 0x100000 to 0x1FFFFF. Alternatively, multiple MXT3020's can be supported with the addition of a device selection PAL. See "Multiple MXT3020 implementation" on page 130.

CHAPTER 2 *Circuit Interface*



The Circuit Interface logic receives and transmits TDM serial data. The Circuit Interface supports eight serial link pairs.

- Each link pair supports structured and unstructured data transfer (SDT and UDT) modes.
- Each link pair is completely independent with its own clocks and unidirectional mode frame synchronization.
- TDM serial data rates supported are 1.544, 2.048, 4.096, 6.144, or 8.192 Mhz.

- Each link pair can be configured in a unidirectional or a bidirectional mode. In unidirectional mode, each link pair is configured as one 2-wire port with a dedicated receive and transmit line, and can connect to T1, E1, and J2 framers. In bidirectional mode, each link pair is configured as two single-wire bidirectional links, and can directly support SCSA and MVIP telephony busses.
- A tri-state control map for each link can be stored in a section of the Gather Memory reserved for control purposes. The size of the tri-state control map for a link is determined by the number of DS0s occurring in each frame, as a tri-state control bit is provided for each DS0 on the link. Synchronized by Frame Sync on the link, the MXT3020 steps through the tri-state control map as each DS0 occurs, enabling or disabling the tri-state output enable on the link as directed by the tri-state control bits in the map. In bidirectional mode, each wire is individually tri-statable on a DS0 by DS0 basis. In unidirectional mode, each transmit wire is individually tri-statable on a DS0 by DS0 basis.
- Serial data received by the links is packed into halfwords and written to buffers in Gather Memory. Data to be transmitted is read from Scatter Memory and shifted out onto the transmit links. The Circuit Interface maintains all necessary pointers to Scatter/Gather Memory to support constant-bit-rate data with minimal intervention from the SWAN processor in the MXT3010.

CIRCUIT INTERFACE REGISTERS

Per-Link registers

Each of the eight link pairs has a configuration register, three address counters, and five link service clock and Synchronous Residual Time Stamp (SRTS) control registers:

TABLE 1. Circuit Interface registers (for each link)

Register	*Function*
Link Configuration register	See “Link Configuration register” on page 12.
Link Tx Buffer Address counter	See “Link Buffer Address counters” on page 23.
Link Rx Buffer Address counter	
Link Tri-state Control Address counter	See “Link Tri-state Control Address counter” on page 27.
Link service clock control registers: N register, K register, L counter, FTC counter, Link SRTS Value register	See “Link service clock generation registers” on page 33.

Global registers

In addition to the eight registers provided for each link, the Circuit Interface has six global configuration/status registers:

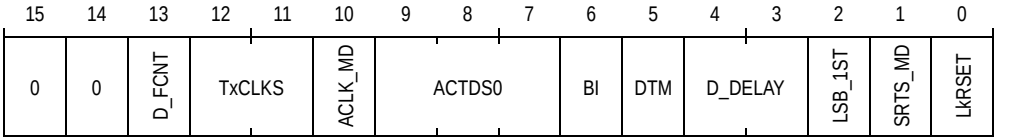
TABLE 2. Circuit Interface registers (global)

Register	*Function*
CI Tri-state Control Base Address register	See “CI Tri-state Control Base Address register” on page 37.
CI Configuration register	See “CI Configuration register” on page 37.
CI Quiet Frame Base Address register	See “CI Quiet Frame Base Address register” on page 42.
CI SRTS FTC register	See “CI SRTS FTC register” on page 45.
CI SRTS Valid Status register	See “CI SRTS Valid Status register” on page 45.
CI Status register	See “CI Status register” on page 46.

LINK CONFIGURATION REGISTER

The Link Configuration register controls the operating mode of each link. There is one of these read/write registers for each link. The bit map is shown in Figure 3 and the bit functions are described in the paragraphs which follow.

FIGURE 3 Link Configuration register bit assignments



D_FCNT (bit 13)

The Link Buffer Address counters (page 23) are 15 bits long and consist of a DS0 portion and a frame count portion. The number of bits used in the DS0 portion depends upon the number of DS0's in the frame (MAXDS0 in the "CI Configuration register" on page 37). The bits not used for the DS0 portion are the frame count portion. The D_FCNT bit enables/disables incrementing the frame count portion. Setting this bit to one (1) disables incrementing the frame count portion, causing programs used in special applications to repeatedly use a single portion of frame buffer storage. Clearing this bit to zero (0) enables incrementing the frame count portion. This latter choice is used by most programs.

TxCLKS (bits 12 and 11)

These bits select the link transmitter clock source (SRTS, BCLK, ACLK). See "Added detail: Bits 12-10 – clock source control" on page 15..

Bit 12	*Bit 11*	*Clock Source*
0	0	Internal SRTS Clock
0	1	External BCLK input
1	0	External ACLK input
1	1	Reserved

ACLK_MD (bit 10)

This bit selects the ACLK pin I/O direction. When this bit is one (1), the ACLK pin is an output. When this bit is zero (0), the ACLK pin is an input. See “Added detail: Bits 12-10 – clock source control” on page 15.

ACTDS0 (bits 9, 8, and 7)

These bits indicate the actual number of DS0's per frame. This information is used by the tri-state enable control map (see “The tri-state enable control map” on page 28).

Bit 9	*Bit 8*	*Bit 7*	*Actual number of DS0’s per frame*
0	0	0	24
0	0	1	32
0	1	0	64
0	1	1	96
1	0	0	128
101 through 111			Reserved

BI (bit 6)

This bit controls the mode of a link pair. When this bit is one (1), the link operates in bidirectional mode. When this bit is zero (0), the link operates in unidirectional mode. These modes are described in greater detail in “Added detail: Bit 6 – link pair mode” on page 19.

DTM (bit 5)

This bit selects the Data Transfer Mode of the link pair. When this bit is one (1), the link operates in Structured Data Transfer (SDT) mode. When this bit is zero (0), the link operates in Unstructured Data Transfer (UDT) mode.

D_DELAY (bits 4 and 3)

These bits control the position of Frame Sync relative to the first bit of a frame.

Bit 4	*Bit 3*	*Frame Sync to first data delay*
0	0	1 cycle
0	1	2 cycles
1	0	3 cycles
1	1	Reserved

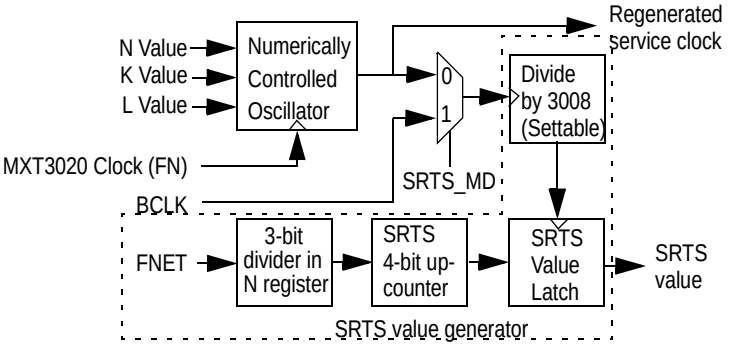
LSB_1ST (bit 2)

This bit controls the direction of the link shift registers. When this bit is one (1), the link shift registers operate in LSB mode, shifting the least significant bit of each byte onto the serial line first. When this bit is zero (0), the link shift registers operate in MSB mode, shifting the most significant bit of each byte onto the serial line first.

SRTS_MD (bit 1)

This bit controls the mode of the SRTS value generator. Figure 4 shows a simplified diagram of the SRTS support hardware.

FIGURE 4 SRTS support hardware



When this bit is one (1), the SRTS value generator operates in master mode. In this mode, the value generator uses the externally supplied TDM service clock (BCLK pin) to generate SRTS values for transmission over the ATM link with the data.

When the SRTS_MD bit is zero (0), the SRTS value generator operates in slave mode. In slave mode, the SRTS value generator measures the service clock generated by the numerically controlled oscillator within the MXT3020. When SRTS is used for clock recovery, the SRTS values thus generated are used by software to adjust the values of N and K and lock the regenerated TDM service clock to the remote TDM service clock.

LkRESET (bit 0)

This bit controls the reset state of a link pair. Setting this bit to one (1) places the link in the reset state. Clearing this bit to zero (0) removes the link from the reset state.

Added detail: Bits 12-10 – clock source control

ACLK pin

In addition to the ASER and BSER pins, an ACLK pin is provided. The state of bit 10 (ACLK_MD) in the Link Configuration register determines whether this pin is used as an input or as an output:

- If ACLK_MD is clear (0), the ACLK pin functions as an input and functions as the Transmit Input Clock in both the bidirectional (bus) and unidirectional (framer) modes.
- If ACLK_MD is set (1), the ACLK pin functions as an output and functions as the Transmit Link (Output) Clock in the unidirectional (framer) mode. In bidirectional (bus) mode, the tri-state control that is being used to control the ASER line is presented on the ACLK pin and can be used to control external output-enabled devices.

BCLK pin

The BCLK pin provides the Receive Clock signals for both ASER and BSER in both the unidirectional and bidirectional modes.

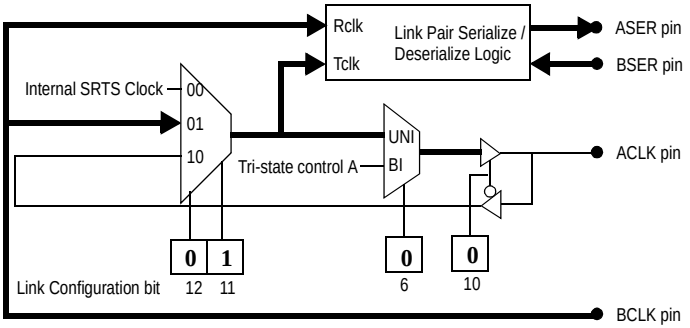
Table 3 shows four configurations using ACLK and BCLK to provide clocking in unidirectional mode. Table 3 also shows three configurations using ACLK and BCLK to provide clocking in bidirectional mode. The figures cited in the table provide additional information about the clocking configurations.

TABLE 3. Configurations for ACLK and BCLK

Bits 12,11,10,6	***Description***	***Figure***
0100	(Unidirectional mode) An external clock on the BCLK pin provides the clocking for both the link logic transmitter and receiver. The ACLK pin is unused.	Figure 5, "Configuration Uni-1," on page 17
1000	(Unidirectional mode) An external clock on the ACLK pin provides clocking for the link logic transmitter. An external clock on the BCLK pin provides clocking for the link logic receiver.	Figure 6, "Configuration Uni-2," on page 17
0010	(Unidirectional mode) The internal SRTS clock provides clocking for the link logic transmitter and outputs this clock on the ACLK pin. An external clock on the BCLK pin provides clocking for the link logic receiver.	Figure 7, "Configuration Uni-3," on page 18
0110	(Unidirectional mode) An external clock on the BCLK pin provides the clocking for both the link logic transmitter and receiver and provides the same clock signal as an output on the ACLK pin.	Figure 8, "Configuration Uni-4," on page 18
0101	(Bidirectional mode) An external clock on the BCLK pin provides the clocking for both the link logic transmitter and receiver. The ACLK pin is unused.	Figure 9, "Configuration Bi-1," on page 18
1001	(Bidirectional mode) An external clock on the ACLK pin provides clocking for the link logic transmitter. An external clock on the BCLK pin provides clocking for the link logic receiver.	Figure 10, "Configuration Bi-2," on page 19
0111	(Bidirectional mode) An external clock on the BCLK pin provides the clocking for both the link logic transmitter and receiver. The tri-state control that is being used to control the ASER line is presented on the ACLK pin and can be used to control external output-enabled devices.	Figure 11, "Configuration Bi-3," on page 19

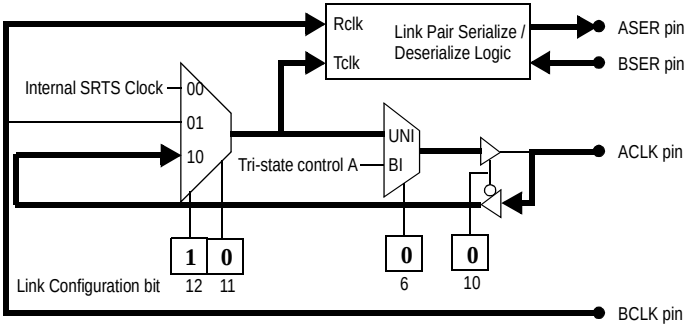
Note: While sixteen configurations are theoretically possible (four bits are used), only the seven shown in the figures are functional. Other configurations are non-functional and must not be used.

FIGURE 5 Configuration Uni-1



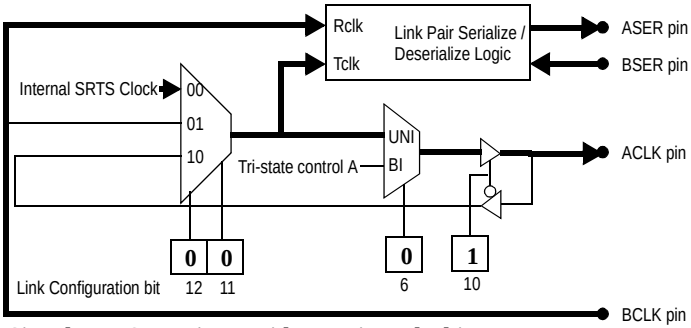
Signal on BCLK pin provides all clocking

FIGURE 6 Configuration Uni-2



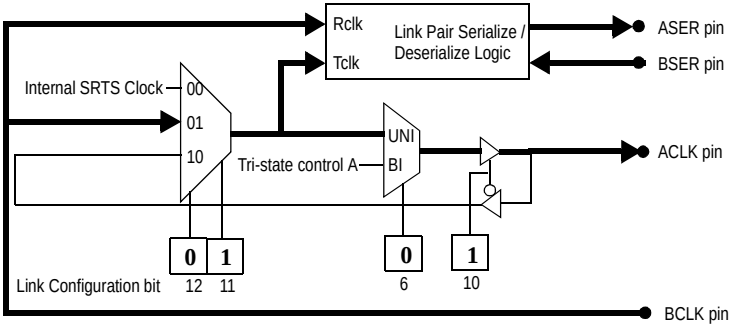
Signal on BCLK pin provides receiver clocking
Signal on ACLK pin provides transmitter clocking

FIGURE 7 Configuration Uni-3



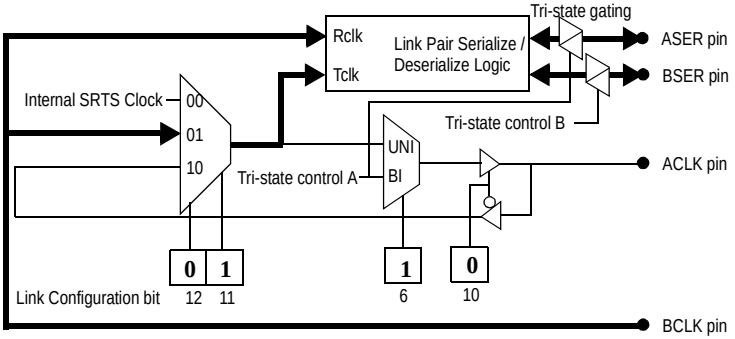
Signal on BCLK pin provides receiver clocking
Internal SRTS provides transmit clocking and output on ACLK

FIGURE 8 Configuration Uni-4



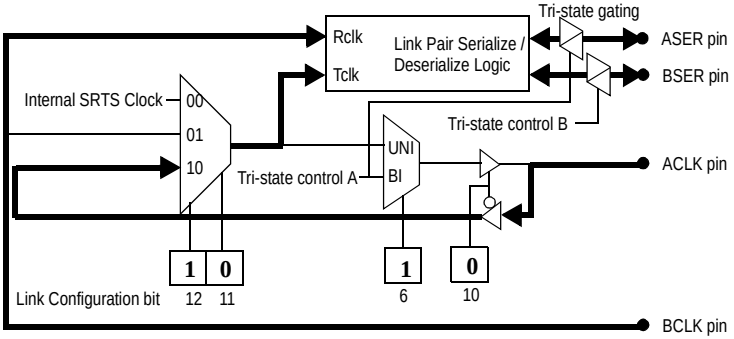
Signal on BCLK pin provides all clocking and output on ACLK

FIGURE 9 Configuration Bi-1



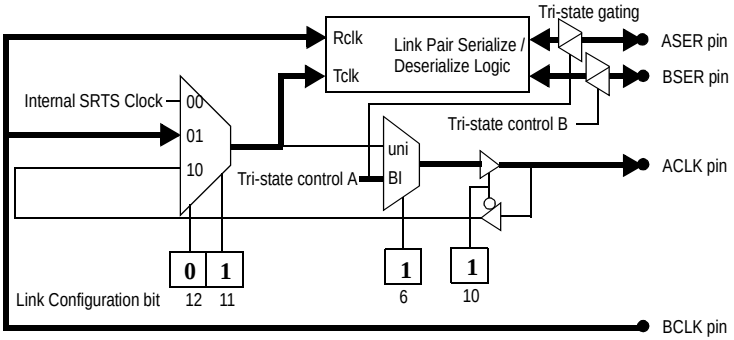
Signal on BCLK pin provides all clocking

FIGURE 10 Configuration Bi-2



Signal on BCLK pin provides receiver clocking
Signal on ACLK pin provides transmitter clocking

FIGURE 11 Configuration Bi-3

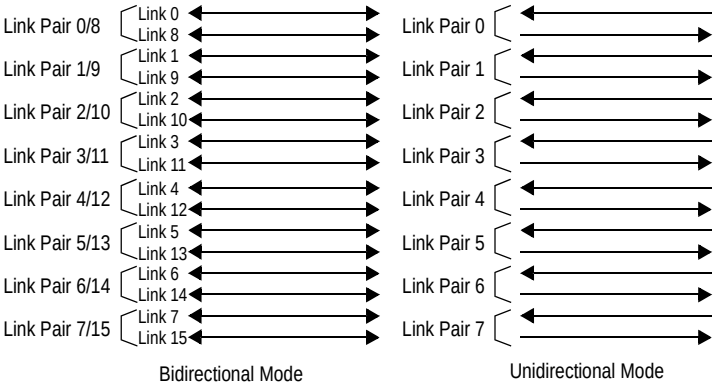


Signal on BCLK pin provides all clocking
Tri-state control for external gating is provided on ACLK pin

Added detail: Bit 6 – link pair mode

The MXT3020 circuit interface can be operated in either bidirectional or unidirectional mode. The BI bit (bit [6]) in the Link Configuration register selects the mode of operation. Bidirectional mode is typically used with telephony busses, and unidirectional mode is typically used with TDM framers. Figure 12 shows the two modes.

FIGURE 12 MXT3020 operating modes - eight link pairs



As indicated in the figure, the MXT3020 treats the TDM links as eight link pairs, regardless of operating mode. Thus, the two operating modes can be compared by analyzing a single link pair.

Pins associated with a link pair

There are five pins associated with each link pair. Depending upon the configuration of the link, these pins may be connected to a variety of different signals within the MXT3020. Table 4 summarizes the connection possibilities, and the paragraphs which follow discuss each pin and its configuration alternatives in greater detail.

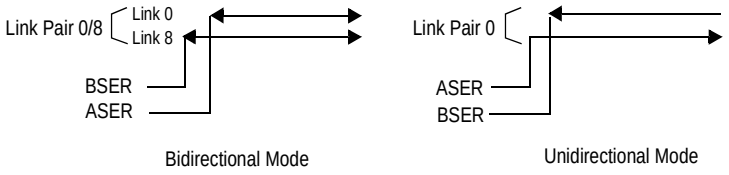
TABLE 4. MXT3020 link interface pins and signals

Pin	*Signals carried in unidirectional mode*	*Signals carried in bidirectional mode*
ASER	TxData (“A Serial”) for link *n*	Tx/RxData for link *n*
BSER	RxData (“B Serial”) for link *n*	Tx/RxData for link *n* + 8
ACLK (output) (input)	 Tx Link Clk for ASER Tx Input Clk for ASER/BSER	 Copy of tri-state enable for ASER Tx Input Clk for ASER/BSER
BCLK	RxCLK for ASER/BSER	RxCLK for ASER/BSER
BSCE	Frame Sync	Copy of tri-state enable for BSER

ASER and BSER pins

Figure 13 shows the ASER (A serial) and BSER (B serial) pins. As indicated in Table 4, in the unidirectional mode, ASER and BSER are TxData and RxData respectively. In bidirectional mode, ASER and BSER carry Tx/RxData. Since ASER and BSER are bidirectional busses in that mode, the transmit functions of ASER and BSER need to be disabled when received data is arriving. Enabling and disabling of the transmit function are accomplished by tri-state enable circuitry described in “Link Tri-state Control Address counter” on page 27.

FIGURE 13 MXT3020 operating modes - ASER and BSER



Added detail: BSCE pin

As indicated in Table 4 on page 20, the BSCE pin can be used either as a Frame Sync pin, or as a copy of the tri-state enable for the BSER pin. Usage of the BSCE pin depends upon the MXT3020 operating mode and is explained below.

BSCE usage in unidirectional mode

When the MXT3020 is used in unidirectional mode, it is typically connected to TDM framers that provide Frame Sync signals. Two methods of wiring the Frame Sync signal are available:

- If a single Frame Sync signal is used, it can be connected to the FSYNC pin (pin 160) of the MXT3020. In addition, the FS_MODE bit (bit [0] of the CI Configuration register) should be cleared to zero (0) to indicate that a Frame Sync signal common to all links is being supplied on the FSYNC pin.

- If the application calls for different link pairs to use different Frame Sync signals, Frame Sync signals can be wired to the BSCE pins of individual link pairs. In addition, the FS_MODE bit (bit [0] of the CI Configuration register) should be set to one (1) to indicate that Frame Sync signals for each link pair are being provided on their respective BSCE pins.

In addition to the actions listed above, bits [4:3] of the Link Configuration register and bit [10] of the CI Configuration register are used to control the use of Frame Sync in various applications.

BSCE usage in bidirectional mode

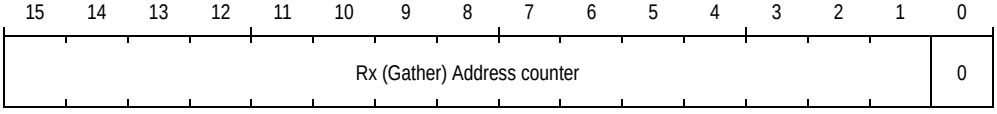
Bidirectional mode is used with telephony busses, where per-link Frame Sync signals are not required. Therefore, the BSCE pin is not used for Frame Sync in bidirectional mode. In bidirectional mode, the tri-state control that is being used to control the BSER line is presented on the BSCE pin and can be used to control external output-enabled devices.

LINK BUFFER ADDRESS COUNTERS

Link Rx Buffer Address counter

This register indicates where the next data received on this TDM link pair will be stored in the Gather Memory. The contents of this counter combined with the link number provide an 18-bit pointer to a halfword-aligned circular queue in Gather Memory where received data can be written by the Circuit Interface logic. Every link pair has a separate queue for received data, thus each link pair has one of these read/write registers. The counters are initialized by the SWAN processor in the MXT3010 and are incremented automatically by the link hardware during data transfers.

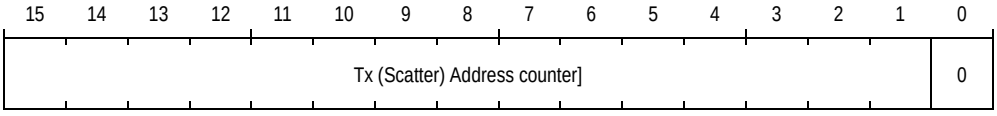
Restriction on Link Rx Buffer Address counter
Values in this register can only be modified when the link is in the reset state. This restriction does not affect the automatic incrementing that the MXT3020 performs during data transfers.



Link Tx Buffer Address counter

This register indicates the location in Scatter Memory from which the next data to be transmitted on the TDM link will be taken. The contents of this counter combined with the link number provide an 18-bit pointer to a halfword-aligned circular queue in Scatter Memory where transmit link data can be read by the Circuit Interface logic. There is a separate transmit data queue for every link pair; thus, there is one of these read/write registers for each link pair. The counters are initialized by the SWAN processor in the MXT3010 and are incremented automatically by the link hardware during data transfers.

Restrictions on *Link Tx Buffer Address counter*
Values written to this register must be 4 higher than those written to the Link Rx Buffer Address counter. Values in this register can only be modified when the link is in the reset state. These restrictions do not affect the automatic incrementing that the MXT3020 performs during data transfers.



Details of the Link Tx/Rx Buffer Address counters

Each address counter is composed of a frame counter (FC) and a DS0 counter. The DS0 counter wraps based upon the settings of the ACTDS0 field in the "Link Configuration register" on page 12. The frame counter wraps based on the MAXDS0 and MAXTBS/MAXRBS fields in the "CI Configuration register" on page 37.

The MAXDS0 and MAXTBS/MAXRBS settings determine the buffer size allocated to each link. This is shown in Table 5.

TABLE 5. Buffer size allocation

MAXDS0	*MAXTBS/MAXRBS*	*Buffer size (Wrap boundary)*
24 or 32	64 frames	2K
64	64 frames	4K
96 or 128	64 frames	8K
24 or 32	128 frames	4K
64	128 frames	8K
96 or 128	128 frames	16K
24 or 32	256 frames	8K
64	256 frames	16K
96 or 128	256 frames	32K
24 or 32	512 frames	16K
64	512 frames	32K
96 or 128	512 frames	64K[a]

a. The 64K buffer size is not supported in bidirectional mode.

The positions of the frame counter and DS0 counter portions of the Link TX/RX Buffer Address counters are shown in Table 6.

TABLE 6. Frame Counter / DS0 Counter

	Unidirectional mode		***Bidirectional mode***	
MAXDS0	***Frame counter***	***DS0 counter***	***Frame counter***	***DS0 counter***
24 or 32	[*y*:5]	[4:0]	[*y*:6]	[5:1]
64	[*y*:6]	[5:0]	[*y*:7]	[6:1]
96 or 128	[*y*:7]	[6:0]	[*y*:8]	[7:1]

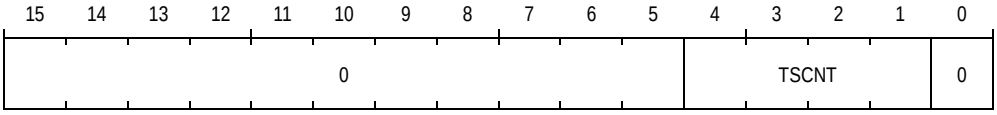
In Table 6, variable *y* represents the highest order bit used by the frame counter. The value of *y* depends upon the buffer size, as shown in Table 7.

TABLE 7. Relation of buffer size to variable *y*

Buffer size	***Unidirectional mode, y=***	***Bidirectional mode, y=***
2K	10	11
4K	11	12
8K	12	13
16K	13	14
32K	14	15
64K	15	Not supported

LINK TRI-STATE CONTROL ADDRESS COUNTER

The contents of the Tri-state Control (TSC) Address counter are combined with the Tri-state Control Base Address register and the Link ID number (LID) to create an 18-bit pointer to a halfword-aligned entry in Gather Memory containing the tri-state control map for the link. The format of the Tri-state Control Address counter is shown below. The format of the complete pointer is shown in “Creation of the tri-state control map pointer” on page 28.

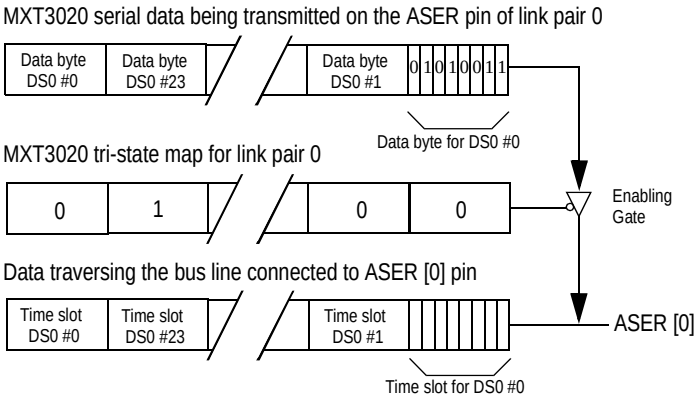


An example of the tri-state control process

The MXT3020 provides tri-state control of ASER [7:0] and BSER [7:0] when operating in bidirectional mode. The MXT3020 also provides tri-state control of ASER [7:0] when operating in unidirectional mode. Tri-state control is provided by tri-state control maps located in Gather Memory.

Figure 14 shows, in simplified form, the fundamentals of tri-state control. To simplify the figure, only the tri-state control of ASER [0], operating in bidirectional mode, is shown.

FIGURE 14 Tri-state enable for the ASER pins



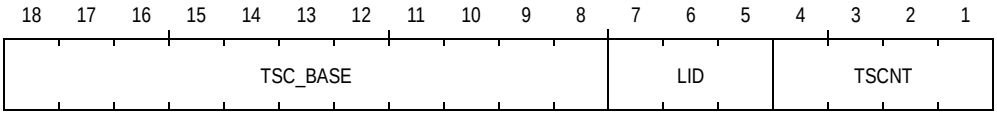
In the example shown, data is being generated in T1 format. That is, a byte of data is presented for a 64 Kbps channel (DS0); then a byte of data is presented for the next DS0. When data has been presented for 24 DS0's, a new frame begins and a new byte of data for the first DS0 is presented. The data being presented is only enabled onto the ASER[0] pin if the bit in the tri-state map for this link indicates that enabling during the time period of this DS0 is permitted. Specifically, a zero (0) in the tri-state map indicates that the data should be enabled onto the ASER [0] pin. Thus, in the example, data is being applied to DS0 #0 and DS0 #1, but not to DS0 #23.

If one thinks of the registers shown in Figure 14 as shift registers, it is important to note that the tri-state map "shifts" at one-eighth the rate of the data registers, as a bit in the tri-state map controls the enabling or disabling of an entire byte of data. For each byte time (at a DS0 rate) on each link, the MXT3020 provides two tri-state enable bits, one for ASER and one for BSER.

The tri-state enable control map

Creation of the tri-state control map pointer

The contents of the Tri-state Control Address counter (page 27) are combined with the Tri-state Control Base Address register (page 37) and the Link ID number (LID) to create an 18-bit pointer to a halfword-aligned entry in Gather Memory containing the tri-state control map for the link. The pointer has the following format:



Bidirectional mode

The tri-state control map pointer identifies eight buffers (LID [7:5]) in memory, one buffer for each link pair. Each buffer consists of 16 halfwords (TSCNT [4:1]) or 256 bits. For each link pair, during each DS0 data byte, the MXT3020 reads two of these tri-state enable bits, one for ASER and one for BSER. Figure 15 shows the sequence in which these enable bits are used.

FIGURE 15 ASER and BSER tri-state enable bit usage[1] (bidirectional)

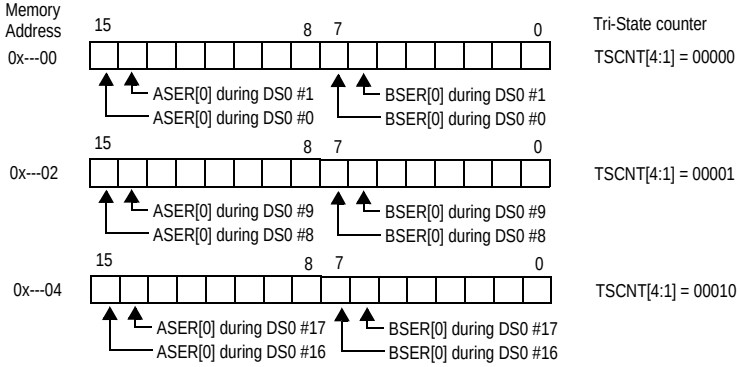


Figure 15 shows a tri-state map that would be sufficient for 24 DS0's on one link pair. By changing the value of LID and using the same three values for TSCNT, similar maps could be loaded into memory to control 24 DS0's on each of the eight link pairs served by the MXT3020.

The tri-state control map shown in Figure 15 serves 24 DS0's and uses 48 bits. The MXT3020 can accommodate tri-state control maps with as many as 128 DS0's (256 bits) for each link pair.

1. In all MXT3020 figures, memory addressing is Big Endian.

If the actual number of DS0s is less than 128, there is no requirement that the tri-state control map be filled with 256 bits, as the Tri-state Control Address counter wraps based upon the actual number of DS0s indicated in the ACTDS0 bits in the Link Configuration register. The wrap boundaries for the Tri-state Address counter are indicated in the following table:

Mode	*ACTDS0*	*Wrap Boundary*
Bidirectional	24	6 bytes
Bidirectional	32	8 bytes
Bidirectional	64	16 bytes
Bidirectional	96	24 bytes
Bidirectional	128	32 bytes

Unidirectional mode

The contents of the Tri-state Control Address counter (page 27) are combined with the Tri-state Control Base Address register (page 37) and the Link ID number (LID) to create an 18-bit pointer to a halfword-aligned entry in Gather Memory containing the tri-state control map for the link.

The tri-state control map pointer (page 28) identifies eight buffers (LID [7:5]) in memory, one buffer for each link pair. Each buffer consists of eight halfwords (TSCNT [3:1])[1] or 128 bits. For each link pair, during each DS0 data byte, the MXT3020 reads one of these tri-state enable bits to control the enabling of data onto the ASER pin. Figure 16 shows the sequence in which these enable bits are used.

1. TSCNT[4] is not used in unidirectional mode.

FIGURE 16 ASER and BSER tri-state enable bit usage (unidirectional)

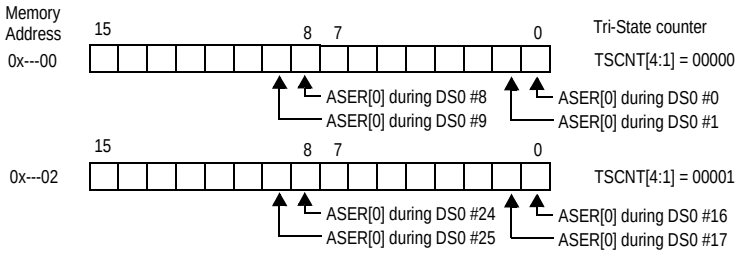


Figure 16 shows a tri-state map that would be sufficient for 32 DS0's (E1) on one link pair. By changing the value of LID and using the same two values for TSCNT, similar maps could be loaded into memory to control 32 DS0's on each of the eight link pairs served by the MXT3020.

The tri-state control map shown in Figure 16 serves 32 DS0's and uses 32 bits. The MXT3020 can accommodate tri-state control maps with as many as 128 DS0's (128 bits) for each link pair.

If the actual number of DS0s is less than 128, there is no requirement that the tri-state control map be filled with 128 bits, as the Tri-state Control Address counter wraps based upon the actual number of DS0s. The actual number of DS0s is determined by the ACTDS0 bits in the Link Configuration register, and the wrap boundaries for the Tri-state Address counter are indicated in the following table:

Mode	***ACTDS0***	***Wrap Boundary***
Unidirectional	24	4 bytes
Unidirectional	32	4 bytes
Unidirectional	64	8 bytes
Unidirectional	96	12 bytes
Unidirectional	128	16 bytes

Summary of tri-state control operation

The MXT3020 provides tri-state control of ASER [7:0] and BSER [7:0] when operating in bidirectional mode. The MXT3020 also provides tri-state control of ASER when operating in unidirectional mode. To set up a tri-state map, software selects a TSC Base Address, and for each link pair (LID) provides up to 16 (bidirectional mode) or 8 (Unidirectional mode) halfwords specifying the tri-state enable map.

During operation, the MXT3020 services each link pair sequentially. For each link, the MXT3020 maintains a Tri-state Control Address counter (TSCNT) based on the receipt of Frame Sync by the link. Thus, for each link (LID), the MXT3020 has an indication (TSCNT) of which DS0 is being handled by that link. The MXT3020 uses this information (LID and TSCNT), in conjunction with the TSC Base Address, to obtain the tri-state control map appropriate to the DS0 being serviced by that link at that time. In bidirectional mode, two tri-state enable bits from the map are used to control the gating of bytes of DS0 data onto the ASER and BSER pins respectively (see Figure 14 and Figure 15). In unidirectional mode, a tri-state enable bit from the map is used to control the gating of a byte of DS0 data onto the ASER pin (see Figure 14 and Figure 16).

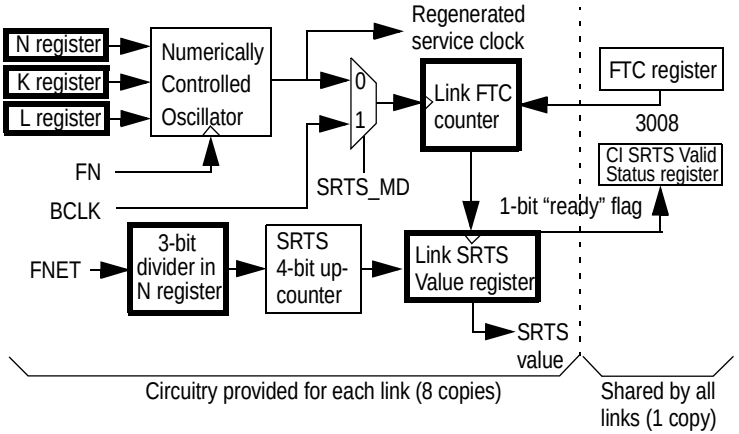
LINK SERVICE CLOCK GENERATION REGISTERS

For each link, there are five registers used for link service clock generation:

- Link Service Clock N register
- Link Service Clock K register
- Link Service Clock L register
- Link FTC counter
- Link SRTS Value register

Figure 17 shows the relationship of these registers and counters to the link service clock circuitry.

FIGURE 17 Link service clock generation registers

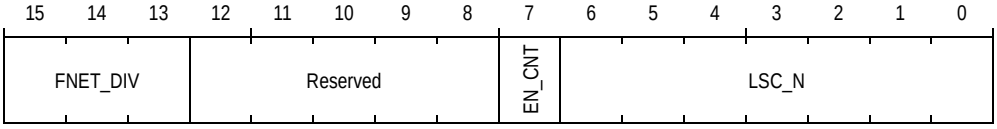


Three of the registers (N, K, L) control a numerically controlled oscillator which has an integer portion N and a fractional portion, K/L. The equation for the output clock is shown below:

$$\text{Link Service Clock period} = \left(N + \frac{K}{L}\right) \times \text{System clock period}$$

Link Service Clock N register

The LSC_N value in the Link Service Clock N register represents the integer multiple of system clock periods used to generate the link service clock. This is the coarse setting of the numerically controlled oscillator used to generate the service clock. Setting bit 7 of this register to one (1) enables the SRTS counters. There is one of these registers for each link.



Bits 15-13 of this register provide a programmable divider for the network clock (FNET) (see Figure 17 on page 33). The table below shows the division provided for various settings of bits 15-13. In addition, the table shows the settings that should be used to support common TDM service clock frequencies when an FNET clock of 19.44 MHz is used.

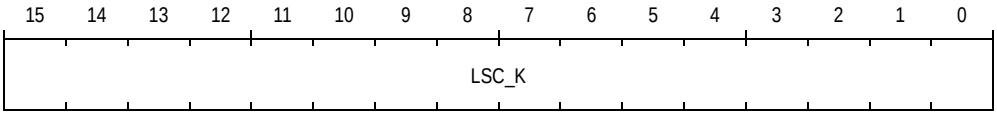
Bits 15-13	*Divide by*	*Divider output frequencies for FNET = 19.44 MHz*	*TDM service clock frequencies for FNET= 19.44 MHz*
0	1	19.44 MHz	
1	2	9.72 MHz	8.192 MHz
2	3	6.48 MHz	
3	4	4.86 MHz	4.096 MHz
4	5	3.888 MHz	
5	6	3.24 MHz	
6	7	2.777 MHz	
7	8	2.43 MHz	2.048 MHz and 1.544 MHz

.

Restriction on FNET_DIV bits
Bits [15:8] of this register are write only and read back as all zeroes. All bits in this register are cleared to zeroes by Reset. Thus, the FNET_DIV bits are write-only and are cleared to zeroes by Reset.

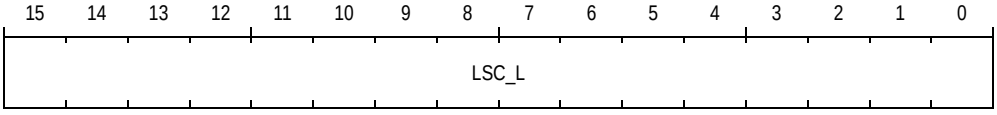
Link Service Clock K register

The value in the Link Service Clock K register represents the fraction control numerator used to generate the link service clock. This is the fine setting of the numerically controlled oscillator used to generate the service clock.There is one of these read/write registers for each link.



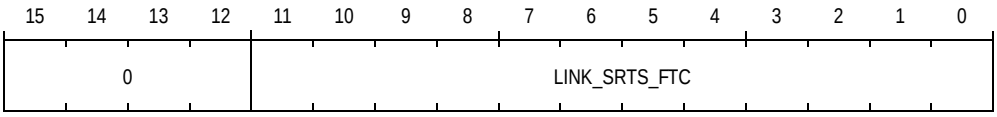
Link Service Clock L counter

The value in the Link Service Clock L counter represents the fraction control denominator used to generate the link service clock. This is a free running counter that can be read and written for diagnostic purposes and can be seeded with an initial value. However, the value loaded into this counter does not alter the effective denominator value, which is always 65,536 (2^{16}). This counter advances once for each service clock cycle generated. There is one of these counters for each link.



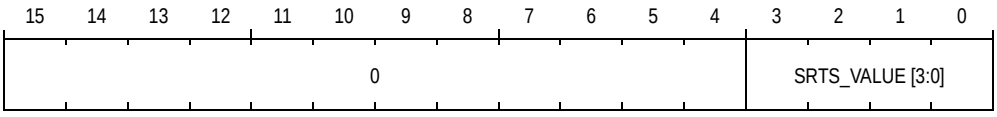
Link FTC counter

The SRTS value generator for each link creates an SRTS value by using an FTC counter to count a designated number (Frequency Terminal Count) of service clock periods. A register common to all links (see “CI SRTS FTC register” on page 45) establishes the number of service clock periods to be counted. This number is generally 3008 (decimal). There is one readable Link FTC counter for each link (see “Link FTC read back” entries in “Circuit Interface per-link registers” on page 85). To change the count used by the Link FTC counter, write the CI SRTS FTC register (page 45).



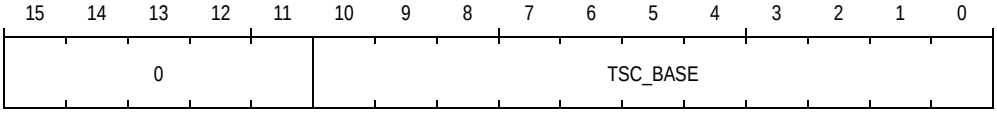
Link SRTS Value register

Each link has a single SRTS generator controlled by the SRTS_MD bit for that link (see “Link Configuration register” on page 12). A 4-bit value is latched into the local SRTS value each time the FTC counter expires (see “Link FTC counter” on page 36). There is one of these read/write registers for each link.



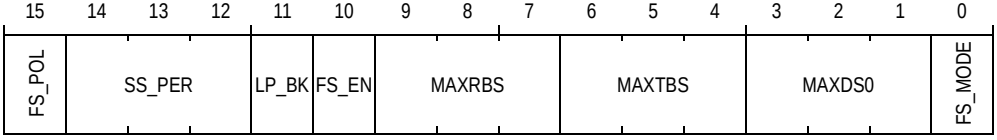
CI TRI-STATE CONTROL BASE ADDRESS REGISTER

The value in this register defines the base address of the Tri-state Control table in Gather Memory. The contents of this register and the Tri-state Control Address counter are combined with the Link ID number (LID) to create an 18-bit pointer to a halfword-aligned entry in Gather Memory containing the tri-state control map for the link. See “Creation of the tri-state control map pointer” on page 28.. There is only one CI Tri-state Control Base Address register; it is a read/write register.



CI CONFIGURATION REGISTER

This register controls the operating mode of the Circuit Interface. There is only one CI Configuration register; it is a read/write register.



FS_POL (Bit 15)

This bit controls the interpretation of the polarity of the Frame Sync input. If this bit is set to one (1), FSYNC active level is zero (0). If this bit is cleared to zero (0), FSYNC active level is one (1).

Snap Shot Period (Bits 14-12)

These bits should be written as 010 and ignored on reads. With this setting, the maximum unidirectional-mode link speed is 8 MHz and the maximum bidirectional-mode link speed is 4, 6, or 8 MHz for System Clock speeds of 33 MHz, 40 MHz, and 50 MHz respectively.

LP_BK (Bit 11)

This bit enables link loopback mode. If this bit is set to one (1), the link pair is placed in loopback mode. In unidirectional mode, all links loop their ASER transmit output back to BSER receive input. In bidirectional mode, the ASER transmit output is looped back to the ASER receive input, and the BSER transmit output is looped back to the BSER receive input. If this bit is cleared to zero (0), the link pair is placed in normal operating mode.

When a link pair is in loopback mode, no data is presented to the ASER or BSER pins. Clock signals are not affected by loopback. During loopback mode, clocking and Frame Sync must be supplied in the same fashion as for normal operating mode.

FS_EN (Bit 10)

This bit enables/disables Frame Sync detection. When asserted, this global bit permits all SDT mode links to acquire Frame Sync simultaneously. If this bit is set to one (1), Frame Sync detection is enabled, enabling all SDT mode links that have their LkRESET bits[1] clear (0). If this bit is cleared to zero (0), Frame Sync detection is disabled, disabling all SDT mode links. This bit only applies to SDT mode links. UDT mode links ignore this bit.

MAXRBS (Bits 9-7)

These bits specify the maximum amount of TDM to ATM receive frame storage allocated for any link in the system.

Bit 6	*Bit 5*	*Bit 4*	*Maximum TDM to ATM receive frame storage*
0	0	0	64 frames
0	0	1	128 frames
0	1	0	256 frames
0	1	1	512 frames
100 through 111			Reserved

1. LkRESET bits are in the Link Configuration registers for each link. See “LkRESET (bit 0)” on page 15.

MAXTBS (Bits 6-4)

These bits specify the maximum amount of ATM to TDM transmit frame storage allocated for any link in the system.

Bit 9	*Bit 8*	*Bit 7*	*Maximum ATM to TDM transmit frame storage*
0	0	0	64 frames
0	0	1	128 frames
0	1	0	256 frames
0	1	1	512 frames
100 through 111			Reserved

MAXDS0 (Bits 3-1)

These bits specify the maximum number of DS0's per frame for any link in the system.

Bit 3	*Bit 2*	*Bit 1*	*Maximum DS0's per frame*
0	0	0	24
0	0	1	32
0	1	0	64
0	1	1	96
1	0	0	128
101 through 111			Reserved

FS_MODE (bit 0)

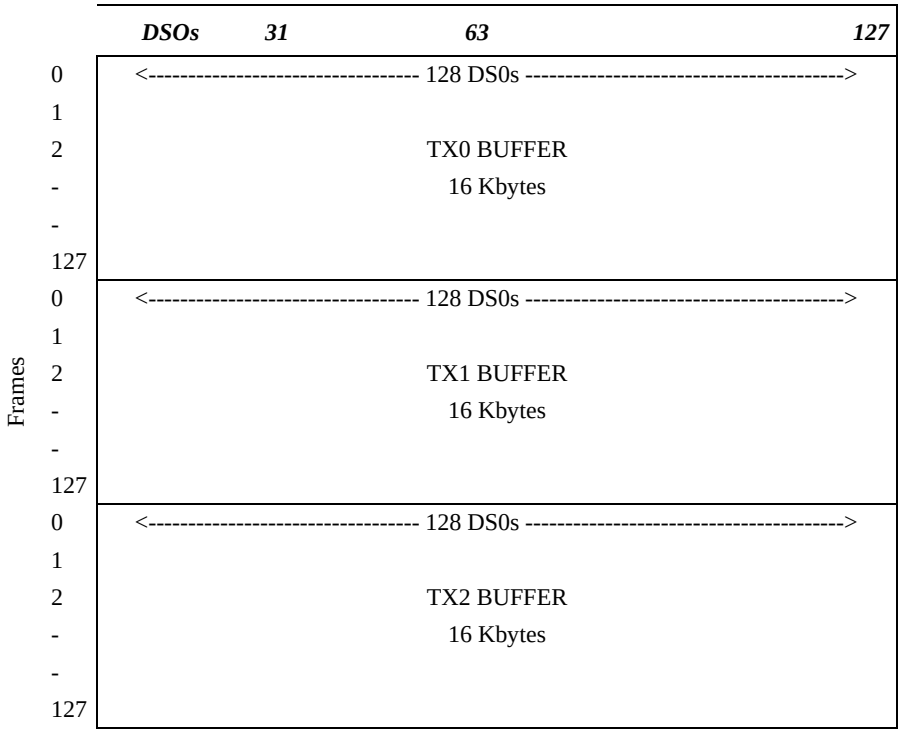
This bit determines the frame sync source for all of the links. If this bit is set to one (1), each link receives its frame sync (or an equivalent signal) from that link's BSCE pin. If this bit is cleared to zero (0), all links receive frame sync (or an equivalent signal) from the chip Frame Sync pin. This bit must be cleared to zero (0) when any links are in the bidirectional mode.

MAXDS0, MAXTBS, and MAXRBS and buffer sizing

The MAXDS0, MAXTBS, and MAXRBS bits define a common buffer size in Scatter/Gather memory for all of the TDM links. The common buffer size for all links is the maximum required by any link being served. The following examples are provided:

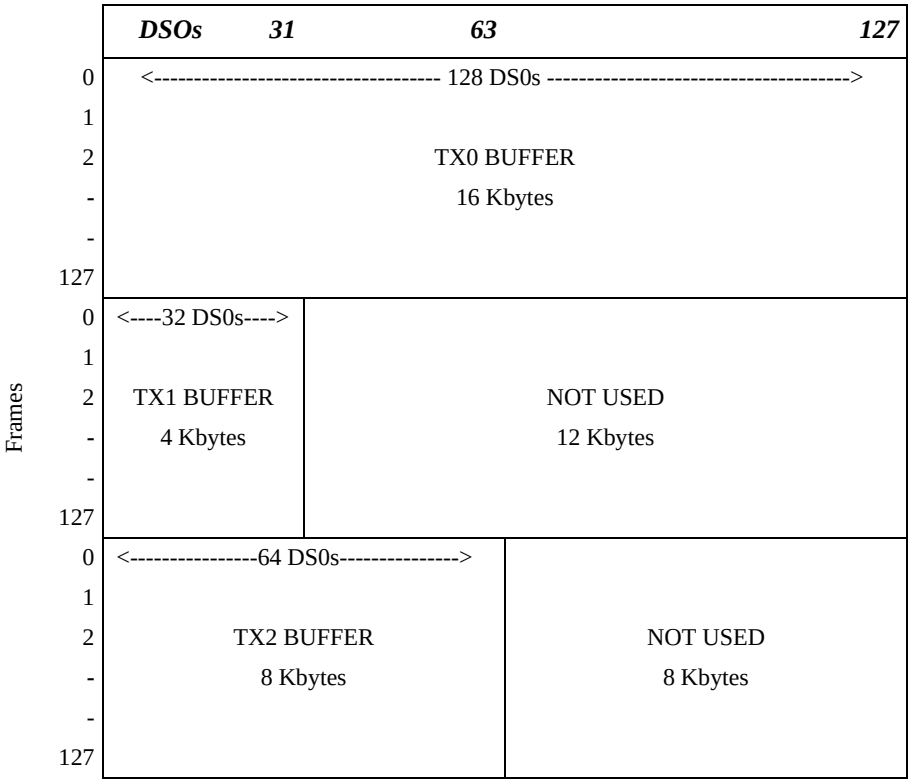
Example #1:

If three active transmit links have a MAXRBS = 128 frames and a MAXDS0 = 128 DS0s, each link buffer is 16 Kbytes (128 x 128).



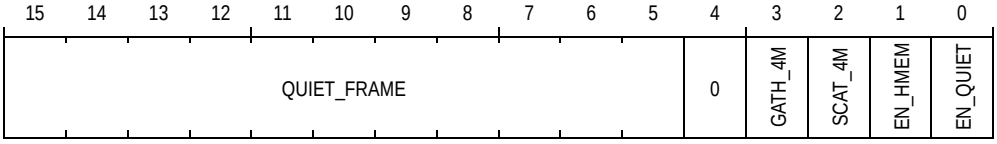
Example #2:

If there are three active transmit links with a MAXRBS = 128 frames and MAXDS0s of 128, 32, and 64 DS0s per frame, all three buffers are the 16 Kbytes (128 x 128) required by the link with the largest buffering requirement. Utilization of the three 16 Kbyte buffers is shown in the figure.



CI QUIET FRAME BASE ADDRESS REGISTER

An 11-bit value in this register defines the base address of the Quiet Frame in Gather Memory. Additional bits indicate the type of Scatter and Gather SRAM used, control access to the high memory space (FE000 - FFC00), and enable the quiet logic. There is only one CI Quiet Frame Base Address register; it is a read/write register.



QUIET_FRAME (Bits 15-5)

These bits define the base address of the Quiet Frame in Gather Memory.

RESERVED (Bit 4)

This bit is reserved. It should be written as a zero and ignored on reads.

GATH_4M (Bit 3)

Set this bit to one (1) if the Gather Memory is implemented with a single 256K x 16 (4 Mb) SRAM. Set this bit to zero (0) if smaller parts are used. This bit controls the Gather Memory address decoding and asserts GATH_OE_[0] when any Gather Memory location is read. GATH_OE_[0] connects to the SRAM output enable pin.

SCAT_4M (Bit 2)

Set this bit to one (1) if the Scatter Memory is implemented with a single 256K x 16 (4 Mb) SRAM. Set this bit to zero (0) if smaller parts are used. This bit controls the Scatter Memory address decoding and asserts SCAT_OE_[0] when any Scatter Memory location is read. SCAT_OE_[0] connects to the SRAM output enable pin.

EN_HMEM (Bit 1)

This bit enables MXT3020 response to Port 2 access in the co-processor high memory space (i.e. FE000 - FFC00). If this bit is set to one (1), the MXT3020 responds to high memory addresses. If this bit is cleared to zero (0), the MXT3020 does not respond to high memory addresses.

When two MXT3020s are present on the same P2 bus, all 2 Mbytes of available address space are assigned. However, the MXT3020 defaults at power-up to not use a small window of space between FE000 and FC000. If a designer wishes, these addresses can be used for a CAM or similar burst mode device. If, however, it is desired to use this space for additional Gather Memory, programs utilizing the MXT3020 should set EN_HMEM to one (1).

EN_QUIET (Bit 0)

This bit enables the quiet logic. If this bit is set to one (1), the quiet logic is enabled. If this bit is cleared to zero (0), the quiet logic is disabled.

Quiet Logic

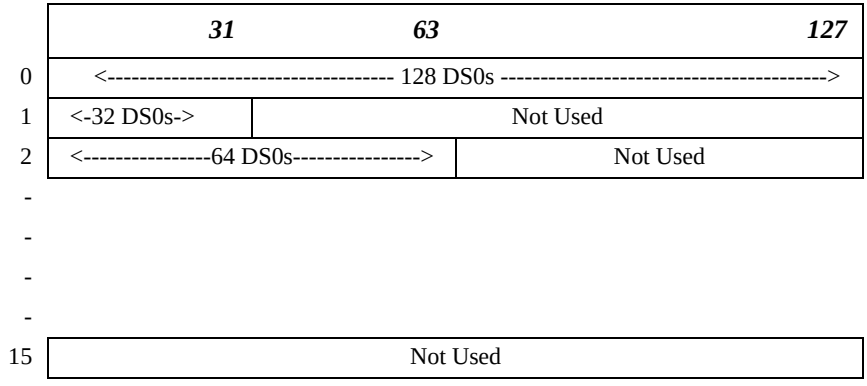
The Circuit Interface contains the *quiet logic*. For each transmit link, a dummy *quiet frame* exists in Gather Memory with the proper quiet value for each DS0 in the frame. The quiet logic copies these quiet frames into the Transmit Link buffers as soon as the data in the buffers has been shifted out onto the TDM lines. The quiet frames reside in the Transmit Link buffers until replaced by data arriving from the ATM link. In this way, the MXT3020 ensures that in the event of an ATM link underflow, the correct quiet values (rather than stale data) are transmitted on the TDM links.

The quiet logic reads the quiet values from predefined quiet frames in Gather Memory and writes them into the TDM transmit frame buffers in Scatter Memory. There must be one quiet frame for every active TDM transmitter. The logic reads bursts of eight halfwords from the Gather Memory and then arbitrates for, and writes, the halfwords into Scatter Memory. This process continues until one frame in each TDM transmit buffer is overwritten.

To be efficient, and to reduce the overhead required for this operation, the controllers use the Link Reset and Data Transfer Mode bits of the Link Configuration registers to skip links that are unused or are setup for unstructured data transfer mode. The quiet frame area can be a maximum size of 16 frames with 128 bytes per frame, or 2 Kbytes in size. The 2-Kbyte area can be located anywhere in Gather Memory above the TDM Link receive buffers, which always begin at Port2 address 0x80000. The Quiet Frame Base Address register locates the quiet frame area and must be setup at initialization. Bit [0] of this register enables/disables the quiet logic.

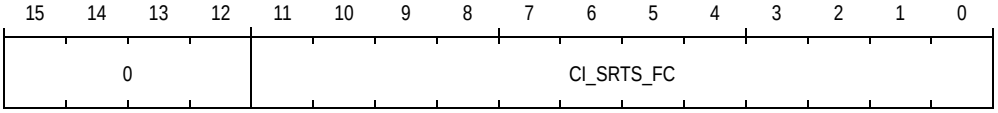
Although each quiet frame has storage for 128 quiet DS0 values, only the actual number of quiet DS0's need to be initialized for each frame. For example, if only three links were transmitting, links 0, 1, and 2, and they supported 128, 32 and 64 DS0's respectively, then the allocation map for the quiet frame area would look like the following:

FIGURE 18 Quiet Frame Allocation Map



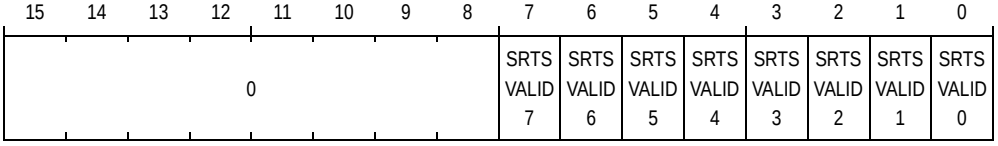
CI SRTS FTC REGISTER

The value in the CI SRTS FTC register represents the number of service clock[1] periods between updates of the Link SRTS Value register (see “Link FTC counter” on page 36 and “Link SRTS Value register” on page 36). Whenever a Link FTC counter counts down to zero, it is reloaded from the CI SRTS FTC register, which is typically set to 3008. There is only one CI SRTS FTC register; it is a read/write register.



CI SRTS VALID STATUS REGISTER

This status register indicates whether a link has a valid SRTS value latched. When a link’s SRTS value is read by the SWAN processor in the MXT3010, its respective valid bit is cleared. There is only one CI SRTS Valid Status register; it is a read-only register.



SRTS_VALID (Bits 7-0)

These bits indicate, on a bit-per-link basis, whether the link corresponding to that bit has a valid SRTS value latched. If bit *n* is a one (1), a valid SRTS value for link *n* has been latched. If bit *n* is a zero (0), a valid SRTS value for link *n* has not been latched.

1. The service clock rate for a T1 facility is 1.544 MHz.

CI STATUS REGISTER

Bits in this register indicate, for each link, whether that link has acquired Frame Sync or has lost Frame Sync. There is only one CI Status register; it is a read-only register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACQ7	LOST7	ACQ6	LOST6	ACQ5	LOST5	ACQ4	LOST4	ACQ3	LOST3	ACQ2	LOST2	ACQ1	LOST1	ACQ0	LOST0

ACQ *n*

These bits indicate whether link *n* has acquired Frame Sync. If an ACQ bit is a one (1), link *n* has acquired Frame Sync. If an ACQ bit is a zero (0), Link *n* has not acquired Frame Sync or has lost Frame Sync.

LOST *n*

These bits indicate whether link *n* has lost Frame Sync. If a LOST bit is a one (1), link *n* has lost Frame Sync after acquiring it. When this bit sets, the corresponding ACQ bit is cleared. This bit is cleared by Link Reset or Chip Reset. If a LOST bit is a zero (0), link *n* has not lost Frame Sync.

INTERFACE PINS

There are five pins associated with each of the eight link pairs. The modes are controlled by the Link Configuration registers.

TABLE 8. Circuit Interface pins, per link

Pin	*Mode*	*Function*	*Drive*
ASER (*y*)	Unidirectional Bidirectional	Transmit Data Transmit/Receive Data (A)	Output I/O
BSER (*y*)	Unidirectional Bidirectional	Receive Data Transmit/Receive Data (B)	Input I/O
ACLK (*y*)	Unidirectional ACLK_MD = 1 ACLK_MD = 0 Bidirectional ACLK_MD = 1 ACLK_MD = 0	 Transmit Link Clock Transmit Input Clock ASER (y) Tri-state Enable Transmit Input Clock	 Output Input Output Input
BCLK (*y*)	All	Receive Clock (for A and B)	Input
BSCE (*y*)	Unidirectional Bidirectional	Link Frame Sync BSER (y) Tri-state Enable	Input Output

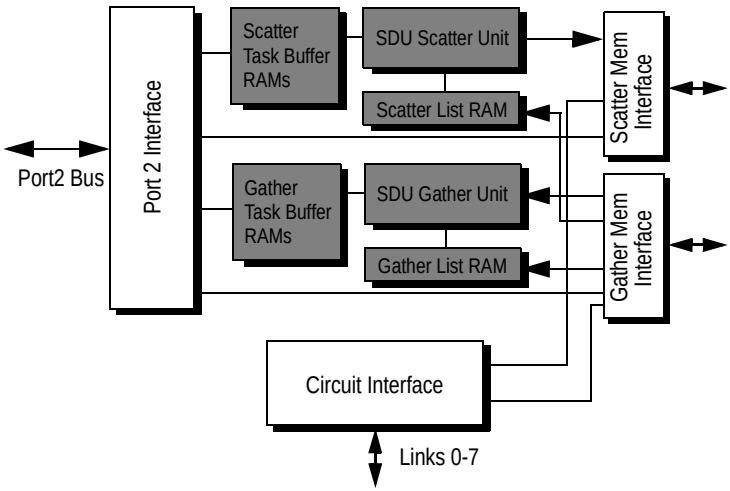
Note: *y* represents the link number, where $0 \le y \le 7$.

In addition, the FSYNC and FNET interface pins are common to all of the links in the Circuit Interface.

TABLE 9. Circuit Interface pins, common to all links

Pin	*Name*	*Function*	*Drive*
FSYNC	Frame Sync	Common Frame Sync input for all links when FS_MODE bit is 0.	Input
FNET	Network Clock	Network clock used for SRTS (reference input)	Input

CHAPTER 3 *Data Mover Units and Task Buffer RAMs*



The Data Mover Units (DMUs) are specialized scatter/gather machines. One DMU disassembles SAR Service Data Units (SDUs) from ATM cells and prepares data for transmission over TDM links. Simultaneously, the other DMU receives data from TDM links and assembles it into ATM SAR SDUs. The Data Mover Units are programmed by the MXT3010, which loads lists of instructions for controlling the data transfer process into a designated area of Gather Memory. The MXT3010 then loads control information and data into the Task Buffer RAM of the scatter machine or control information into the Task Buffer RAM of the gather machine.

When the MXT3010 has finished loading a task buffer in the scatter machine's Task Buffer RAM, the scatter machine does the following:

- Accesses the Gather Memory, fetches the list of instructions ("list block") for controlling the scattering process, and stores those instructions in its List RAM (LRAM),
- Reads sequential bytes of ATM SAR SDU data from the Task Buffer RAM, writing that data into locations in Scatter Memory (TDM transmit frame storage). The locations written are those designated by the instructions in the list block.

When the MXT3010 has finished loading a task buffer in the gather machine's Task Buffer RAM, the gather machine does the following:

- Accesses the Gather Memory, fetches the list of instructions ("list block") for controlling the gathering process, and stores those instructions in its List RAM (LRAM),
- Reads bytes of data from Gather Memory (TDM receive frame storage) and packs the bytes into sequential locations in its Task Buffer RAM. The locations read are those designated by the instructions in the list block.

Additional information about the Task Buffer RAMs is provided in "Task Buffer Format" on page 55, and the Data Mover Interfaces to the Scatter/Gather Memories are discussed in "Scatter and Gather Memory Interfaces" on page 75.

LISTS AND TASKS

A Data Mover Unit program (for scatter or gather operations) consists of two pieces, a *list* and a *task*. The *list* is the code that contains the data moving instructions. It is essentially a list of link numbers and DS0s within a frame that are associated with an ATM SAR SDU. The MXT3010 stores the list in contiguous halfword-aligned locations in Gather Memory, and the Data Mover Unit transfers it to the List RAM. An example is shown below, and additional examples are shown under the heading "Examples of Scatter and Gather Operations" on page 72.

```
LIST:   gath link# ds0#
        gath link# ds0#
   .     (up to 128 instructions per list)
        gath link# ds0#
```

The MXT3010 stores the *task* (a group of control words) in the Task Buffer RAM, and the Data Mover Unit uses it to set up the DMU registers and memory pointers previous to the data move. The formats and functions for these control words are provided in "Task Buffer Format" on page 55.

ACTIVATING THE DATA MOVER UNIT

Activation of a Data Mover Unit requires two steps, loading the list blocks and loading the Task Buffer RAM. Completion of the Task Buffer RAM loading operation automatically initiates Data Mover Unit operation.

Loading list blocks

As the first step toward activating the Data Mover Unit, the MXT3010 loads a *list block* into the portion of Gather Memory that is used for control purposes. A list block is a list of instructions that is responsible for moving data between Task Buffer RAM and Scatter/Gather Memory. After loading the list block, the MXT3010 loads control information into the MXT3020 Task Buffer RAM, including a Channel Map Pointer which points to the first instruction of the list block to be executed.

When the Data Mover Unit begins operation, it fetches the list block, or a piece of the list if it is greater than 64 instructions long, and loads it into its private onboard List RAM (LRAM), from which it subsequently reads and executes instructions. One of the Task Buffer RAM registers contains a counter (List Size) indicating the number of instructions that have been placed in the list block. The Data Mover Unit uses this information to determine when it has reached the end of a set of list block instructions.

Restrictions on list blocks

The list block must start on a Gather memory boundary based on the size of the list.

List Size (See "List Size" on page 57.)	*Address Boundary*
0 - 1	8 bytes
2 - 3	16 bytes
4 - 7	32 bytes
8 - 15	64 bytes
16 - 31	128 bytes
32 - 63	256 bytes
64 - 127	512 bytes

Each list block must be repeated twice (concatenated on the end.) For example, if a list block had four instructions, A, B, C, and D, it would appear in memory as: A, B, C, D, A, B, C, D. In this example, the list block and its copy would occupy a total of eight halfwords (sixteen bytes).

Generating addresses

The generation of addresses for the MXT3010 to load the list blocks and for the Data Mover Unit to access the list blocks is described in greater detail in "Address Generation" on page 67. The generation of List RAM addresses is also covered in that section.

Transfer Count and Instruction Pointer

A list block identifies only the DS0s within a frame that need to be moved. A frame can contain 24, 32, 64, 96, or 128 DS0s, but a list block can call out a small number for transfer. Once all the instructions in the list block have been executed, and all of the selected DS0s in a data frame have thus been transferred, the Data Mover Unit moves to the next frame in memory and executes the list block instructions again. Once all the data has been transferred, the Data Mover Unit will stop executing the list at a particular instruction in the List RAM. When this list block is used again, the Channel Map Pointer in the Task Buffer Register must point to this same instruction

so that the scatter or gather process can continue from where it left off. To calculate the proper new value for the Channel Map Pointer, software must keep track of the size of the list and the number of bytes transmitted.

The list blocks only have to be set up when a virtual circuit is first established and can remain untouched while that virtual circuit exists. As more data arrives, the MXT3010 re-loads the task buffer changing only the Channel Map Pointer and the Frame Number to new values which it has calculated.

Loading the Task Buffer RAM

The Data Mover Unit is activated by the MXT3010 performing a burst mode DMA transfer into the Task Buffer RAM of the selected Data Mover Unit. In the case of a scatter operation, this DMA transfer includes both control information and SAR data. In the case of a gather operation, the DMA transfer includes only control information.

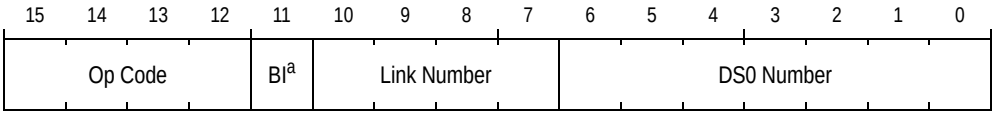
Completion of the DMA transfer automatically sets the Task Buffer Busy flag for that task buffer. The control words are automatically read from the buffer and loaded into the Data Mover Unit's state registers. In addition, an instruction list is fetched from Gather Memory and loaded into the List RAM. Starting at location zero in the List RAM, each element of the list is fetched and memory-to-memory transfers begin.

In the case of a scatter operation, these transfers occur between the Task Buffer RAM and the Scatter Memory. This process continues until the command queues are exhausted, at which point access control to the Scatter Memory is released. The completion status is indicated to the MXT3010 via the Scatter Task Buffer Busy flags (STBR_BSY[1:0]).

For a gather operation, the procedure is identical except that the memory-to-memory transfers occur between the Gather Memory and the Task Buffer RAM, as SAR data is placed into the task buffer by the Data Mover Unit. When all the SAR data has been gathered, the completion status is indicated to the MXT3010 via the Task Buffer Busy flags (see "Status (STAT)" on page 65). The MXT3010 must then perform DMA transfers to move the SAR data back over the Port2 bus.

DATA MOVER UNIT INSTRUCTION SET

The list blocks contain instructions for the Data Mover Unit. The Data Mover Unit instruction set consists of seven instructions.All seven instructions are of the following format:



a. The BI bit is one (1) if the link is bidirectional and zero (0) if the link is unidirectional.

:

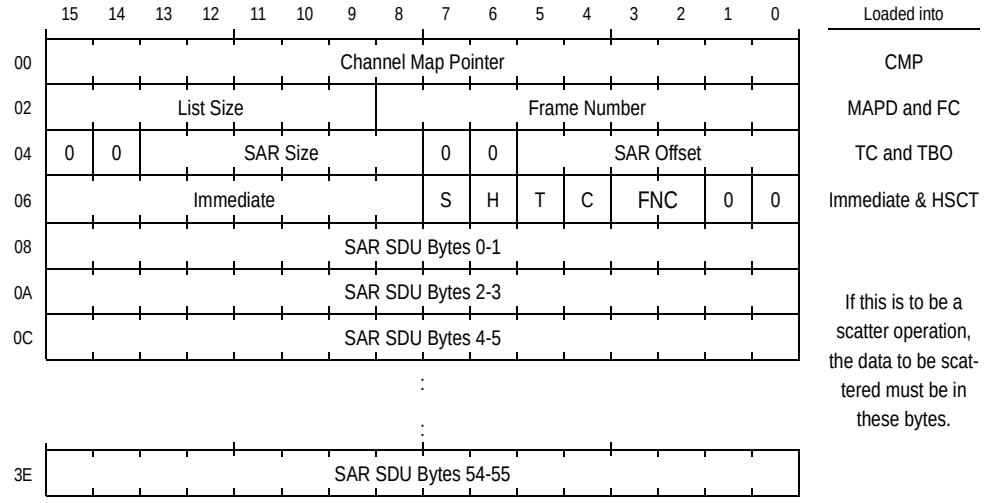
TABLE 10. Data Mover Unit Instructions

Op Code	*Instruction*	*Action*
0000	No-Op	None
0001	Reserved	Reserved
0010	Gather	Reads one byte of data from the Gather Memory address (Link#, Frame Count, DS0#) and stores it in the Task Buffer RAM at the Task Buffer Offset (TBO) location. TBO is incremented; Transfer Count (TC) is decremented.
0011	Gather Immediate	Stores the value of a constant (CONST) in the Task Buffer RAM at the Task Buffer Offset (TBO) location. TBO is incremented and Transfer Count (TC) is decremented.
0100	Scatter	Reads one byte of data from the Task Buffer RAM at the Task Buffer Offset (TBO) location and stores it into Scatter Memory Address (Link#, Frame Count, DS0#). TBO is incremented; Transfer Count (TC) is decremented.
0101	Scatter Immediate	Stores the value of a constant (CONST) into Scatter Memory Address (Link#, Frame Count, DS0#). TBO is incremented; TC is decremented.
0110	Mcast	Reads one byte of data from the Task Buffer RAM at the Task Buffer Offset (TBO) location and stores it into Scatter Memory Address (Link#, Frame Count, DS0#). TBO and TC are NOT changed.
0111	Mcast Immediate	Stores the value of a constant (CONST) into Scatter Memory Address (Link#, FC, DS0#). TBO and TC values are NOT changed.

TASK BUFFER FORMAT

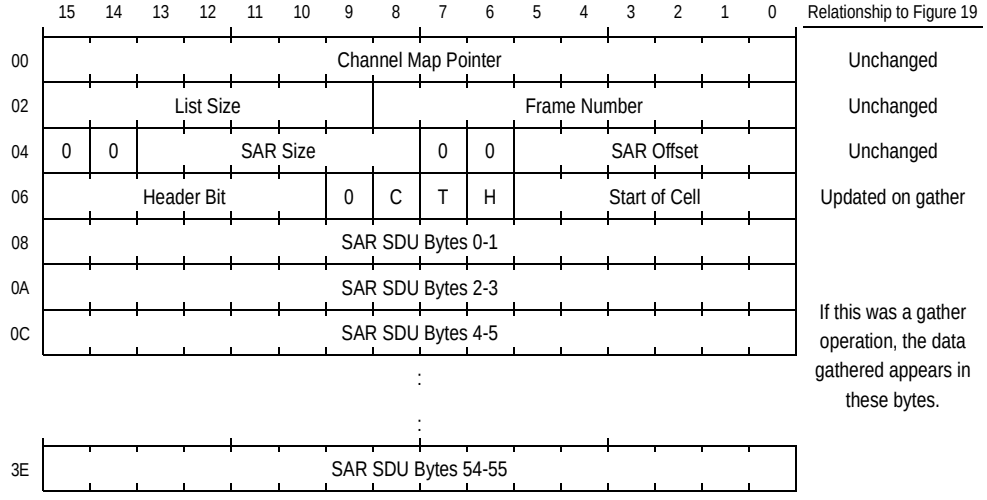
Each Task Buffer RAM has space for two 64-byte task buffers. The addresses shown are the offsets into a single task buffer. At the conclusion of the DMA transfer that loads the task buffer, control information is automatically transferred to the Data Mover Unit registers indicated. Therefore, the information placed into the buffer must correspond to the format shown.

FIGURE 19 Pre-Scatter/Gather Task Buffer Register Format



When the scatter or gather operation has completed, the Task Buffer Register has the following format:

FIGURE 20 Post-Scatter/Gather Task Buffer Register Format



Channel Map Pointer

When combined with the contents of the ISEG[1:0] register, this 16-bit field points to a list block located in Gather Memory. The bits represent Gather Memory address bits [16:1]. As indicated in “Restrictions on list blocks” on page 52, there are two copies of each list block stored in memory. The Channel Map Pointer must point to an instruction in the first copy of the list block. This pointer is loaded into the Channel Map Pointer register in the DMU.

List Size

This is a 7-bit counter indicating the number of instructions that have been placed in the list block. It should be set to *n*-1, where *n* is the number of instructions in a single copy of the list block. This information is loaded into the MAPD and Fill registers in the DMU, where it is used to determine when the DMU has reached the end of the list block instructions.

Frame Number

This is a 9-bit counter identifying the data frame within a Link Buffer in Scatter/ Gather Memory. This information is loaded into the Frame Counter (FC) register in the DMU.

SAR Size

This is a 6-bit counter initialized with the number of data transfers to be performed, as determined by the amount of SAR data. It should be set to *n*-1, where *n* is the number of data transfers to be performed (including HEC if enabled). This information is loaded into the Transfer Counter (TC) register in the DMU, where it is decremented after each data transfer is completed.

SAR Offset

The 6-bit value in the SAR Offset field should equal the offset of the first byte of SAR SDU data. For example, if the first task buffer was loaded, and the first SAR SDU data byte was placed at location 0A, then the contents of the SAR Offset field should equal 0A. The SAR SDU data can start anywhere beyond the first eight bytes of the buffer but must be contiguous. This information is loaded into the Task Buffer Offset (TBO) register in the DMU. In the DMU, this value is used as a counter that points to the next location to be read or written in the Task Buffer. Since the Task Buffers are 64 bytes, only bits [5:0] are required for addressing, and bits [7:6] are always zero.

Register 06 (write) - Additional control bits

The bits in this register function as control bits when the Task Buffer Register is written.

Bits 18-8 (write) Immediate

On writes to the Task Buffer RAM, this register contains data for use by gather immediate, scatter immediate and mcast immediate instructions. This information is loaded into the CONST register in the DMU. This register must be loaded, even if it is loaded with all zeroes, because loading this register initiates Data Mover Unit operation.

Bit 7 (write) S-bit

On writes to the Task Buffer RAM, this bit controls the scrambling of data on scatter operations and the descrambling of data on gather operations. When this bit is zero (0), scrambling/descrambling is disabled. When this bit is one (1), scrambling/descrambling is enabled.

Bit 6 (write) H-bit

On writes to the Task Buffer RAM, this bit controls the generation of the Header Error Control (HEC) on scatter operations[1] and the checking of the HEC on gather operations[2]. When HEC generation/checking is enabled on gather operations, cell delineation is also enabled, as the MXT3020, in the process of searching for HEC, will report its position in the Start of Cell register (see “Bits 5-0 (read) SOC” on page 61) When this bit is zero (0), HEC generation/checking is disabled. When this bit is one (1), HEC generation/checking is enabled.

1. The generated HEC byte is inserted as the fifth byte of the cell.
2. The received HEC byte is discarded after checking.

Bit 5 (write) T-bit

On writes to the Task Buffer RAM, this bit enables/disables the Threshold Test on gather operations. See “TT register” on page 66. When this bit is zero (0), the Threshold Test is disabled. When this bit is one (1), the Threshold Test is enabled.

Bit 4 (write) C-bit

On writes to the Task Buffer RAM, this bit enables/disables CRC-10 operation. When this bit is zero (0), CRC-10 operation is disabled. When this bit is one (1), CRC-10 operation is enabled.

Bits 3-2 (write) FNC bits

On writes to the Task Buffer RAM, these bits control details of the CRC-10 function on gather and scatter operations.

Bits 3-2	***Description***
0 0	Check CRC and save partial result[a]
0 1	Check CRC and discard
1 0	Generate CRC and save partial result[a]
1 1	Generate CRC and append to transmission

a. The 00 and 01 codes are used when the scatter/gather being performed does not include the end of a message. Codes 01 and 11 are used at the end of a message,

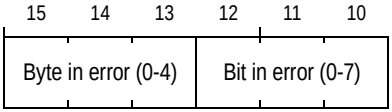
When the burst DMA which loaded this task buffer concludes, this information is automatically transferred to the HSCT Flags register in the DMU register set.

Register 06 (read) - Additional status bits

The bits in this register function as status bits when the Task Buffer Register is read.

Bits 15-10 (read) Header Bit in Error

If HEC checking is enabled and a header bit error occurs, the H-bit (bit [06]) is set. These bits identify which bit in the 5-byte header was received in error. The following format is used:



If no header bit error is detected, this register will return the value 3F.

Bit 9 (read)

This bit reads as zero (0).

Bit 8 (read) C-bit

On reads of the Task Buffer RAM, this bit indicates the result of the CRC-10 checking. If this bit is zero (0), no CRC-10 error was detected. If this bit is one (1), a CRC-10 error was detected.

Bit 7 (read) T-bit

On reads of the Task Buffer RAM, this bit indicates the result of the Threshold Test. If this bit is zero (0), Threshold Test checking was successful. If this bit is one (1), Threshold Test checking failed.

Bit 6 (read) H-bit

On reads of the Task Buffer RAM, this bit indicates the result of the Header Error Control (HEC) checking. If this bit is zero (0), HEC checking was successful. If this bit is one (1), HEC checking failed. The bit in error is reported in bits 15-10.

Bits 5-0 (read) SOC

On reads of the Task Buffer RAM, these bits (SOC) indicate the number of the gathered byte in which the HEC code (indicating the Start Of Cell (SOC)) was found.

DATA MOVER UNIT REGISTERS

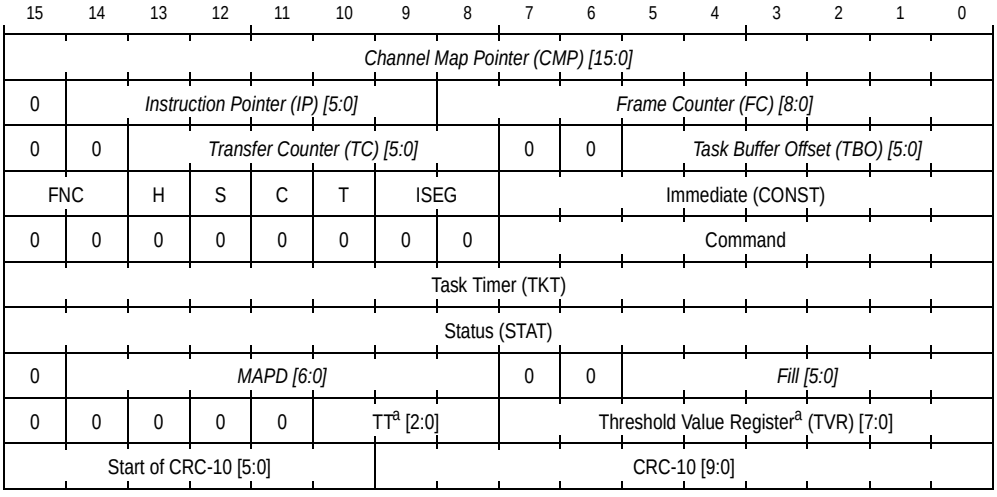
The Data Mover Unit register set contains 17 registers. Many of these registers (marked “A” in Table 11) are loaded automatically by the Data Mover Unit once the list block instructions have been loaded into the Gather Memory, and commands have been loaded into the Task Buffer RAMs.

TABLE 11. Data Mover Unit Registers

Register	*A*	*Function*
Channel Map Pointer (CMP)	A	See “Channel Map Pointer” on page 56.
Instruction Pointer (IP)		See “Instruction Pointer (IP)” on page 63.
Frame Counter (FC)	A	See “Frame Number” on page 57.
Transfer Counter (TC)	A	See “SAR Size” on page 57.
Task Buffer Offset (TBO)	A	See “SAR Offset” on page 57.
CRC Function Code (FNC)	A	See “CRC Function Code (FNC)” on page 63.
HSCT Flags (HSCT)	A	See “Register 06 (write) - Additional control bits” on page 58.
Instruction Segment (ISEG)		See “Instruction Segment (ISEG)” on page 64.
Immediate (IMMED)	A	See “Register 06 (write) - Additional control bits” on page 58.
Command (CMD)		See “Command (CMD)” on page 64.
Task Timer (TKT)		See “Task Timer (TKT)” on page 65.
Status (STAT)		See “Status (STAT)” on page 65.
Map Data counter (MAPD)	A	See “List Size” on page 57.
Fill Address counter (FILL)	A	See “List Size” on page 57.
Threshold Test Type register (TT)		See “TT register” on page 66.
Threshold Value register (TVR)		See “TVR register” on page 66.
Start of CRC-10 register		See “Start of CRC-10 [5:0]” on page 66.
CRC-10 register		See “CRC-10 [9:0]” on page 66.

The Data Mover Unit register set is organized as shown in Figure 21. With the exception of the Instruction Segment (ISEG) Task Timer (TKT), Scatter Count (SC), Gather Count (GC), Threshold Test Type (TT), and Threshold Value (TVR) registers, none of these registers are accessed by the MXT3010 except in debug or diagnostic situations.

FIGURE 21 DMU register set organization



a. The TT and TVR registers exist only in the Gather DMU.

The registers shown in italics are initialized from task buffer entries and are described in “Task Buffer Format” on page 55. Once the initial values have been automatically loaded from the task buffer, they are incremented/decremented by the Data Mover Unit as appropriate during the data transfer process.

Channel Map Pointer (CMP)

This register is loaded from the Task Buffer RAM. See “Channel Map Pointer” on page 56.

Instruction Pointer (IP)

This is a 7-bit counter that points to the next location to be read in the List RAM. Bits [15:9] represent Instruction Pointer bits [6:0]. Since the List RAMs are 64x16, only bits [5:0] are required for addressing, and bit [6] is always zero.

Frame Counter (FC)

This register is loaded from the Task Buffer RAM. See “Frame Number” on page 57.

Transfer Counter (TC)

This register is loaded from the Task Buffer RAM. See “SAR Size” on page 57.

Task Buffer Offset (TBO)

This register is loaded from the Task Buffer RAM. See “SAR Offset” on page 57.

CRC Function Code (FNC)

This register is loaded from the Task Buffer RAM. See “Register 06 (write) - Additional control bits” on page 58.

HSCT Control Flags

This register is loaded from the Task Buffer RAM. See “Register 06 (write) - Additional control bits” on page 58.

Instruction Segment (ISEG)

This is a static register which holds bits [18:17] of the List Block address in Gather Memory. These bits are appended to the Channel Map Pointer and must be initialized by the MXT3010. ISEG[1:0] correspond to Gather Memory address bits [18:17].

TABLE 12. Relation of ISEG register to address ranges

ISEG	*Offset in Gather Memory*	*MXT3020 Address*[a]
00	00000-1FFFE	0x80000-0x9FFFE
01	20000-3FFFE	0xA0000-0xBFFFE
10	40000-5FFFE	0xC0000-0xDFFFE
11	60000-7FFFE	0xE0000-0xFFFFE (Do not use beyond 0xFE000)[a]

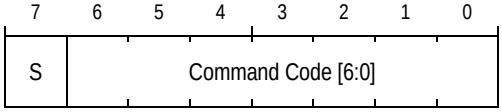
a. See “MXT3020 address space” on page 83.

Immediate (CONST)

This register is loaded from the Task Buffer RAM. See “Register 06 (write) - Additional control bits” on page 58.

Command (CMD)

This register controls the Task Timer and the DMU Halt Flag. Unlike other registers in which the programmer must consider the proper states of all bits before writing to the register, each seven bit command code in this register controls a single specific function independently of other functions. For example, entering 1000 0001 to enable the Task Timer has no effect on the Halt/Continue function.



S-bit	*Command Code*	*Description*
0	000 0000	Halt (For diagnostic use)
1	000 0000	Continue (For diagnostic use and to recover from an illegal instruction trap)
0	000 0001	Disable Task Timer
1	000 0001	Enable Task Timer
All other codes are reserved.		

Task Timer (TKT)

This is a 16-bit counter, intended for performance monitoring, that counts clock cycles while either Task Buffer Busy flag is set.

Status (STAT)

This is an 16-bit register containing the status of the Data Mover Unit.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	HALT	TKT_EN	0	IIT	0	IDU_BSY	DMI_BSY	TBI_BSY	0	0	TBR_BSY	

Bit	*Name*	*Function*
15-12	0	Unused
11	HALT	DMU Halt Flag (To set/clear, see “Command (CMD)” on page 64)
10	TKT_EN	Task Timer Enable (To set/clear, see “Command (CMD)” on page 64)
9	0	Unused
8	IIT	Illegal Instruction Trap (To recover, see “Command (CMD)” on page 64)
7	0	Unused
6	IDU_BSY	IDU Busy Flag (For debug use only)
5	DMI_BSY	DMI Busy Flag (For debug use only)
4	TBI_BSY	TBI Busy Flag (For debug use only)
3	0	Unused
2	0	Unused
1-0	TBR_BSY	Task Buffer Busy Flags[a]

a. Use these bits to detect scatter/gather completion. In addition to their appearance in the Status registers, these flags also appear on external pins. See “Interface Pins” on page 71.

MAPD and Fill

These registers are loaded from the Task Buffer RAM. See “List Size” on page 57.

TT register

This register controls the type of thresholding done on the gathered data. The three bits permit a choice of eight comparisons between gathered data and the value in the Threshold Value Register (TVR).

TT register	*Threshold comparison*	*Comparison type*
000	gath_data [7:0] ≤ TVR [7:0]	unsigned integer
001	gath_data [7:0] > TVR [7:0]	unsigned integer
010	gath_data [7:0] = TVR [7:0]	unsigned integer
011	gath_data [7:0] ≠ TVR [7:0]	unsigned integer
100	gath_data [6:0] ≤ TVR [6:0]	sign-magnitude integer
101	gath_data [6:0] > TVR [6:0]	sign-magnitude integer
110	gath_data [6:0] = TVR [6:0]	sign-magnitude integer
111	gath_data [6:0] ≠ TVR [6:0]	sign-magnitude integer

TVR register

This register contains an 8-bit value used for threshold testing done on the gathered data.

CRC-10 [9:0]

The value loaded into this register before a scatter/gather operation is used as the initial or seed value for CRC-10 calculations. The MXT3020 clears this register at the end of a scatter/gather if CRC-10 was enabled and a function code (FNC) of 01 or 11 was selected. At all other times, this register is updated with the CRC-10 partial result.

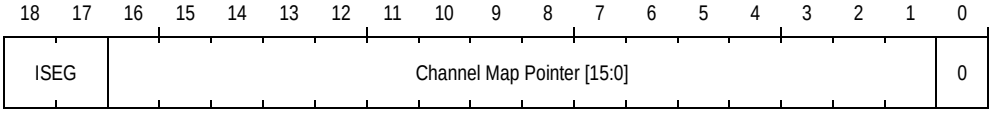
Start of CRC-10 [5:0]

This register indicates the number of bytes that are to be processed before CRC-10 calculation begins. Any value from 0 to 63 may be selected.

ADDRESS GENERATION

List Block Address Generation

The list block, or a portion thereof, is loaded into the List RAM at the start of a Data Mover Unit operation. The Channel Map Pointer contains a 16-bit value that is used to form an 19-bit index into Gather Memory where the list is located. The 19-bit address, [18:0], is formed by using the Channel Map Pointer contents as address bits [16:1]. Bit [0] is always set to zero since list entries are always 16-bits wide. Bits [18:17] are set up by the MXT3010 at initialization time by writing the ISEG Register (page 64).:



The Channel Map Pointer in the Data Mover Unit increments automatically as instructions are fetched from Gather memory, and the process continues until 64 instructions have been fetched, or until the List Size has been downcounted to indicate that the instruction being fetched is the last instruction.

List RAM Address Generation

The List RAM address is a 6-bit address that is generated by the Data Mover Unit's Instruction Pointer (IP) (page 63). The list block is copied into the List RAM, starting at the address pointed to by the Channel Map Pointer. As the list block is being copied into the List RAM, the Instruction Pointer begins cycling through the valid List RAM entries, starting at zero (the first entry in the List RAM), and the instructions at those locations are executed. The Instruction Pointer wraps to zero when the List Size register indicates that the last instruction of the List RAM has been reached, and repeats the process of cycling through the instructions in the List RAM until the Transfer Counter (TC) register reaches zero, and the last byte of data has been processed.

For diagnostic and debug purposes, the List RAM is mapped into the address space of the MXT3020 and can be read and written directly through the Port2 interface.

Scatter/Gather Memory Address Generation

Although Data Mover Unit instruction lists are located in Gather Memory, Scatter/Gather Memory address generation refers to the creation of addresses required to access a specific DS0 within a data frame. These DS0 addresses are generated by merging the contents of the link number field of the data move instruction, the Frame Counter (FC) and the DS0 number field of the data move instruction into a 19-bit memory address. How these elements are merged depends on the content of the MAXDS0[2:0] and MAXTBS/MAXRBS[2:0] fields in the CI Configuration register. The MAXDS0 field identifies how many DS0s compose a serial data frame. The MAXTBS/MAXRBS field tells how many DS0-sized frames make up the serial data buffer storage area for the link number. The figures below show how the complete address is built for various settings of the MAXDS0[2:0] and MAXTBS/MAXRBS[2:0] fields in the Circuit Interface Configuration register.

Figure 22, Figure 23, and Figure 24 apply to unidirectional mode. Figure 25, Figure 26, and Figure 27 apply to bidirectional mode.

FIGURE 22 Creation of 19-Bit Address (Uni) When DS0 = 24 or 32

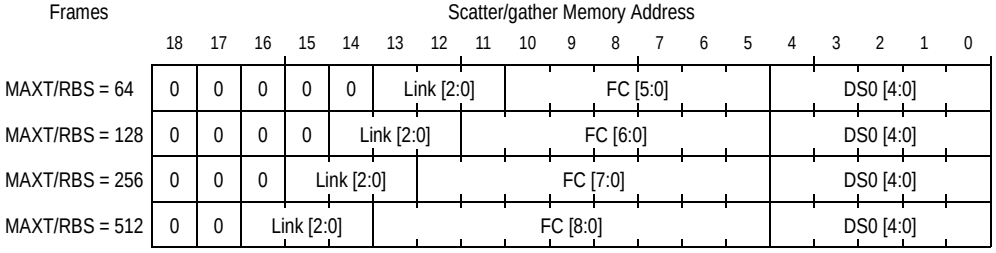


FIGURE 23 Creation of 19-Bit Address (Uni) When DS0 = 64

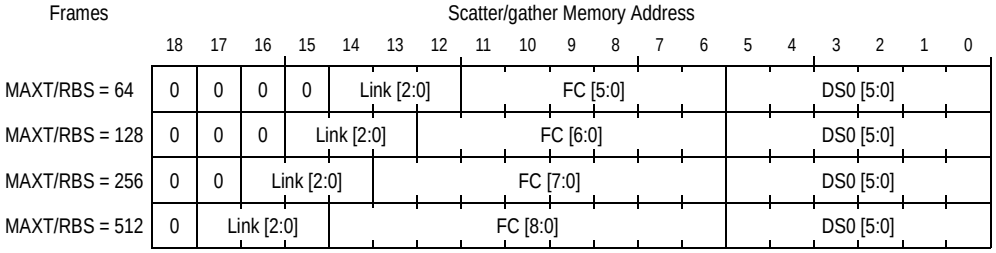


FIGURE 24 Creation of 19-Bit Address (Uni) When DS0 = 96 or 128

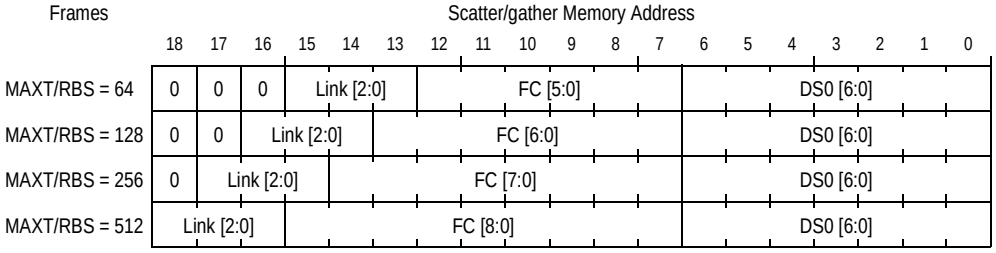


FIGURE 25 Creation of 19-Bit Address (Bi) When DS0 = 24 or 32

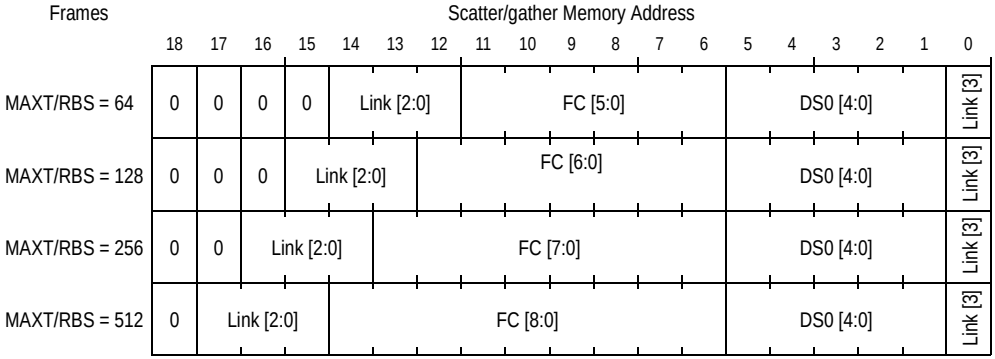


FIGURE 26 Creation of 19-Bit Address (Bi) When DS0 = 64

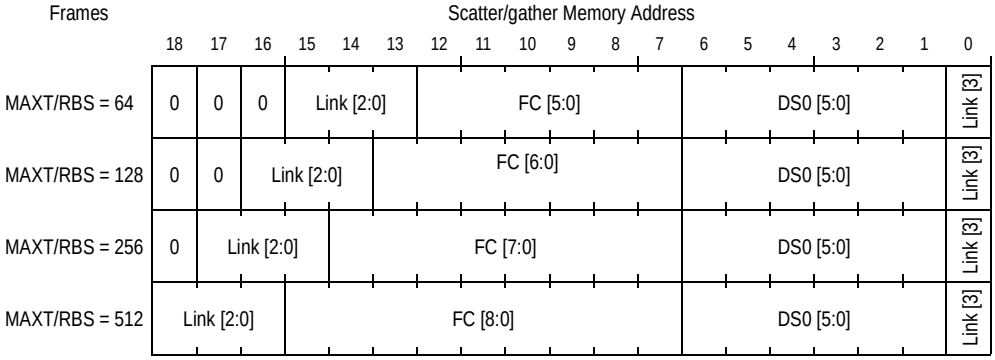
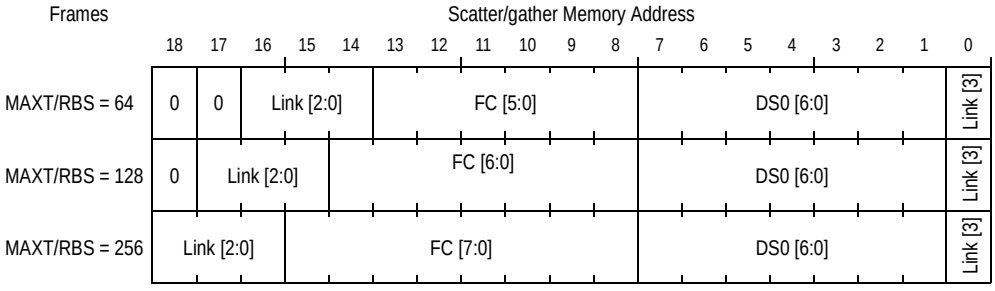


FIGURE 27 Creation of 19-Bit Address (Bi) When DS0 = 96 or 128



Task Buffer RAM Address Generation

The address for accesses to the Task Buffer RAM are provided directly by the Task Buffer Offset register. The Task Buffer Offset points to the byte location in Task Buffer RAM that is to be moved to Scatter Memory or which is to be written with data from Gather Memory. After the execution of a scatter or gather instruction, the Task Buffer Offset is automatically incremented to point at the next sequential location in the Task Buffer RAM.

INTERFACE PINS

The MXT3020C provides a means for the MXT3010 to read the status of the Scatter Task Buffer Busy flags and Gather Task Buffer Busy flags for up to four MXT3020C devices with a single non-burst DMA read of location 0x400000. This read also clears the flags.

TABLE 13. Data Mover Unit and Task Buffer RAM interface pins

Pin number	*Pin name*	*Function*	*Drive*
188	STBRBSY1	Scatter Task Buffer 1 Busy flag	Output
187	STBRBSY0	Scatter Task Buffer 0 Busy flag	Output
186	GTBRBSY1	Gather Task Buffer 1 Busy flag	Output
185	GTBRBSY0	Gather Task Buffer 0 Busy flag	Output
184	M20INT	MXT3020 Interrupt	Output

Pins 188 through 185 are tri-stated until a single non-burst DMA read of location 0x400000 occurs. Wiring details for using these signals in both single MXT3020 and multiple MXT3020 configurations are provided in “Scatter/Gather Task Buffer Busy flags” on page 134.

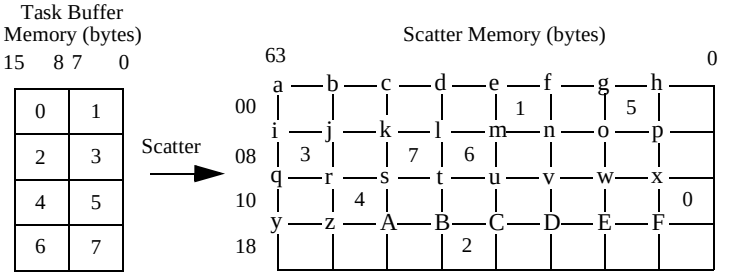
EXAMPLES OF SCATTER AND GATHER OPERATIONS

Scatter

The scatter process reads bytes of data from sequential locations in the Task Buffer RAM and writes them into random locations in Scatter Memory. The Data Mover Unit constructs the Scatter Memory address of the byte to scatter as described in “Scatter/Gather Memory Address Generation” on page 68. The Data Mover Unit also maintains a pointer to the next location in the Task Buffer RAM as described in “Task Buffer RAM Address Generation” on page 71. Both addresses are automatically incremented after the scatter instruction is issued. The figure below demonstrates the results of executing a list of eight scatter instructions. Eight sequential bytes from the Task Buffer RAM are scattered into eight locations in Scatter Memory.

```
scat link(0) ds0(x) //Task Buffer address
scat link(0) ds0(e) //is incremented after
scat link(0) ds0(B) //each instruction
scat link(0) ds0(i)
scat link(0) ds0(r)
scat link(0) ds0(g)
scat link(0) ds0(l)
scat link(0) ds0(k)
```

FIGURE 28 Example of a Scatter Operation

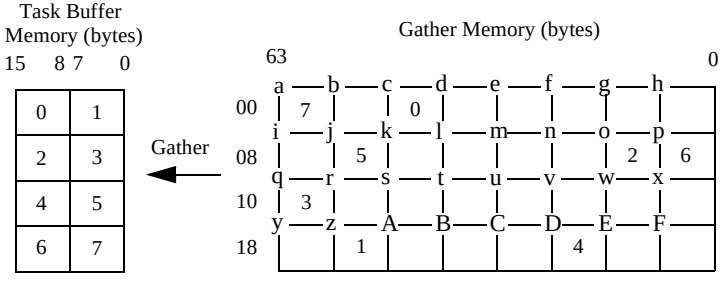


Gather

The gather process reads bytes of data from random locations in Gather Memory and writes them sequentially into the Task Buffer RAM. The Data Mover Unit constructs the Gather Memory address of the byte to gather as described in “Scatter/Gather Memory Address Generation” on page 68. The Data Mover Unit also maintains a pointer to the next location in the Task Buffer RAM as described in “Task Buffer RAM Address Generation” on page 71. Both addresses are automatically incremented after the gather instruction is issued. The figure below demonstrates the results of executing a list of eight gather instructions. Eight bytes from a 32-byte frame are collected and packed sequentially into eight locations in the Task Buffer RAM.

```
gath link(0) ds0(c) //Task Buffer address
gath link(0) ds0(z) //is incremented after
gath link(0) ds0(o) //each instruction
gath link(0) ds0(q)
gath link(0) ds0(D)
gath link(0) ds0(j)
gath link(0) ds0(p)
gath link(0) ds0(a)
```

FIGURE 29 Example of a Gather Operation

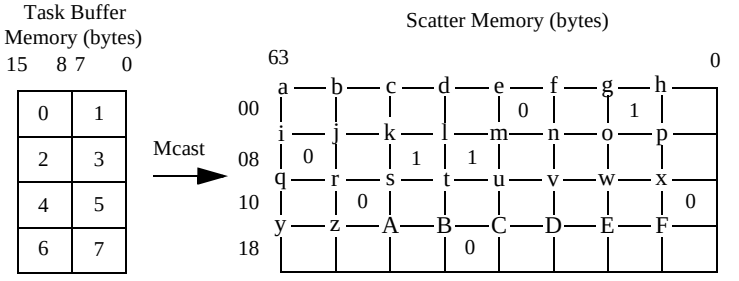


Multicast

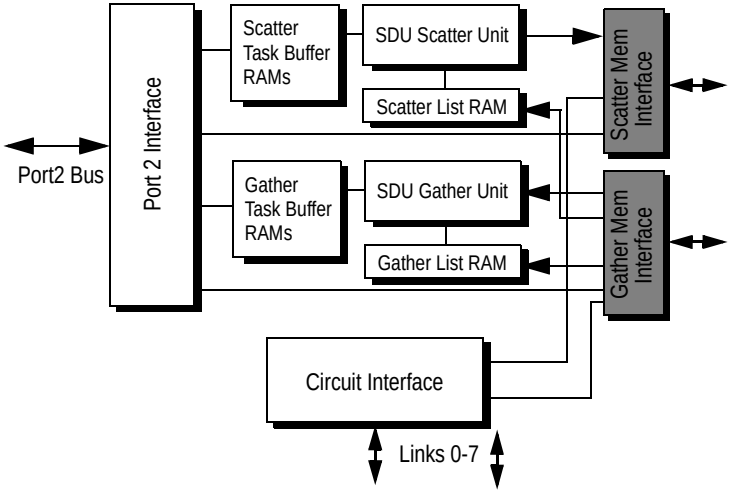
The multicast process is similar to the scatter process in that bytes are moved from Task Buffer RAM to Scatter Memory. The difference is that the scatter process is a one-to-one process, meaning that one byte of Task Buffer RAM data is mapped to exactly one location in Scatter Memory. In contrast, the multicast process is a one-to-many process. A byte of Task Buffer RAM data can be copied to multiple locations in Scatter Memory. As with the scattering process, the Data Mover Unit constructs the Scatter Memory address of the byte to scatter as described in “Scatter/Gather Memory Address Generation” on page 68. The Data Mover Unit also maintains a pointer to the Task Buffer RAM as described in “Task Buffer RAM Address Generation” on page 71. Unlike the scatter operation, however, the Task Buffer RAM pointer is NOT incremented after the issue of an mcast instruction. The following figure demonstrates the results of executing the list of mcast instructions.

```
mcast    link(0)    ds0(x)    //Task Buffer address
mcast    link(0)    ds0(e)    //is NOT incremented
mcast    link(0)    ds0(B)
mcast    link(0)    ds0(i)
scat     link(0)    ds0(r)    //increments TBO
mcast    link(0)    ds0(g)
mcast    link(0)    ds0(l)
mcast    link(0)    ds0(k)
```

FIGURE 30 Example of an Mcast Operation



CHAPTER 4 *Scatter and Gather Memory Interfaces*



The Scatter and Gather Memory interfaces are each 16-bits wide and support up to 512 Kbytes of pipelined synchronous SRAM. Each port can accept accesses from any of the on-chip masters (Circuit Interface, Data Mover Units, or Port2 Interface). Because the Scatter Memory Controller and Gather Memory Controller are independent, operations to these ports can occur simultaneously.

SCATTER MEMORY CONTROLLER

The Scatter Memory Controller connects the Scatter Data Mover Unit, the Circuit Interface, and the Port2 Interface to up to 512 Kbytes of SRAM storage. The Scatter Memory starts at location 0x00000 and can address 128, 256 or 512 Kbytes depending on the number of SRAM parts connected to the Scatter Memory port of the MXT3020. The controller supports one or two 64Kx16 parts, one or two 128Kx16 parts, or one 256Kx16 part.

The Scatter Memory Controller services four different functional units within the MXT3020. Only three units write data into the Scatter Memory: the Port2 Interface, the Scatter Data Mover Unit, and the quiet logic. The TDM transmit section of the Circuit Interface reads this space, and the Port2 Interface can read this space as well. The arbitration is priority-based where the TDM transmit section of the Circuit Interface, quiet logic, and Port2 interfaces have highest priority followed by the Data Mover Unit. A lower priority device's transfer is preempted by a higher priority device, but continues after the higher priority device has completed its transfer.

Restriction on Scatter Memory Usage
Do not issue read operations to Scatter Memory that cross memory banks.

GATHER MEMORY CONTROLLER

The Gather Memory Controller connects the gather Data Mover Unit, the List RAMs, the Circuit Interface, and the Port2 Interface with up to 511 Kbytes[1] of SRAM storage. The Gather Memory starts at located 0x80000 and can address 128 Kbytes or 256 Kbytes depending on the SRAM parts connected to the Gather Memory port of the MXT3020. The controller supports one or two 64Kx16 parts, one or two 128Kx16 parts, or one 256Kx16 part.

The Gather Memory Controller services six different functional units within the MXT3020. All six read data from the Gather Memory but only two, the Port2 Interface, and the TDM receive section of the Circuit Interface, write this space. The arbitration is priority-based where the TDM receive section of the Circuit Interface and Port2 interfaces have highest priority followed by the List RAMs and then the Data Mover Unit. A lower priority device's transfer is preempted by a higher priority device, but continues after the higher priority device has completed its transfer. .

Restriction on Gather Memory Usage
Do not issue read operations to Gather Memory that cross memory banks.

1. The last 1K bytes of address space (0xFFC00 - 0xFFFFF) is mapped to the MXT3020 internal RAMs and registers and is not accessible.

DETERMINING SCATTER AND GATHER MEMORY REQUIREMENTS

Scatter Memory

The Scatter Memory size is determined by how much CDV is required and the speed of the links. T1/E1 links use 32 DS0s per frame per link in frame storage. Storing 128 frames of ATM-to-TDM data (scatter), provides 128 frames times 125 µsec per frame times 0.5[1], or 8 msec of CDV tolerance. The MXT3020 can support up to 512 frames, or 32 msec of CDV tolerance (without buffering in Port1 memory). Thus, the memory used for scatter is number of frames times the number of DS0s per frame times 1 byte per DS0 times the number of links. For example, using 512 frames with T1/E1, would require 512 x 32x 1 x 8=128K of Scatter Memory. The following table summarizes these results:

CDV Tolerance	*Scatter Memory Size (T1/E1)*
8 msec (128 frames)	32K
16 msec (256 frames)	64K
32 msec (512 frames)	128K

Gather Memory

Gather Memory requirements include the following:

- memory for frame storage
- 2K for quiet logic data,
- 2K for tri-state control maps, and
- memory for scatter/gather maps.

1. The 0.5 factor is included because the CDV calculation is relative to the center of the buffer.

The memory requirement for gather frame storage is calculated in a fashion similar to that for scatter frame storage, but less memory is required because fewer frames need to be stored in the TDM-to-ATM direction.

The calculation for scatter/gather maps depends on number of VC's and the alignment of maps. For bi-directional connections, both a scatter and a gather map are needed. Assuming a worst case alignment based on N=32, maps must be aligned on 128-byte boundaries in gather memory. Since both scatter and gather maps are maintained in gather memory, there are two maps per connection and a requirement for 256-byte gather memory blocks per bi-directional connection. If all of the connections on an 8 E1 MXT3020 were N=1, then one would use 8 x 32 x 256 bytes = 64 Kbytes for signal maps in this example. If the application used 128 frames of frame storage (the minimum the software assumes), the frame storage would use 128 x 32 x 1 x 8 = 32K for frame storage. Consequently, the total Gather Memory requirement would be 32K+2K+2K+64K, which is less than 128K.

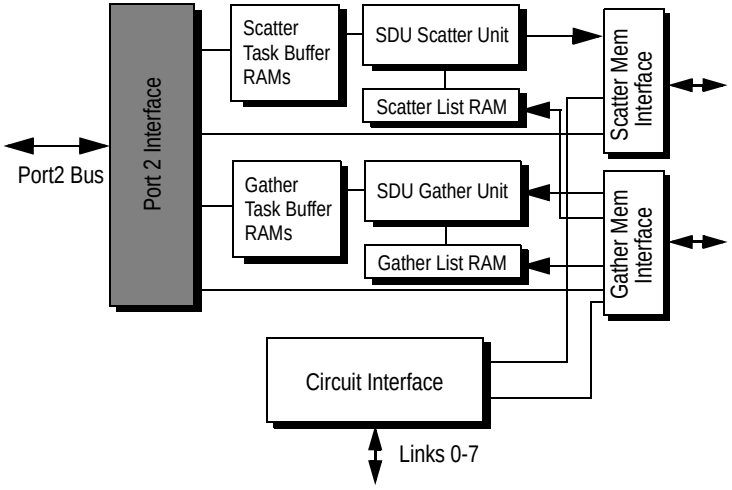
INTERFACE PINS

The MXT3020 provides a complete interface to the scatter and gather memories, which are typically implemented with two 64Kx16, one 128Kx16 or two 128Kx16 3.3 volt synchronous SRAMs.

TABLE 14. Scatter and Gather Memory Interface Pins

Pin	*Function*	*Drive*
GATH_ADDR[18:1]	Gather Memory Address lines	Output
GATH_DATA[15:0]	Gather Memory Data lines	I/O
GATH_WE[1:0]	Gather Memory Write Enables	Output
GATH_OE[1:0]	Gather Memory Output Enables	Output
SCAT_ADDR[18:1]	Scatter Memory Address lines	Output
SCAT_DATA[15:0]	Scatter Memory Data lines	I/O
SCAT_WE[1:0]	Scatter Memory Write Enables	Output
SCAT_OE[1:0]	Scatter Memory Output Enables	Output

CHAPTER 5 *Port2 Interface*



The Port2 Interface is the MXT3020's connection to the MXT3010 device. Through the Port2 Interface, the MXT3010 can read and write Scatter and Gather memory, the Task Buffer RAMs, List RAMs, and all internal registers. Once a Port2 transfer begins, the MXT3020's Port2 Interface demultiplexes the Port2 bus and decodes the address. If the address maps to this chip, the MXT3020 performs the read or write operation to the register or RAM selected by the address.

MXT3020 ADDRESSING

An MXT3020 requires 1 Mbyte of Port2 (P2) address space. Thus, a maximum of two MXT3020s can reside on a P2 bus unless additional address selection logic is provided (see “Multiple MXT3020 implementation” on page 130).

Strapping the P2A20 pin

The P2A20 pin of one MXT3020 is strapped low to decode P2 bus address space 0x000000 to 0x0FFFFF while the P2A20 pin of the other MXT3020 is strapped high to decode P2 address space 0x100000 to 0x1FFFFF.

Slave-only connection

The Port2 Interface provides a slave-only connection to the P2 bus; therefore, all data transfers to and from the MXT3020 are initiated by the MXT3010.

Burst mode transfers

With one exception, the MXT3020 responds only to burst transfers on the P2 bus. This allows the MXT3020 to coexist with non-burst devices on the P2 bus using the same address space. The one exception is a non-burst read of the Scatter/Gather Task Buffer Busy flags. See “Scatter/Gather Task Buffer Busy flags” on page 134.

Restriction on P2 access

MXT3020 internal registers can be written or read only one 16-bit halfword at a time.

Address Space

The breakdown of the MXT3020 address space is shown in Figure 31 on page 83.

MXT3020 ADDRESS SPACE

FIGURE 31 MXT3020 address space

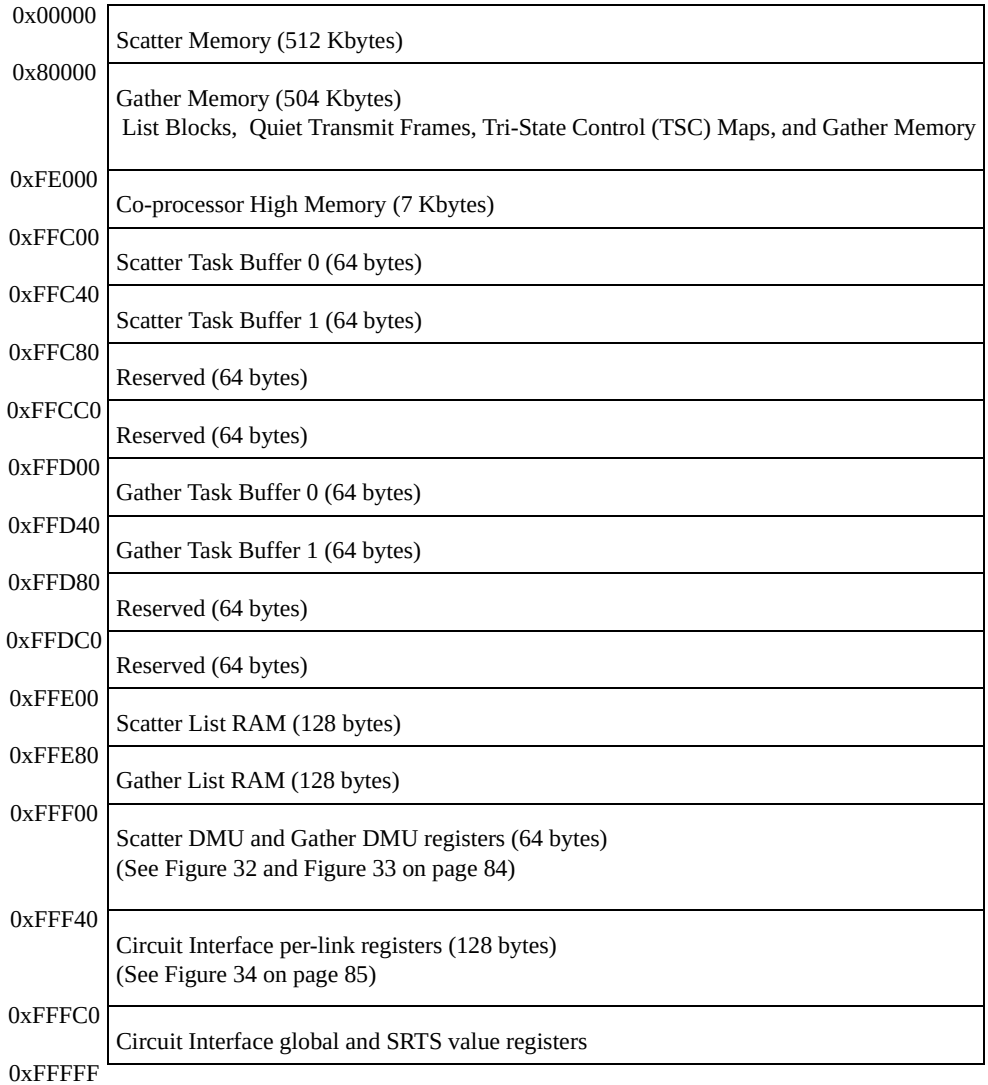


FIGURE 32 Scatter Data Mover Unit Register Map

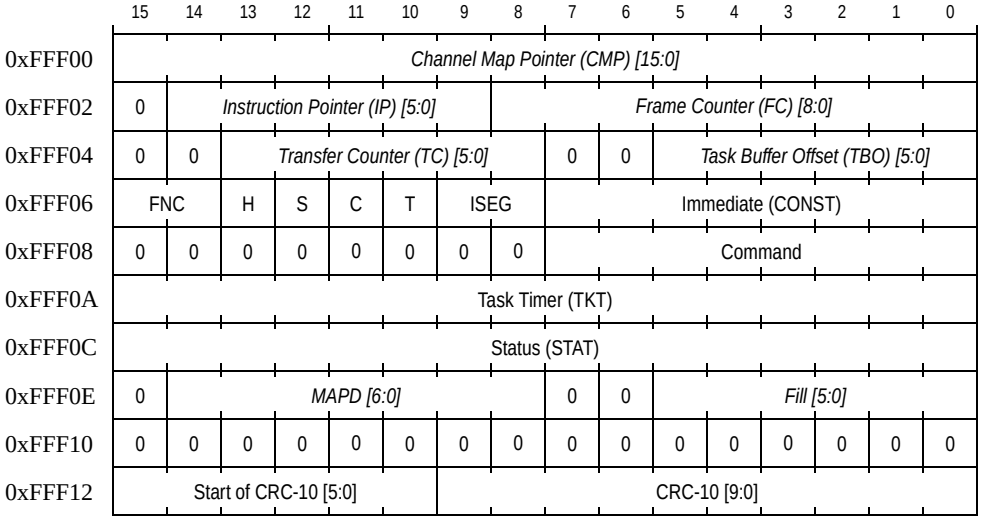


FIGURE 33 Gather Data Mover Unit Register Map

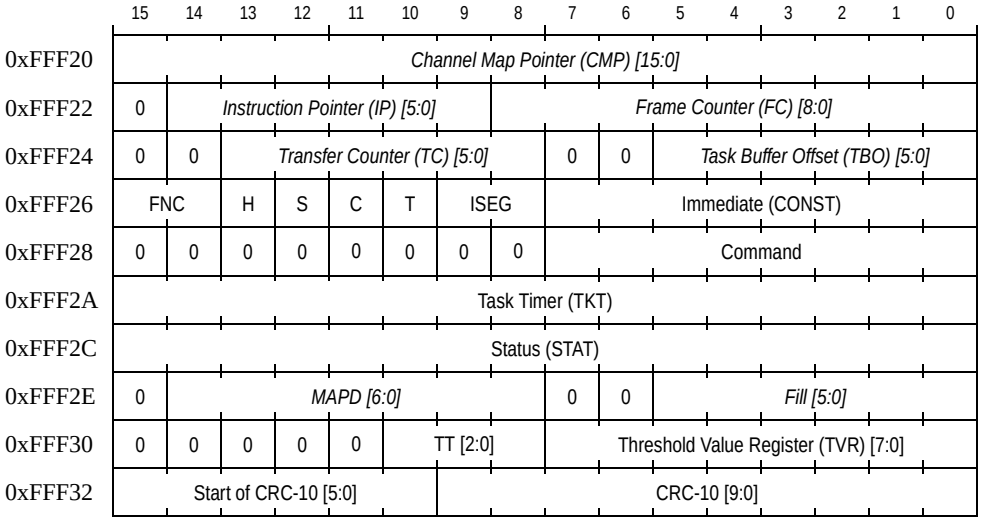


FIGURE 34 Circuit Interface per-link registers

	TDM Link 0 - 3 Register Map
0xFFF40	Link 0 Configuration
0xFFF42	Link 0 Rx Buffer Address counter
0xFFF44	Link 0 Tx Buffer Address counter
0xFFF46	Link 0 TSC Map Address counter
0xFFF48	Link 0 Service Clock N register
0xFFF4A	Link 0 Service Clock K register
0xFFF4C	Link 0 Service Clock L counter
0xFFF4E	Link 0 SRTS FTC read back
0xFFF50	Link 1 Configuration
0xFFF52	Link 1 Rx Buffer Address counter
0xFFF54	Link 1 Tx Buffer Address counter
0xFFF56	Link 1 TSC Map Address counter
0xFFF58	Link 1 Service Clock N register
0xFFF5A	Link 1 Service Clock K register
0xFFF5C	Link 1 Service Clock L counter
0xFFF5E	Link 1 SRTS FTC read back
0xFFF60	Link 2 Configuration
0xFFF62	Link 2 Rx Buffer Address counter
0xFFF64	Link 2 Tx Buffer Address counter
0xFFF66	Link 2 TSC Map Address counter
0xFFF68	Link 2 Service Clock N register
0xFFF6A	Link 2 Service Clock K register
0xFFF6C	Link 2 Service Clock L counter
0xFFF6E	Link 2 SRTS FTC read back
0xFFF70	Link 3 Configuration
0xFFF72	Link 3 Rx Buffer Address counter
0xFFF74	Link 3 Tx Buffer Address counter
0xFFF76	Link 3 TSC Map Address counter
0xFFF78	Link 3 Service Clock N register
0xFFF7A	Link 3 Service Clock K register
0xFFF7C	Link 3 Service Clock L counter
0xFFF7E	Link 3 SRTS FTC read back

	TDM Link 4 - 7 Register Map
0xFFF80	Link 4 Configuration
0xFFF82	Link 4 Rx Buffer Address counter
0xFFF84	Link 4 Tx Buffer Address counter
0xFFF86	Link 4 TSC Map Address counter
0xFFF88	Link 4 Service Clock N register
0xFFF8A	Link 4 Service Clock K register
0xFFF8C	Link 4 Service Clock L counter
0xFFF8E	Link 4 SRTS FTC read back
0xFFF90	Link 5 Configuration
0xFFF92	Link 5 Rx Buffer Address counter
0xFFF94	Link 5 Tx Buffer Address counter
0xFFF96	Link 5 TSC Map Address counter
0xFFF98	Link 5 Service Clock N register
0xFFF9A	Link 5 Service Clock K register
0xFFF9C	Link 5 Service Clock L counter
0xFFF9E	Link 5 SRTS FTC read back
0xFFFA0	Link 6 Configuration
0xFFFA2	Link 6 Rx Buffer Address counter
0xFFFA4	Link 6 Tx Buffer Address counter
0xFFFA6	Link 6 TSC Map Address counter
0xFFFA8	Link 6 Service Clock N register
0xFFFAA	Link 6 Service Clock K register
0xFFFAC	Link 6 Service Clock L counter
0xFFFAE	Link 6 SRTS FTC read back
0xFFFB0	Link 7 Configuration
0xFFFB2	Link 7 Rx Buffer Address counter
0xFFFB4	Link 7 Tx Buffer Address counter
0xFFFB6	Link 7 TSC Map Address counter
0xFFFB8	Link 7 Service Clock N register
0xFFFBA	Link 7 Service Clock K register
0xFFFBC	Link 7 Service Clock L counter
0xFFFBE	Link 7 SRTS FTC read back

FIGURE 35 Circuit Interface global and SRTS value registers

0xFFFC0	CI Tri-state Control Base Address register
0xFFFC2	CI Configuration register
0xFFFC4	CI Quiet Frame Base Address register
0xFFFC6	CI SRTS FTC register
0xFFFC8	CI SRTS Valid Status register
0xFFFCA	CI Status register
0xFFFCC	Reserved
0xFFFCE	Reserved
0xFFFD0	Link 0 Local SRTS Value
0xFFFD2	Link 1 Local SRTS Value
0xFFFD4	Link 2 Local SRTS Value
0xFFFD6	Link 3 Local SRTS Value
0xFFFD8	Link 4 Local SRTS Value
0xFFFDA	Link 5 Local SRTS Value
0xFFFDC	Link 6 Local SRTS Value
0xFFFDE	Link 7 Local SRTS Value
0xFFFE0	Reserved

INTERFACE PINS

TABLE 15. Port2 Interface pins

Pin	*Direction*	*Function*
P2AD[15:0]	I/O	Port 2 Address/Data lines
P2AI[3:0]	I	Port 2 Address Index Bus
P2RQ_	I	Port 2 Request
P2RD	I/O	Port 2 Read/Write Select
P2END_	I	Port 2 End
P2QBRST	I	Port 2 Burst
P2TRDY_	O	Port 2 Target Ready
P2ASEL_	O	Port 2 Address Select
P2A20	I	Port 2 Address bit [20] Polarity

CHAPTER 6 *Register Reference*

This chapter contains reference information on each of the following registers and multi-bit fields within those registers:

Register or multi-bit field
“Link Configuration register” on page 103
“Link FTC counter” on page 105
“Link Service Clock K register” on page 107
“Link Service Clock L counter” on page 108
“Link Service Clock N register” on page 109
“LINK_SRTS_FTC” on page 105
“Link SRTS Value register” on page 110
“Link Tri-state Control Address counter” on page 111
“Link Tx Buffer Address counter” on page 112
“List Size and Frame Number” on page 113
“LSC_K” on page 107
“LSC_L” on page 108
“LSC_N” on page 109
“MAPD and FILL registers” on page 114
“MAXDS0” on page 91
“MAXRBS” on page 91
“MAXTBS” on page 91
“QUIET_FRAME” on page 92
“RX_ADR” on page 106
“SAR SDU registers” on page 115
“SAR Size and SAR Offset” on page 116
“SRTS_VALUE” on page 110
“Start of Cell (SOC)” on page 100
“Start of CRC-10” on page 98
“Status register” on page 118
“Task Buffer Offset (TBO)” on page 120
“Task Timer register” on page 119
“TBR_BSY” on page 118
“Transfer Counter and Task Buffer Offset registers” on page 120
“Transfer Counter (TC)” on page 120
“TSC_BASE” on page 96
“TSCNT” on page 111
“TT register” on page 117
“TVR register” on page 117
“TX_ADR” on page 112
“TxCLKS” on page 103

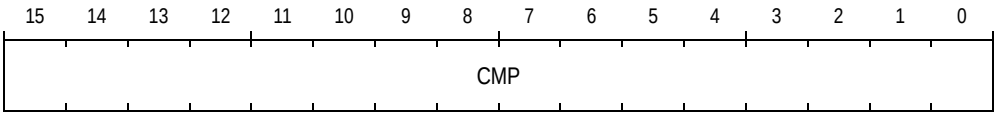
CHANNEL MAP POINTER (TBR AND DMU)

When combined with the contents of the ISEG register (see “Immediate and Control Flags register (TBR and DMU)” on page 99), this 16-bit field points to an instruction in a list block located in Gather Memory. The bits represent Gather Memory address bits [16:1]. This pointer is loaded from a Task Buffer RAM into the Channel Map-Pointer register in a DMU.

Locations: 0xFFC00 (Scatter Task Buffer 0)
0xFFC40 (Scatter Task Buffer 1)
0xFFD00 (Gather Task Buffer 0)
0xFFD40 (Gather Task Buffer 1)
0xFFF00 (Scatter DMU)
0xFFF20 (Gather DMU)

Reset Value: 0000 0000 0000 0000

Format in Task Buffer RAMs and DMUs:



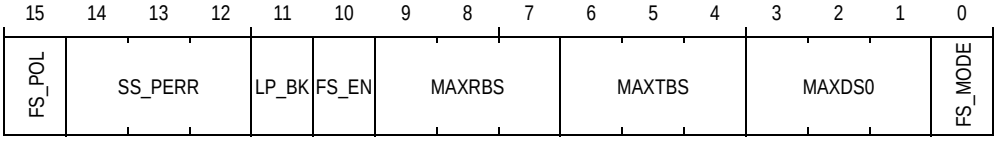
Bit	*Name*	*Function*	*Additional information*
15-0	CMP	When combined with the contents of the ISEG[1:0] register, this 16-bit field points to an instruction in a list block located in Gather Memory. The bits represent Gather Memory address bits [16:1].	

CI CONFIGURATION REGISTER

This register controls the operating mode of the Circuit Interface. There is only one CI Configuration register; it is a read/write register.

Location: 0xFFFC2

Reset Value: 0000 0000 0000 0001



Bit	*Name*	*Function*	*Additional information*
15	FS_POL	This bit controls the interpretation of the polarity of the Frame Sync input. 0 FSYNC Active Level is 1 1 FSYNC Active Level is 0	
14-12	SS_PERR	These bits control the TDM Request Snap Shot Period. They determine the rate at which link requests for the Gather or Scatter Memories are sampled. A setting of 010 causes the sample rate to be 30 times the system clock and is the recommended setting. At this setting, the following combinations of clock speed and link speed are valid: System Clock / Max Uni Link Speed / Max Bi Link Speed 33 MHz / 8 MHz / 4 MHz 40 MHz / 8 MHz / 6 MHz 50 MHz / 8 MHz / 8 MHz	
11	LP_BK	This bit enables link loopback mode. 0 Normal operating Mode 1 In unidirectional mode, all links will loop their ASER transmit output back to BSER receive input. In bidirectional mode, the ASER transmit output is looped back to the ASER receive input, and the BSER transmit output is looped back to the BSER receive input.	

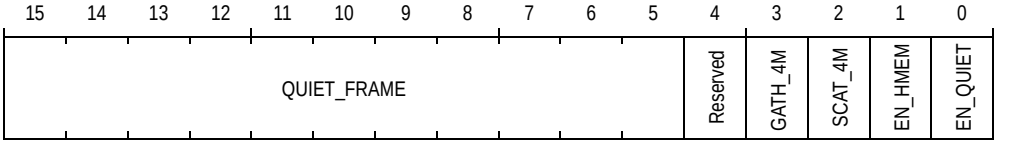
Bit	*Name*	*Function*	*Additional information*
10	FS_EN	This bit enables/disables Frame Sync detection. This bit only applies to SDT mode links. UDT mode links ignore this bit. To cause SDT links to acquire Frame Sync simultaneously, first clear their LkRESET bits (“Link Configuration register” on page 103) to zero (0) and then assert this global bit. 0 Disable Frame Sync acquisition (This disables all SDT mode links.) 1 Enable Frame Sync acquisition (This enables all SDT mode links.)	
9-7	MAXRBS	These bits specify the maximum amount of TDM to ATM receive frame storage allocated for any link in the system. 0 64 frames 1 128 frames 2 256 frames 3 512 frames	
6-4	MAXTBS	These bits specify the maximum amount of ATM to TDM transmit frame storage allocated for any link in the system. 0 64 frames 1 128 frames 2 256 frames 3 512 frames	
3-1	MAXDS0	These bits specify the maximum number of DS0's per frame for any link in the system. 0 24 1 32 2 64 3 96 4 128	
0	FS_MODE	This bit determines the frame sync source for all of the links. This bit must be cleared to zero (0) when any links are in the bidirectional mode. 0 Frame Sync from Frame Sync pin 1 Frame Sync from link BSCE pin	

CI QUIET FRAME BASE ADDRESS REGISTER

An 11-bit value in this register defines the base address of the Quiet Frame in Gather Memory. Additional bits indicate the type of SRAM used, control access to the high memory space (FE000 - FFC00), and enable the quiet logic. There is only one CI Quiet Frame Base Address register; it is a read/write register.

Location: 0xFFFC4

Reset Value: 0000 0000 0000 0000



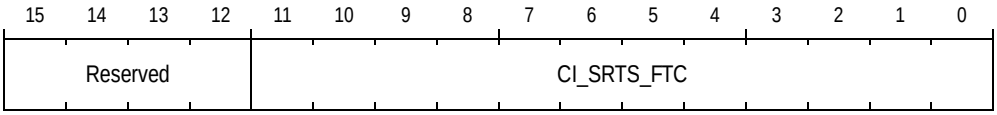
Bit	*Name*	*Function*	*Additional Information*
15-5	QUIET_FRAME	These bits define the base address of the Quiet Frame in Gather Memory.	"Quiet Logic" on page 43
4	Reserved		
3	GATH_4M	Set this bit to one (1) if 256K x 16 (4Mb) SRAMs are used to implement Gather Memory. Otherwise, write as zero (0).	"GATH_4M (Bit 3)" on page 42
2	SCAT_4M	Set this bit to one (1) if 256K x 16 (4Mb) SRAMs are used to implement Scatter Memory. Otherwise, write as zero (0).	"SCAT_4M (Bit 2)" on page 42
1	EN_HMEM	This bit enables MXT3020 response to Port 2 access in the Co-Processor High Memory Space (i.e. FE000 - FFC00). 0 Access to high memory is disabled 1 Access to high memory is enabled	
0	EN_QUIET	This bit enables the quiet logic. 0 Quiet logic is disabled 1 Quiet logic is enabled	"Quiet Logic" on page 43

CI SRTS FTC REGISTER

The value in the CI SRTS FTC register represents the number of service clock[1] periods between updates of the Link SRTS Value register (see “Link SRTS Value register” on page 110). Whenever a Link FTC counter counts down to zero, it is reloaded from the CI SRTS FTC register, which is typically set to 3008. There is only one CI SRTS FTC register; it is a read/write register..

Location: 0xFFFC6

Reset Value: 0000 0000 0000 0000



Bit	*Name*	*Function*	*Additional information*
15-12	Reserved		
11-0	CI_SRTS_FTC	These bits are the number of service clock periods between Local SRTS samples of the network clock counter. Service clock periods are counted by FTC counters in each link. The bits in this register are used to refresh those counters.	Figure 17 on page 33

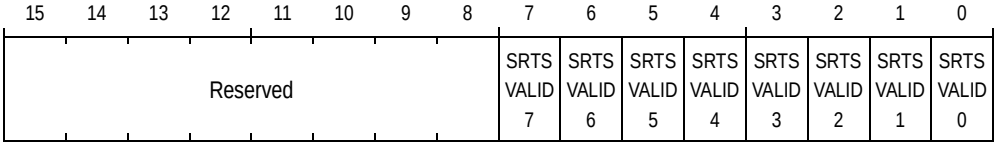
1. The service clock rate for a T1 facility is 1.544 MHz.

CI SRTS VALID STATUS REGISTER

This status register indicates whether a link has a valid SRTS value latched. When a link's SRTS value is read by the SWAN processor in the MXT3010, its respective valid bit is cleared. There is only one CI SRTS Valid Status register; it is a read-only register..

Location: 0xFFFC8

Reset Value: 0000 0000 0000 0000



Bit	*Name*	*Function*	*Additional information*
15-8	Reserved		
7-0	SRTS_VALID *n*	These bits indicate, on a bit-per-link basis, whether the link corresponding to that bit has a valid SRTS value latched. 0 A valid SRTS value has not been latched 1 A valid SRTS value has been latched	"CI Status register" on page 46

CI STATUS REGISTER

Bits in this register indicate, for each link, whether that link has acquired Frame Sync or has lost Frame Sync. There is only one CI Status register; it is a read-only register..

Location: 0xFFFCA

Reset Value: 0000 0000 0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACQ7	LOST7	ACQ6	LOST6	ACQ5	LOST5	ACQ4	LOST4	ACQ3	LOST3	ACQ2	LOST2	ACQ1	LOST1	ACQ0	LOST0

Bit	***Name***	***Function***		***Additional information***
15, 13, 11, 9, 7, 5, 3, 1	ACQ *n*	0	Link *n* has not acquired Frame Sync	
		1	Link *n* has acquired Frame Sync	
14, 12, 10, 8, 6, 4, 2, 0	LOST *n*	0	Link *n* has not lost Frame Sync	
		1	Link *n* has lost Frame Sync after acquiring it. When this bit sets, the corresponding ACQ bit is cleared. This bit is cleared by Link Reset or Chip Reset. If a LOST bit is a zero (0), link n has not lost Frame Sync.	

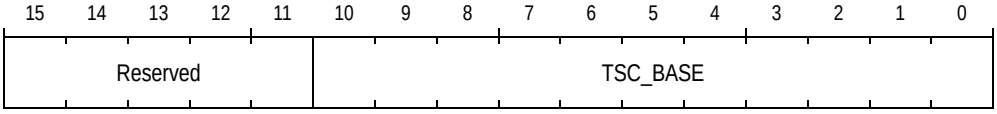
Note: *n* represents the link number, where $0 \leq n \leq 7$.

CI TRI-STATE CONTROL BASE ADDRESS REGISTER

The contents of this register and the Tri-state Control Address counter are combined with the Link ID number (LID) to create an 18-bit pointer to a halfword-aligned entry in Gather Memory containing the tri-state control map for the link. There is only one CI Tri-state Control Base Address register; it is a read/write register..

Location: 0xFFFC0

Reset Value: 0000 0000 0000 0000



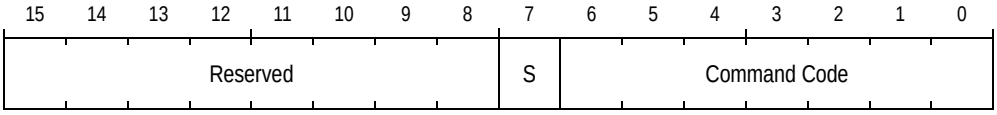
Bit	*Name*	*Function*	*Additional information*
15-11	Reserved		
10-0	TSC_BASE	These bits define the base address of the Tri-state Control table in Gather Memory.	“Creation of the tri-state control map pointer” on page 28

COMMAND REGISTER

This register is part of the DMUs. It controls the Task Timer and the DMU Halt Flag..

Locations: 0xFFF08 (Scatter DMU)
0xFFF28 (Gather DMU)

Reset Value: 0000 0000 0000 0000



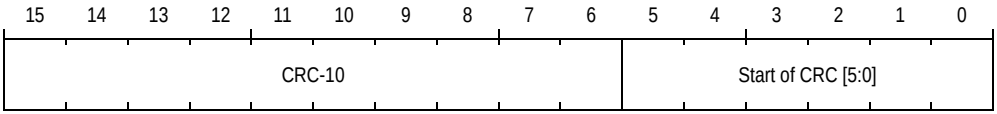
Bit	***Name***	***Function***	***Additional information***
15-8	Reserved		
7-0	S-bit and Command Code	This bit and field contains the command to be executed by the DMU. Bits 7 654 3210 1 000 0000 Halt 0 000 0000 Continue 1 000 0001 Disable Task Timer 0 000 0001 Enable Task Timer All other codes are reserved	

CRC-10 REGISTER

This register is part of the DMUs. It controls the starting point of the CRC-10 accummulation process and contains the accumulated CRC-10 partial result.

Locations: 0xFFF12 (Scatter DMU)
0xFFF32 (Gather DMU)

Reset Value: 0000 0000 0000 0000



Bit	*Name*	*Function*	*Additional information*
15-6	CRC-10	The value loaded into this field before a scatter/gather operation is used as the initial or seed value for CRC-10 calculations. The MXT3020 clears this field at the end of a scatter/gather if CRC-10 was enabled and a function code (FNC) of 01 or 11 was selected. At all other times, this field is updated with the CRC-10 partial result.	
5-0	Start of CRC-10	This field indicates the number of bytes that are to be processed before CRC-10 calculation begins. Any value from 0 to 63 may be selected.	

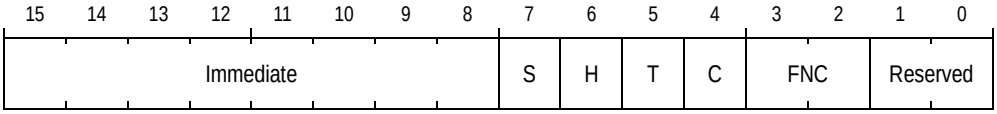
IMMEDIATE AND CONTROL FLAGS REGISTER (TBR AND DMU)

At the conclusion of the DMA that loads the Task Buffer RAM, the Immediate information is automatically loaded into the CONST register in the DMU and the Control Flags information is automatically loaded into the Control Flags in the DMU

Locations: 0xFFC06 (Scatter Task Buffer 0)
0xFFC46 (Scatter Task Buffer 1)
0xFFD06 (Gather Task Buffer 0)
0xFFD46 (Gather Task Buffer 1)
0xFFF06 (Scatter DMU)
0xFFF26 (Gather DMU)

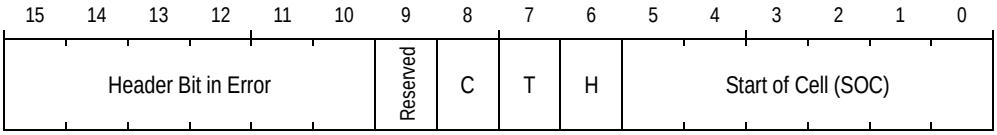
Reset Value: 0000 0000 0000 0000

Pre-Scatter/Gather Task Buffer RAM format



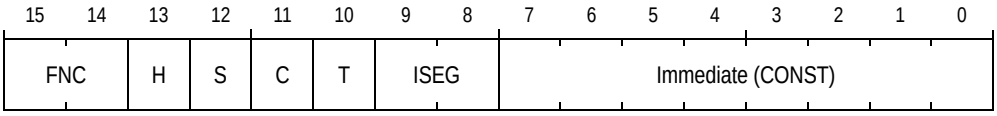
Bit	*Name*	*Function when written*		*Additional Information*
15-8	Immediate	Data for use by gather immediate, scatter immediate and mcast immediate instructions. This register must be loaded to initiate DMU operation.		
7	S	0 1	Scrambling/descrambling is disabled. Scrambling/descrambling is enabled.	
6	H	0 1	HEC generation/checking is disabled. HEC generation/checking is enabled.	
5	T	0 1	Threshold Test is disabled Threshold Test is enabled	
4	C	0 1	CRC-10 operation is disabled CRC-10 operation is enabled	
3-2	FNC	00 01 10 11	Check CRC and save partial result Check CRC and discard Generate CRC and save partial result Generate CRC and append to transmission	00 & 10 used mid-message; 01 & 11 at end
1-0	Reserved			

Post-Scatter/Gather Task Buffer RAM format:



Bit	***Name***	***Function when read***		***Additional Information***
15-10	Header Bit in Error	If HEC checking is enabled and a header bit error occurs, this field identifies which bit in the 5-byte header was received in error. Bits 15-13 indicate the byte in error; bits 12-10 indicate the bit in error within the byte. If no header bit error is detected, this register will return the value 3F.		
9	Reserved	Reserved - reads as zero (0).		
8	C	0 1	There were no CRC-10 errors detected. A CRC-10 error was detected.	
7	T	0 1	Threshold Test was successful Threshold Test checking failed	
6	H	0 1	There were no HEC errors detected. An HEC error was detected; see bits 15-10	
5-0	SOC (read)	Number of the gathered byte in which the HEC code (indicating the Start Of Cell (SOC)) was found.		

Data Mover Unit format



Bit	*Name*	*Function*		*Additional Information*
15-14	FNC	00 01 10 11	Check CRC and save partial result Check CRC and discard Generate CRC and save partial result Generate CRC and append to transmission	00 & 10 used mid-message; 01 & 11 at end
13	H	0 1	HEC generation/checking is disabled. HEC generation/checking is enabled.	
12	S	0 1	Scrambling/descrambling is disabled. Scrambling/descrambling is enabled.	
11	C	0 1	CRC-10 operation is disabled CRC-10 operation is enabled	
10	T	0 1	Threshold Test is disabled Threshold Test is enabled	
9-8	ISEG	Bits [18:17] of the List Block address in Gather Memory. These bits are appended to the Channel Map Pointer and must be initialized by the MXT3010.		
7-0	Immediate (CONST)	Data for use by gather immediate, scatter immediate and mcast immediate instructions.		

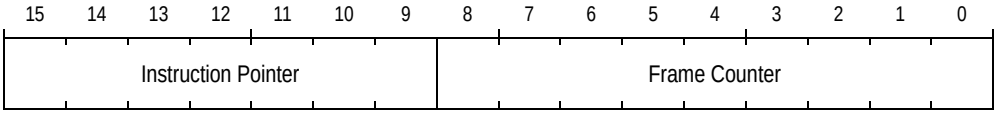
Note: All of the bits in this register are internal state bits used by the Data Mover Unit (DMU). If desired, they can be accessed for debug purposes, but they are not normally accessed by programs. Rather, the DMU is programmed by loading the Task Buffer RAMs.

INSTRUCTION POINTER AND FRAME COUNTER

This is a register in the DMUs.

Locations: 0xFFF02 (Scatter DMU)
0xFFF22(Gather DMU)

Reset Value: 0000 0000 0000 0000



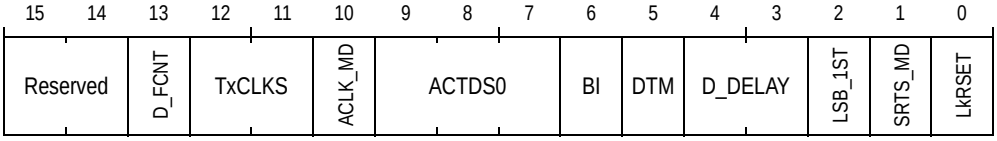
Bit	*Name*	*Function*	*Additional information*
15-9	Instruction Pointer	This is a 7-bit counter that points to the next location to be read in the List RAM. Bits [15:9] represent Instruction Pointer bits [6:0]. Since the List RAMs are 64x16, only bits [5:0] are required for addressing, and bit [6] is always zero.	
8-0	Frame Counter	This is a 9-bit counter identifying the data frame within a Link Buffer in Scatter/Gather Memory.	

LINK CONFIGURATION REGISTER

This register controls the operating mode of each link pair. There is one of these read/write registers for each link pair.

Location: 0xFFF*n*0 (*n* = 4 + Link#)

Reset Value: 0000 0000 0000 0001



Bit	***Name***	***Function***	***Additional Information***
15-14	Reserved		
13	D_FCNT	This bit enables/disables incrementing the frame count portion 0 Enabled 1 Disabled	
12-11	TxCLKS	These bits select the link transmitter clock source 0 Internal SRTS Clock 1 External RxCLK Input 2 External TxCLK Input	"Added detail: Bits 12-10 – clock source control" on page 15
10	ACLK_MD	This bit selects the ACLK pin I/O direction 0 Input 1 Output	"Added detail: Bits 12-10 – clock source control" on page 15
9-7	ACTDS0	These bits indicate the Actual # of DS0's per frame. This information is used by the tri-state enable control map. 0 24[a] 1 32 2 64 3 96 4 128 (5-7 are reserved)	"The tri-state enable control map" on page 28

Bit	Name	Function	Additional Information
6	BI	This bit controls the modes of a link pair. 0 Unidirectional mode 1 Bidirectional mode	"Added detail: Bit 6 – link pair mode" on page 19
5	DTM	This bit selects the Data Transfer Mode of the link pair as shown below: 0 Unstructured Data Transfer (UDT) mode 1 Structured Data Transfer (SDT) mode	
4-3	D_DELAY	These bits control the position of Frame Sync relative to the first bit of a frame on both reception and transmission. Frame Sync to First Data Delay 0 1 cycle 1 2 cycles 2 3 cycles	
2	LSB_1ST	This bit controls the direction of the link shift registers, selecting whether the most significant bit (MSB) or the least significant bit (LSB) is shifted into the serial line first. 0 MSB first 1 LSB first	
1	SRTS_MD	This bit controls the mode of the SRTS value generator. 0 Slave mode 1 Master mode	"DTM (bit 5)" on page 13
0	LkRESET	This bit controls the reset state of a link pair. 0 Removes the link from the reset state 1 Places the link in the reset state	

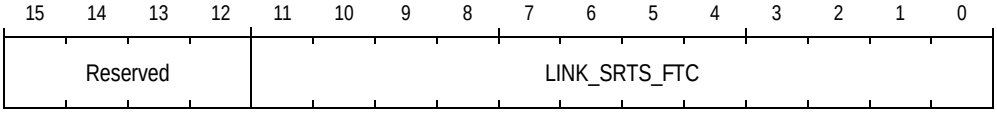
a. In DS0 = 24 mode, the MXT3020 expects a 193-bit frame. It strips the in-band Frame Sync pulse from this frame before forming DS0s in SDT mode.

LINK FTC COUNTER

The SRTS value generator for each link creates an SRTS value by using an FTC counter to count a designated number of service periods. A register common to all links (see “CI SRTS FTC register” on page 93) establishes the number of service clock periods to be counted. This number is generally 3008 (decimal). There is one readable Link FTC counter for each link (see “Link FTC read back” entries in Figure 34, “Circuit Interface per-link registers,” on page 85). To change the count used by the Link FTC counter, write the CI SRTS FTC register (page 93).

Location: 0xFFF*n*E (*n* = 4 + Link #)

Reset Value: 0000 0000 0000 0000



Bit	***Name***	***Function***	***Additional Information***
15-12	Reserved		
11-0	LINK_SRTS_FTC	These bits specify the number of service clock periods that elapse before a new SRTS value is produced.	“Link service clock generation registers” on page 33

LINK RX BUFFER ADDRESS COUNTER

This register indicates where the next data received on this TDM link pair will be stored in the Gather Memory. The contents of this counter combined with the link number provide an 18-bit pointer to a halfword-aligned circular queue in Gather Memory where received data can be written by the Circuit Interface logic.

Location: 0xFFF*n*2 (*n* = 4 + Link #)

Reset Value: 0000 0000 0000 0000

Bit	*Name*	*Function*	*Additional Information*
15-1	RX_ADR	These bits contain the Link Rx Buffer Address counter for this serial pair. These bits correspond to the DS0 and FC fields in the figures shown in "Scatter/Gather Memory Address Generation" on page 68.	"Details of the Link Tx/Rx Buffer Address counters" on page 25
0	Reserved		

.

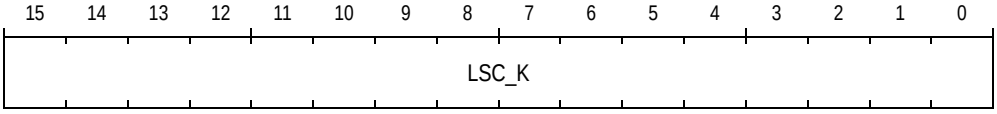
Restriction on Link Rx Buffer Address Counter
Values in this register can only be modified when the link is in the reset state. This restriction does not affect the automatic incrementing that the MXT3020 performs during data transfers.

LINK SERVICE CLOCK K REGISTER

The value in this register represents the fraction control numerator used to generate the link service clock. This is the fine setting of the numerically controlled oscillator used to generate the service clock.There is one of these read/write registers for each link.

Location: 0xFFF*n*A(*n* = 4 + Link #)

Reset Value: 0000 0000 0000 0000



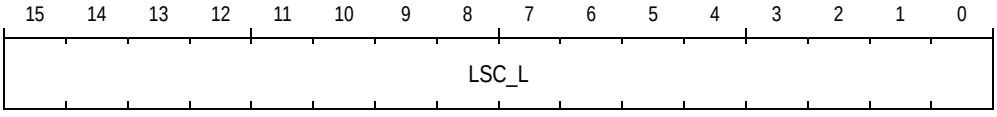
Bit	*Name*	*Function*	*Additional information*
15-0	LSC_K	These bits contain the fraction control numerator used to generate the link service clock.	“Link service clock generation registers” on page 33

LINK SERVICE CLOCK L COUNTER

The value in this counter represents the fraction control denominator used to generate the link service clock. This is a free running counter that can be read and written for diagnostic purposes and can be seeded with an initial value. However, the value loaded into this counter does not alter the effective denominator value, which is always 65,536 (2^{16}). This counter advances once for each service clock cycle generated. There is one of these counters for each link.

Location: 0xFFF*n*C (*n* = 4 + Link #)

Reset Value: 0000 0000 0000 0000



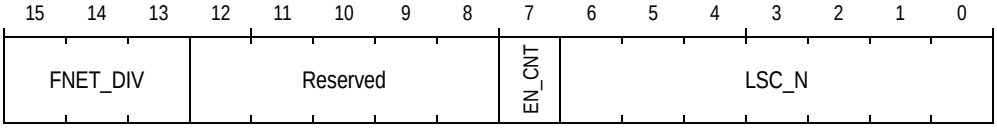
Bit	*Name*	*Function*	*Additional information*
15-0	LSC_L	These bits contain the fraction control denominator used to generate the link service clock.	"Link service clock generation registers" on page 33

LINK SERVICE CLOCK N REGISTER

The value in this register represents the integer multiple of system clock periods used to generate the link service clock. This is the coarse setting of the numerically controlled oscillator used to generate the service clock. There is one of these registers for each link.

Location: 0xFFF*n*8 (*n* = 4 + Link #)

Reset Value: 0000 0000 0000 0000



Bit	*Name*	*Function*		*Additional Information*
15-13	FNET_DIV	000 001 010 011 100 101 110 111	Divide FNET by 1 Divide FNET by 2 Divide FNET by 3 Divide FNET by 4 Divide FNET by 5 Divide FNET by 6 Divide FNET by 7 Divide FNET by 8	"Link Service Clock N register" on page 34
12-8	Reserved			
7	EN_CNT	This bit enables the link service clock counters 0 Disabled 1 Enabled		
6-0	LSC_N	These bits represent the integer multiple of system clock periods used to generate the link service clock. The range of values used is 3 to 127.		"Link service clock generation registers" on page 33

.

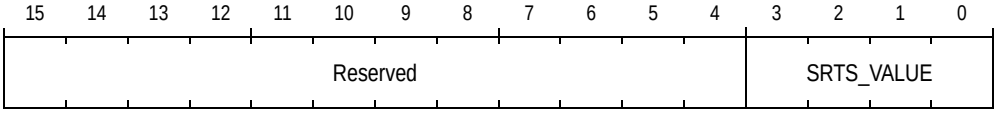
Restriction on FNET_DIV bits
Bits [15:8] of this register are write only and read back as all zeroes. All bits in this register are cleared to zeroes by Reset. Thus, the FNET_DIV bits are write-only and are cleared to zeroes by Reset.

LINK SRTS VALUE REGISTER

Each link has a single SRTS generator controlled by the SRTS_MD bit for that link (see “Channel Map Pointer (TBR and DMU)” on page 89). A 4-bit value is latched into the local SRTS value each time the FTC counter expires (see “Link SRTS Value register” on page 110). There is one of these read/write registers for each link.

Location: 0xFFFD*n* (*n* = Link # *2)

Reset Value: 0000 0000 0000 0000



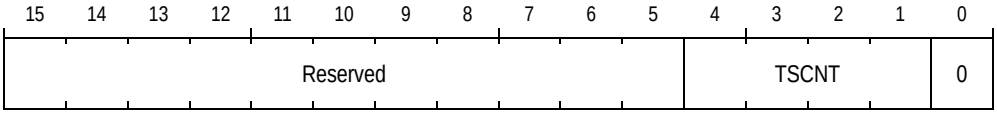
Bit	*Name*	*Function*	*Additional information*
15-4	Reserved		
3-0	SRTS_VALUE	These bits are the new SRTS value.	“Link service clock generation registers” on page 33

LINK TRI-STATE CONTROL ADDRESS COUNTER

The contents of this counter are combined with the Tri-state Control Base Address register and the Link ID number (LID) to create an 18-bit pointer to a halfword-aligned entry in Gather Memory containing the tri-state control map for the link.

Location: 0xFFF*n*6 (*n* = 4 + Link #)

Reset Value: 0000 0000 0000 0000



Bit	*Name*	*Function*	*Additional information*
15-5	Reserved		
4-1	TSCNT	These bits contain the 4-bit Tri-state Control Address counter for this serial pair.	"Creation of the tri-state control map pointer" on page 28
0	Reserved		

LINK TX BUFFER ADDRESS COUNTER

This register indicates the location in Scatter Memory from which the next data to be transmitted on the TDM link will be taken. The contents of this counter combined with the link number provide an 18-bit pointer to a halfword-aligned circular queue in Scatter Memory where transmit link data can be read by the Circuit Interface logic.

Location: 0xFFF*n*4 (*n* = 4 + Link #)

Reset Value: 0000 0000 0000 0000

Bit	*Name*	*Function*	*Additional Information*
15-1	TX_ADR	These bits contain the Link Tx Buffer Address counter for this serial pair. These bits correspond to the DS0 and FC fields in the figures shown in "Creation of the tri-state control map pointer" on page 28.	"Details of the Link Tx/Rx Buffer Address counters" on page 25
0	Reserved		

.

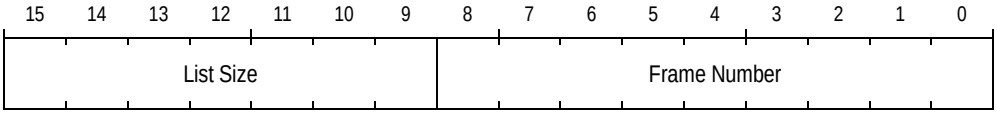
Restrictions on Link Tx Buffer Address Counter
Values written to this register must be 4 higher than those written to the Link Rx Buffer Address counter. Values in this register can only be modified when the link is in the reset state. These restrictions do not affect the automatic incrementing that the MXT3020 performs during data transfers.

LIST SIZE AND FRAME NUMBER

This is a register in the Task Buffer RAM. At the conclusion of the DMA that loads the Task Buffer RAM, the List Size information is automatically loaded into the MAPD register in the DMU and the Frame Number information is automatically loaded into the FC register in the DMU.

Locations: 0xFFC02 (Scatter Task Buffer 0)
0xFFC42 (Scatter Task Buffer 1)
0xFFD06 (Gather Task Buffer 0)
0xFFD42 (Gather Task Buffer 1)

Reset Value: 0000 0000 0000 0000



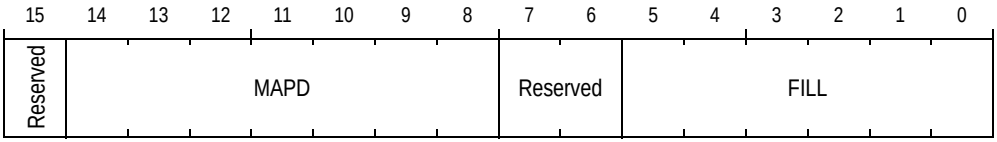
Bit	*Name*	*Function*	*Additional information*
15-9	List Size	This is a 7-bit counter indicating the number of instructions that have been placed in the list block. It should be set to *n*-1, where *n* is the number of instructions in a single copy of the list block. This information is loaded into the MAPD and Fill registers in the DMU, where it is used to determine when the DMU has reached the end of the list block instructions.	
8-0	Frame Number	This is a 9-bit counter identifying the data frame within a Link Buffer in Scatter/Gather Memory. This information is loaded into the Frame Counter (FC) register in the DMU.	

MAPD AND FILL REGISTERS

These registers, part of the DMUs, report on scatter and gather completions. They also control threshold testing.

Locations: 0xFFF0E (Scatter DMU)
0xFFF2E(Gather DMU)

Reset Value: 0000 0000 0000 0000



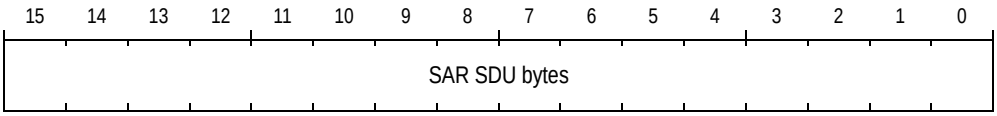
Bit	*Name*	*Function*	*Additional information*
15	Reserved		
14-8	MAPD	This counter, loaded from the List Size in the Task Buffer RAM, is used to determine when the DMU has reached the end of the list block.	
7-6	Reserved		
5-0	FILL	This counter, loaded from the List Size in the Task Buffer RAM, is used to determine when the DMU has reached the end of the list block.	

SAR SDU REGISTERS

These registers are part of the Task Buffer RAM. In a scatter operation, they contain the data to be scattered. In a gather operation, they contain the gathered data..

Locations: 0xFFC3E (Scatter Task Buffer 0)
0xFFC7E (Scatter Task Buffer 1)
0xFFD3E (Gather Task Buffer 0)
0xFFD7E (Gather Task Buffer 1)

Reset Value: 0000 0000 0000 0000



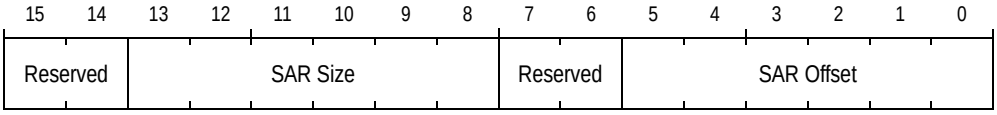
Bit	*Name*	*Function*	*Additional information*
15-0	SAR SDU bytes	SAR SDU bytes	

SAR SIZE AND SAR OFFSET

This register is part of the Task Buffer RAMs. At the conclusion of the DMA that loads a Task Buffer RAM, the SAR Size information is automatically loaded into the Transfer Counter (TC) in the DMU and the SAR Offset information is automatically loaded into the Task Buffer Offset (TBO) register in the DMU.

Locations: 0xFFC04 (Scatter Task Buffer 0)
0xFFC44 (Scatter Task Buffer 1)
0xFFD04 (Gather Task Buffer 0)
0xFFD44 (Gather Task Buffer 1)

Reset Value: 0000 0000 0000 0000



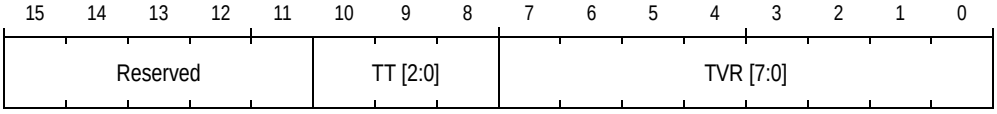
Bit	*Name*	*Function*	*Additional information*
15-14	Reserved		
13-8	SAR Size	This is a 6-bit counter initialized with the number of data transfers to be performed. It should be set to *n*-1, where *n* is the number of data transfers to be performed (including HEC if enabled). This information is loaded into the Transfer Counter (TC) register in the DMU.	
7-6	Reserved		
5-0	SAR Offset	The 6-bit value in the SAR Offset field should equal the offset of the first byte of SAR SDU data. This information is loaded into the Task Buffer Offset (TBO) register in the DMU. Since the Task Buffers are 64 bytes, only bits [5:0] are required for addressing, and bits [7:6] are always zero.	

TT AND TVR REGISTERS

These registers, part of the gather DMU, control threshold testing of gathered data.

Location: 0xFFF30(Gather DMU)

Reset Value: 0000 0000 0000 0000



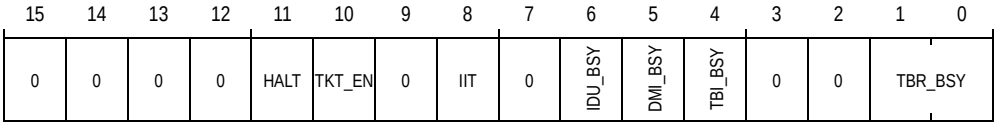
Bit	*Name*	*Function*	*Additional information*
15-11	Reserved		
10-8	TT register	This register controls the type of thresholding done on the gathered data. The thresholding is an unsigned comparison of gathered data to the value in the Threshold Value Register. 000 gath_data [7:0] ≤ TVR [7:0] 001 gath_data [7:0] > TVR [7:0] 010 gath_data [7:0] = TVR [7:0] 011 gath_data [7:0] ≠ TVR [7:0] 100 gath_data [6:0] ≤ TVR [6:0] 101 gath_data [6:0] > TVR [6:0] 110 gath_data [6:0] = TVR [6:0] 111 gath_data [6:0] ≠ TVR [6:0]	
7-0	TVR register	This register contains an 8-bit value used for threshold testing.	

STATUS REGISTER

This register contains the status of the DMU.

Locations: 0xFFF0C (Scatter DMU)
0xFFF2C (Gather DMU)

Reset Value: 0000 0000 0000 0000



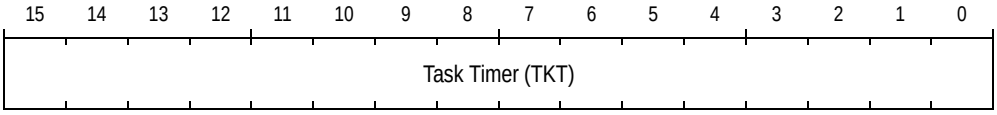
Bit	***Name***	***Function***	***Additional information***
15-12	0	Unused	
11	HALT	DMU Halt Flag	
10	TKT_EN	Task Time Enabled	
9	0	Unused	
8	IIT	Illegal Instruction Trap	
7	0	Unused	
6	IDU_BSY	IDU Busy Flag	The IDU_BSY, DMI_BSY, and TBI_BSY bits are for debug purposes only.
5	DMI_BUSY	DMI Busy Flag	
4	TBI_BSY	TBI Busy Flag	
3	0	Unused	
2	0	Unused	
1-0	TBR_BSY	Task Buffer Busy Flags	Used to detect scatter/gather completion

TASK TIMER REGISTER

This register, part of the DMUs, is used for performance monitoring.

Locations: 0xFFF0A (Scatter DMU)
0xFFF2A (Gather DMU)

Reset Value: 0000 0000 0000 0000



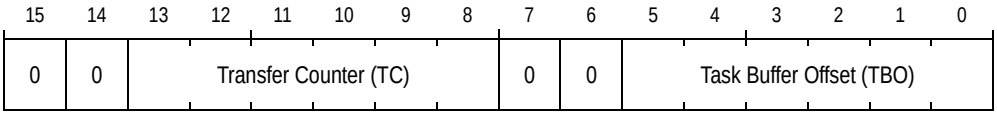
Bit	*Name*	*Function*	*Additional information*
15-0	Task Timer (TKT)	This is a 16-bit counter, intended for performance monitoring, that counts clock cycles while either TBR_BSY flag is set.	

TRANSFER COUNTER AND TASK BUFFER OFFSET REGISTERS

These registers are part of the DMUs. At the conclusion of the DMA that loads the Task Buffer RAM, the SAR Size information is automatically loaded into the Transfer Counter register, and the SAR Offset information is automatically loaded into the Task Buffer Offset register.

Locations: 0xFFF04 (Scatter DMU)
0xFFF24 (Gather DMU)

Reset Value: 0000 0000 0000 0000



Bit	*Name*	*Function*	*Additional information*
14-8	Transfer Counter (TC)	This is a 6-bit counter loaded from the SAR Size register in the Task Buffer RAM. It is decremented after each data transfer is completed.	
5-0	Task Buffer Offset (TBO)	This value is used as a counter that points to the next location to be read or written in the Task Buffer. Since the Task Buffers are 64 bytes, only bits [5:0] are required for addressing, and bits [7:6] are always zero.	

Section 3 Physical Specification

The chapters in this section provide the address maps, timing, pinning, signals, and package mechanical and thermal information for the MXT3020.

CHAPTER 7 *Interfacing Guidelines*

This chapter provides interfacing guidelines for connecting the MXT3020 to:

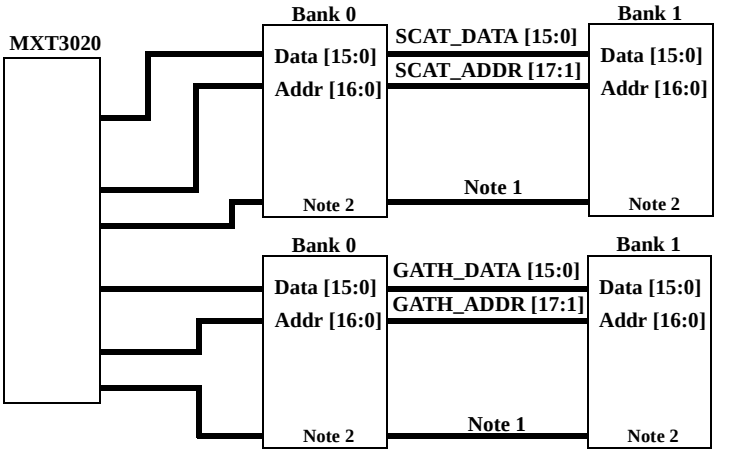
- Scatter/Gather Memory
- The MXT3010
- Other MXT3020's

This chapter also provides information on clock signal distribution.

INTERFACING TO SCATTER/GATHER MEMORY

Both the Scatter/Gather memory interfaces have a 16-bit wide data path and each support up to 512 Kbytes. The memories are *pipeline* synchronous SRAMs with a 20 ns cycle time. An example is the Micron MT58LC128K18D9LG. Figure 36 shows the interconnection of the MXT3020 to 512 Kbytes of Scatter and 512K bytes of Gather memory.

FIGURE 36. Schematic of MXT3020C to Scatter/Gather Memories



Notes: 1. Table 16 on page 125 shows the control lead connections for each SRAM.

2. The MXT3020 receives its clock from FN_CLK1. See “Interfacing the MXT3020 to the MXT3010” on page 126.

TABLE 16. Scatter/Gather Memory control connections

Signal	***Scatter Mem Bank 0***	***Scatter Mem Bank 1***	***Gather Mem Bank 0***	***Gather Mem Bank 1***
3020 SWE_1	WEH#	WEH#	------	------
3020 SWE_0	WEL#	WEL#	------	------
3020 SOE_0	OE#	------	------	------
3020 SOE_1	------	OE#	------	------
SCAT_CE_[a]	CE#	CE#	------	------
3020 GWE_1	------	------	WEH#	WEH#
3020 GWE_0	------	------	WEL#	WEL#
3020 GOE_0	------	------	OE#	------
3020 GOE_1	------	------	------	OE#
GATH_CE_[b]	------	------	CE#	CE#
3020 SCAT_ADDR [18]	CE2#	CE2	------	------
3020 SCAT_ADDR [17:1]	ADDR [16:0]	ADDR [16:0]	------	------
VDD	CE2	------	------	------
GND	------	CE2#	------	------
3020 GATH_ADDR [18]	------	------	CE2#	CE2
3020 GATH_ADDR [17:1]	------	------	ADDR [16:0]	ADDR [16:0]
VDD	------	------	CE2	------
GND	------	------	------	CE2#
FN_CLK2	------	------	CLK	------
FN_CLK3	------	------	------	CLK
FN_CLK4	CLK	------	------	------
FN_CLK5	------	CLK	------	------
VDD	GW#, ADV#, ADSP# of all Scatter/Gather SRAMs			
GND	ADS0#, BWE#, MODE, ZZ of all Scatter/Gather SRAMs			

a. At the Scatter Memory, SCAT_CE_ is a connection to GND through a resistor in the range of 120 ohms to 1K ohms.

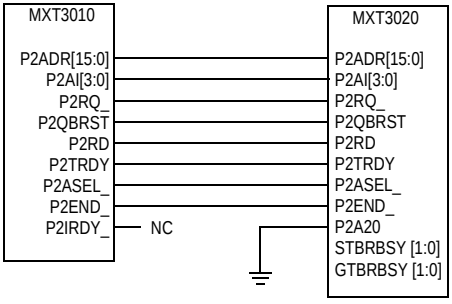
b. At the Gather Memory, GATH_CE_ is a connection to GND through a resistor in the range of 120 ohms to 1K ohms.

INTERFACING THE MXT3020 TO THE MXT3010

The MXT3020 connects to the MXT3010 Port2 interface, which supports both burst and non-burst transfer modes. This interface is used to configure and to transfer SAR SDU data between the MXT3020 and the MXT3010. The MXT3010 can address 2 Mbytes in burst mode and 512 Kbytes in non-burst mode. The P2QBRST signal from the MXT3010 identifies burst (1) or non-burst (0) cycles. The MXT3020 is a burst-mode only device and requires a 1 Mbyte address space.

A single MXT3020 can interface directly to the MXT3010. The schematic below illustrates the connections between the MXT3020 and Port2 of the MXT3010. Only the Port2 signals are shown for clarity. Note that P2A20 is strapped low and therefore the MXT3020 will be mapped into the lower 1 Mbyte region. The MXT3020 should always be mapped into this space in order to function properly with the framework of the CircuitMaker application.

FIGURE 37. MXT3020 to MXT3010 interconnection schematic



Notes: 1. The MXT3020 does not use the P2IRDY_signal.

2.Wiring details for the STBRBSY [1:0] and GTBRBSY [1:0] signals are provided in “Scatter/ Gather Task Buffer Busy flags” on page 134.

Port2 Burst and Non-Burst Operation

Burst mode

Burst mode transfers consist of an address phase followed by one or more data cycles. The MXT3010 Port2 DMA controller uses this mode to transfer data and control information to the MXT3020 (see “Port2 Interface timing” on page 144) The MXT3010 initiates all transfers, as the MXT3020 is always a slave device.

Operating in burst mode, the Port2 Interface can access one million 16-bit halfwords. It does this by using P2AD[15:0] for logical address bits [19:4] and P2AI[3:0] for logical address bits [3:0].

The MXT3020 adds Wait states as needed, using P2TRDY_.

Non-burst mode

Non-burst mode consists of an address phase followed by a single data phase (read or write). The MXT3010 Port2 DMA controller uses this mode to select the appropriate MXT3020 in multiple-MXT3020 configurations (see “Multiple MXT3020 implementation” on page 130). The MXT3010 also uses this mode, in conjunction with the MXT3020, to access non-burst devices (see “MXT3020 Assistance to Non-burst Devices” on page 158) and to read the Scatter/Gather Task Buffer busy flags in the MXT3020(s) (see “Scatter/Gather Task Buffer Busy flags” on page 134).

Operating in non-burst mode, the Port2 Interface can access 512k bytes. It does this by using P2AI[3:2] for logical address bits [17:15] and P2AD[15:0] for logical address bits [15:0].

When non-burst mode is selected, a programmable number of wait states (up to 7 wait states) can be specified within the DMA2R/W instruction.

The DMA2 instructions

A simplified version of the basic MXT3010 DMA2 instruction format is shown below:

DMA2W - Direct Memory operation - Port2 Write

31	30	29	28	27	26	25 24 23 22 21 20 19	18 17 16	15 14 13 12 11 10	9 8 7 6 5	4 3 2 1 0
0	1	1	1	1		BC	rla	rsa	Control	rsb

DMA2R - Direct Memory operation - Port2 Read

31	30	29	28	27	26	25 24 23 22 21 20 19	18 17 16	15 14 13 12 11 10	9 8 7 6 5	4 3 2 1 0
0	1	1	1	0		BC	rla	rsa	Control	rsb

The rsa, rsb and rla fields within the DMA2 instruction specify the registers that supply the source and destination address for the DMA transfers. Burst or non-burst transfer mode is selected by bit 7 of the rsa (source register). Burst transfers are initiated when rsa [7] = 1, and non-burst when rsa [7]=0. Dependent on the transfer type (burst or non-burst mode), the mapping of the bits in rsa and rsb take on different meanings.

Figure 38 and Table 17 illustrate the correspondence between rsa/rsb register values, the Port2 bus signals, and a logical halfword address for Port2 burst DMA transfers.

FIGURE 38 Diagram of Port2 burst DMA instruction bits

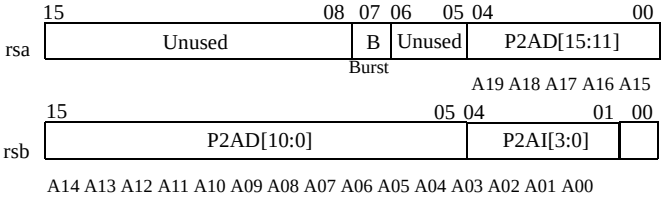
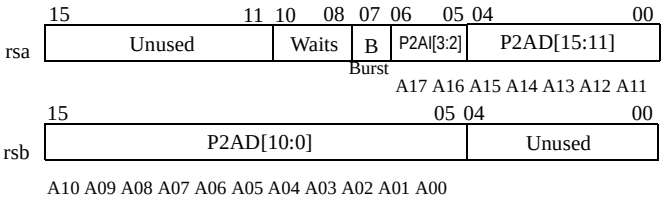


TABLE 17. Port2 burst DMA instruction bit mapping

Reg	*Bits*	*Function*	*Port2 Bus*	*Logical Halfword Bit*
rsa	15:08	Not used	-	-
	07	Burst bit = 1	(selects burst mode)	-
	06:05	Not used	-	-
	04:00	Address	P2AD[15:11]	19:15
rsb	15:05	Address	P2AD[10:0]	14:04
	04:01	Address	P2AI[3:0]	3:0
	00	Discarded	-	-

Figure 39 and Table 18 illustrate the correspondence between rsa/rsb register values, the Port2 bus signals, and a logical halfword address for Port2 non-burst DMA transfers.

FIGURE 39 Diagram of Port2 non-burst DMA instruction bits



.

TABLE 18. Port2 non-burst DMA instruction bit mapping

Reg	*Bits*	*Function*	*Port2 Bus*	*Logical Halfword Bit*
rsa	15:11	Not used	-	-
	10:08	#waits [2:0]	(selects number of wait states)	-
	07	Burst bit = 0	(selects mode)	-
	06:05	Address	P2AI[3:2]	17:16
	04:00	Address	P2AD[15:11]	15:11
rsb	15:05	Address	P2AD[10:0]	10:0
	04:00	Discarded	-	-

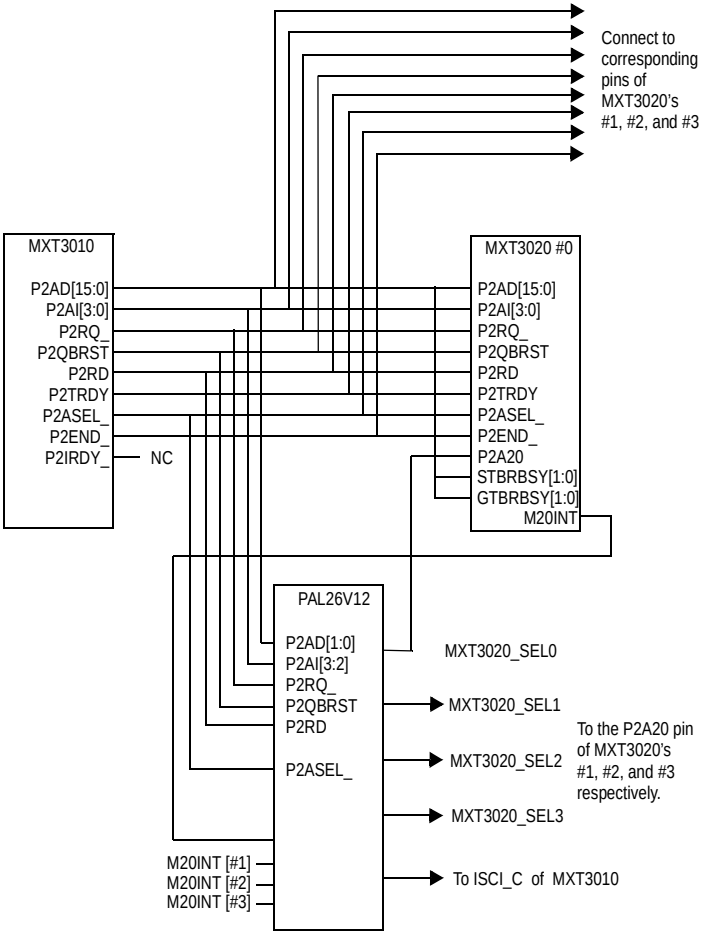
MULTIPLE MXT3020 IMPLEMENTATION

Port2 burst address space can accommodate two MXT3020s; however the CircuitMaker application expects each MXT3020 to be mapped into the lower 1 Mbyte region. Therefore, when interfacing the MXT3010 to multiple MXT3020s, additional logic is necessary for device selection. A registered 2-to-4 decoder, implemented with a 26V12 PAL (10 ns speed), is used to select an individual MXT3020 with the use of P2A20. To select a MXT3020, the registered decoder will be *read* by the MXT3010 utilizing a Port2 non-burst read. CircuitMaker will manipulate the decoder to select a specific MXT3020.

The non-burst read from the registered decoder will result in asserting (low true) the selected MXT3020's P2A20 input. MXT3020 devices that are not selected will be mapped in the upper 1 Mbyte region, since their P2A20 input will be negated. Note that one MXT3020 will be selected to provide the P2ASEL_ for the non-burst cycles. P2ASEL_ is required for this selection mechanism (see "MXT3020 Assistance to Non-burst Devices" on page 158)

Port 2 address bits P2AI[3:2] are used to decode the non-burst address space into four regions. The decoder is selected by a non-burst read cycle in the fourth region by decoding P2AI[3:2] equal to 11. The least significant address bits (P2AD[1:0]) are captured by the decoder and determine which MXT3020 is selected. The data returned by the DMA read should be ignored. Only after an MXT3020 is selected can Port2 burst transfers commence to configure or transfer data. Figure 40 on page 131 shows a schematic of this implementation, and the source code for the PAL is included in Appendix B of this document.

FIGURE 40. Schematic of circuit for interfacing quad MXT3020's



Notes:1.Each device has a clock input and each is driven by a separate FN_CLK output from the clock distribution circuit shown in Figure 43 on page 138.

2. Each device shown has a reset input driven by the RST_ signal. RST_ is synchronous with FN_CLK. All Port2 devices should have RST_ released on the same clock cycle.

3. Wiring details for the STBRBSY [1:0] and GTBRBSY [1:0] signals are provided in "Scatter/ Gather Task Buffer Busy flags" on page 134.

DEVICE SELECTION CODE EXAMPLE

The code fragment below selects a particular MXT3020. The DMA2R instruction transfers a halfword from the source address specified in rsa/rsb to a Cell Buffer RAM location specified in rla. In this example, the source address indentifies the desired MXT3020 to the registered decoder. The data read is ignored by the software, but the Cell Buffer RAM location specified by rla will be overwritten.

Due to an error in the MXT3020B, the non-burst read to the decoder must be followed by a burst operation (please refer to PR 380 for detailed information). The MXT3020 select lines will get asserted on the next Port 2 burst operation. The previously selected device will remain selected until this burst operation is completed. A burst write operation is preferred, since its cycle time is less than a burst read operation. In the code below, suitable for use with the MXT3020B and C, a burst write to the Circuit Interface Status register is used for this purpose. The Circuit Interface Status register is a read-only register and writing to it has no effect.

```
;;; Routine name: $FI_SelectMXT3020 - Select an MXT3020
;;;
;;; On entry: a0     MXT3020 index (0-3)
;;; On exit:  a0-a2  Trashed
;;;
;;; This function selects an MXT3020 by issuing a read
;;; followed by a write to one of the following addresses:
;;;
;;; MXT3020 #0 - 0x600000
;;; MXT3020 #1 - 0x600020
;;; MXT3020 #2 - 0x600040
;;; MXT3020 #3 - 0x600060
;  Set up for burst write transfer to Bit Bucket
;  For FI_BitBkt, any free CBR space will do.
;  We use 0x37E.
;  Note: GPB = r51, a0=r0, a1=r1 and a2=r2
#define         #$RegSpaceHi                    0x008F
#define         #$CIStatusReg                   0xFFCA
```

```
$FI_SelectMXT3020
limd a1 0x60         ; 3020 device select at 0x6000x0
mv GPB a2            ; save RLA register
limd GPB #FI_BitBkt  ; destination for dummy read
sftli a0 5 a0        ; adjust the 3020 index to become the
                     ; least significant 16 bits of
                     ; the appropriate device selection
                     ; address

;non-burst read for device selection

dma2r a1/a0 GPB BC/2 ; Read device selection register
mv a2 GPB            ; Restore RLA register

;dummy burst write

limd a1 #$RegSpaceHi ; MSBs of CI status register
limd a0 #$CIStatusReg; LSBs of CI status register
dma2w a1/a0 GPB BC/2 ; Burst write to CI status reg (r/o)

br
nop
```

SCATTER/GATHER TASK BUFFER BUSY FLAGS

The MXT3010 can read the status of the Scatter Task Buffer Busy flags and Gather Task Buffer Busy flags for four MXT3020C devices in a single non-burst DMA read directed to location 0x400000. This read also clears the flags.

The Scatter/Gather Task Buffer Busy flags also appear on pins 185, 186, 187, and 188. They are tri-stated unless they are being read with a non-burst read. The flags are a one (1) or logic high if the associated Task Buffer is busy and a zero (0) or logic low if the associated Task Buffer is not busy.

The Scatter and Gather Task Buffer Busy flags should be wired directly to the P2_AD lines as shown in Table 19. When wiring a single MXT3020C, use the connections listed in the MXT3020 #0 column.

TABLE 19.Scatter and Gather Task Buffer Busy flag wiring

Signal	***MXT3020 #3***	***MXT3020 #2***	***MXT3020 #1***	***MXT3020 #0***
STBRBSY1	P2AD [15]	P2AD [11]	P2AD [7]	P2AD [3]
STBRBSY0	P2AD [14]	P2AD [10]	P2AD [6]	P2AD [2]
GTBRBSY1	P2AD [13]	P2AD [9]	P2AD [5]	P2AD [1]
GTBRBSY0	P2AD [12]	P2AD [8]	P2AD [4]	P2AD [0]

Quad MXT3020 layout considerations

The Port2 interface bus is a high speed multiplexed address and data bus that must be handled properly to ensure proper system operation. Figure 41 and Figure 42 provide example designs upon which SPICE modeling has been performed.

FIGURE 41Quad MXT3020 interconnect topology #1

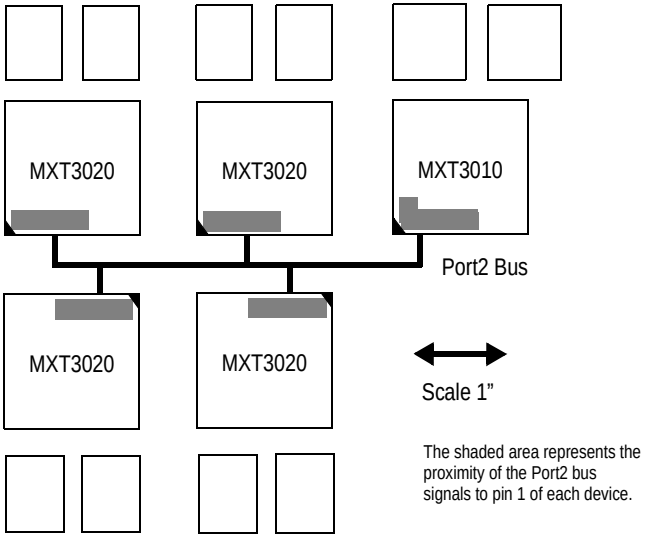
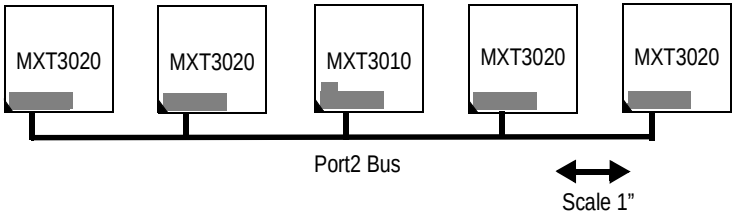


FIGURE 42Quad MXT3020 interconnect topology #2



Timing analysis of quad MXT3020 on Port 2

Timing analyses of the interconnection topologies shown in Figure 41 and Figure 42 are given below. The two topologies were simulated using SPICE. Either of the two interconnection topologies will work; however, topology #1 has better margins.

TABLE 20. Synopsis of MXT3020 and MXT3010 timing requirements

Port 2 interface timing: 50 MHz / 20 nsec cycle	
3020 Setup Required:	> 8.5 nsec (control signals worst case)
3020 Setup Required:	> 4.5 nsec (address / data)
3020 Hold Required:	> 0 nsec
3020 Clock to out:	3 nsec min, 9.5 nsec max (control sigs worst case)
3020 Clock to out:	3 nsec min, 8 nsec max (data out)
3010 Setup Required:	> 8 nsec (control signals worst case)
3010 Setup Required:	> 4 nsec (address / data)
3010 Hold Required:	> 0 nsec
3010 Clock to out:	2.2 nsec min, 9.5 nSec max (addr / data)
3010 Clock to out:	2 nsec min, 7.9 nSec max (control)
Tolerable interconnect delay:	
Control delay:	< 3.6 nsec (3010 -> 3020)
Control delay:	< 2.5 nsec (3020 --> 3010)
Address Delay:	< 4 nsec (either direction)

TABLE 21. Simulated interconnect delay

Clustered topology (Topology#1 - Figure 41)	
Worst case:	MXT3010 (U3) driving furthest MXT3020 (U1)
Interconnect delay:	0.85 nsec (3.6 allowed)
Daisy-chain topology (Topology#2 - Figure 42)	
Worst case:	Outer MXT3020 (U1 or U5) driving MXT3010 (U3)
Interconnect delay:	2.1 nsec (2.5 allowed)

Note: The daisy-chain interconnect delay for the MXT3010 driving to an MXT3020 is worst case for an “inner” MXT3020. That delay is 0.96 nsec (3.6 allowed)

PCB Design Concerns for quad MXT3020

The Port2 interface of the MXT3010 and MXT3020 is a high-speed bidirectional multiplexed address/data bus. To ensure optimum circuit performance, the Port 2 bus should be routed on a single pair of layers. This layer pair should be between the 3.3V plane and the ground plane. This ensures that the signal return currents will be able to follow the transmission lines. It is recommended that the Port2 bus not be routed on a layer which is adjacent to the 5V plane, as there may be no 5V decoupling in the area of this circuit.

CLOCK TREE DISTRIBUTION

The design of the clock tree distribution circuit has two important goals:

1. Reduction of clock skew

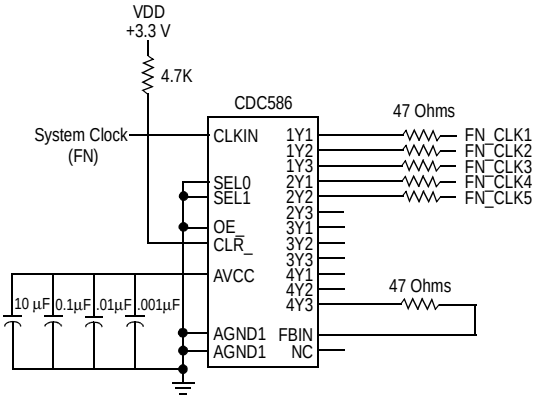
   To reduce clock skew, it is recommended that the designer implement the circuit shown in Figure 43. This circuit uses the Texas Instruments CDC586, which is a clock driver with an integrated PLL. As shown in Table 16 on page 125 and in Figure 43, the CDC586 uses a separate driver for each synchronous SRAM and for the MXT3020.

   To further control clock skew, it is important to make the impedances of the traces equal. This can be done by ensuring that the trace lengths of the clock and feedback signals are equal and have minimum vias. In addition, all of the clock traces should be routed in the same layer.

2. Reduction of clock jitter

   To maintain low jitter, the clock edge rate of the MXT3020 FN input must be kept at 1.5 ns or less. This can be accomplished with the CDC586.

FIGURE 43. Clock Distribution Circuit



Notes: 1.Series termination resistors for the FN_CLK leads should be placed close to the driver pins.

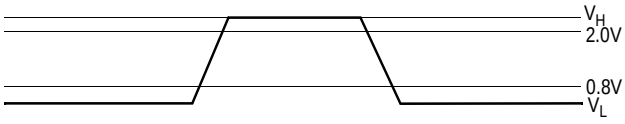
2. There should be no more than a single load on each of the FN_CLK clock leads shown.
3. The feedback etch length from pin 4Y3 to FBIN and the FN_CLK leads should all be the same length.

CHAPTER 8 *Timing*

MXT3020 TIMING - GENERAL INFORMATION

Definition of switching levels

FIGURE 44Switching level voltages



The following switching level information has been used in the generation of the MXT3020 device timing.

- For a low-to-high transition, a signal is considered to no longer be low when it reaches 0.8 V and is considered to be high upon reaching 2.0 V.
- For a high-to-low transition, a signal is considered to no longer be high when it reaches 2.0 V and is considered to be low upon reaching 0.8 V.

Input clock details

FIGURE 45Input clock waveform (pin FN)

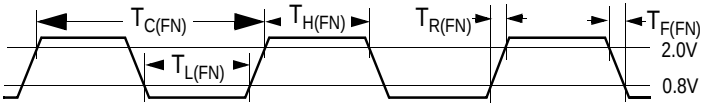


TABLE 22. Input clock timing parameters (in nanoseconds)

	33 MHz		*40 MHz*		*50 MHz*		
	Min	***Max***	***Min***	***Max***	***Min***	***Max***	***Description***
$T_{C(FN)}$	30.28	(1)	24.98	(1)	19.98	(1)	Input clock period
$T_{H(FN)}$	.4xT_C	.6xT_C	.4xT_C	.6xT_C	.4xT_C	.6xT_C	Input clock high duration
$T_{L(FN)}$	.4xT_C	.6xT_C	.4xT_C	.6xT_C	.4xT_C	.6xT_C	Input clock low duration
$T_{R(FN)}$	-	1.4	-	1.4	-	1.2	Input clock rise time (2)
$T_{F(FN)}$	-	1.4	-	1.4	-	1.2	Input clock fall time (2)

1. With the exception of the PLL circuit, the MXT3020 is a fully static design and can operate with $1/T_{C(FN)} = 0$. The device is characterized for operation approaching 0 Hz, but is not tested under this condition.

2. In order to maintain low jitter, pay close attention to the input clock edge rate. One primary component of jitter occurs only during the input clock state transition. To reduce this jitter component, Maker recommends that the FN pin be driven directly from the output of a part designed for clock tree distribution. Maker's reference design uses an FCT3807 device from IDT. Other designs that require a clock driver with an integrated PLL use the CDC586 clock driver from Texas Instruments.

TIMING

This section provides timing tables and diagrams for:

- Circuit Interface
- Port2 Interface
- Scatter/Gather Memory Interface
- SCSA Bus Timing
- MVIP Bus Timing
- MXT3020 Assistance to Non-burst Devices
- MXT3020 Reset Timing

CIRCUIT INTERFACE

This section includes a Circuit Interface timing table and these timing diagrams: Receive Timing in Unidirectional Mode, Transmit Timing in Unidirectional Mode, and Receive/Transmit Timing in Bidirectional Mode.

TABLE 23. Circuit Interface timing

Symbol	***Description***	*50 MHz (Note 1)*	
		min	***max***
Tfps	Receive FSYNC Setup - from valid FSYNC to rising edge of BCLK	4.5	-
Tfph	Receive FSYNC Hold - valid FSYNC after rising edge of BCLK	0	-
Txpd	Transmit Serial Data Delay - from rising clock (Note 2) to serial data valid	3	10.5
Txdz	Transmit Data Disable Delay - from rising clock (Note 2) to serial data tri-state	4	15
Txzd	Transmit Data Enable Delay - from rising clock (Note 2) to serial data valid	5	13
Trds	Receive Serial Data Setup - from valid serial data to falling BCLK	0.5	-
Trdh	Receive Serial Data Hold - valid serial data after falling BCLK	0	-
Ttsd	Tri-state Enable Delay - from rising clock (Note 2) to valid A/BSCE	5	13

Notes:

1. All units are nanoseconds (ns).
2. The clock mentioned in the descriptions of Txpd, Txdz, Txzd, and Ttsd is the worst-case of either ACLK, BCLK, or SRTS generated at the ACLK pin.
3. Each figures shows received or transmitted data for various values of D_DELAY. See “D_DELAY (bits 4 and 3)” on page 14.
4. The numbers shown for 50 MHz operation also apply at lower operating speeds.

FIGURE 46. Receive timing in unidirectional mode

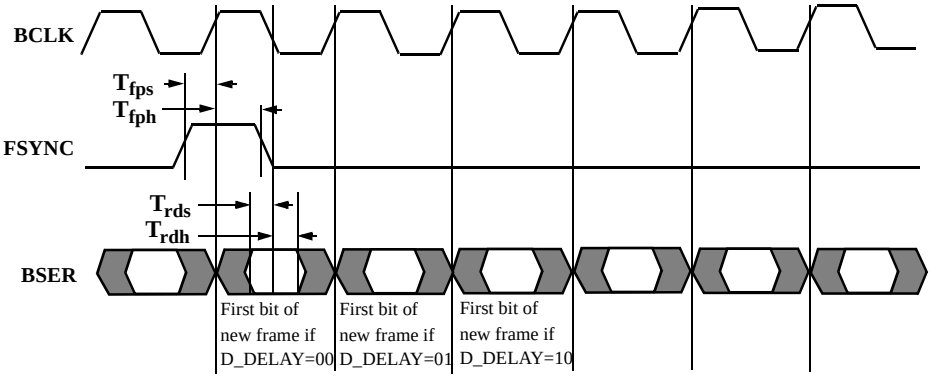
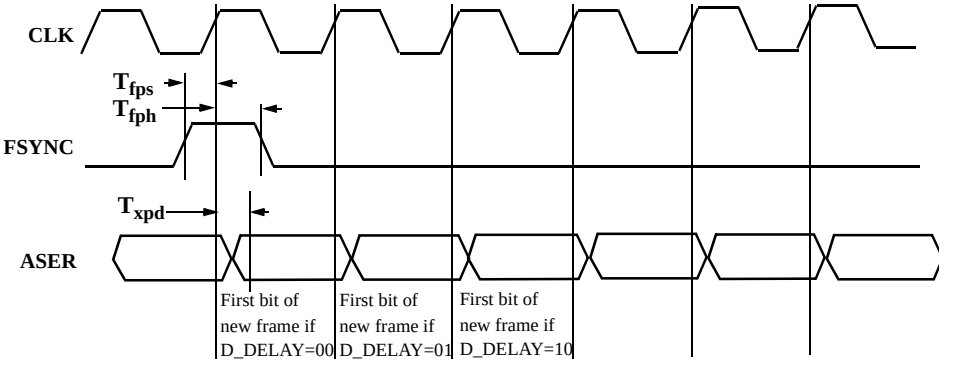
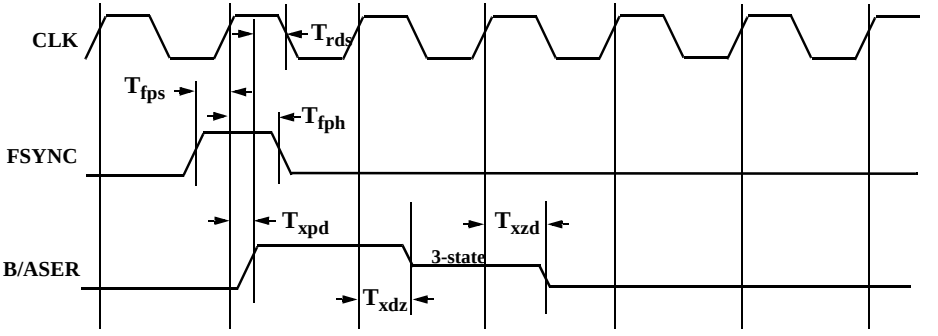


FIGURE 47. Transmit timing in unidirectional mode



Note:The CLK signal shown is the worst-case of ACLK, BCLK, or SRTS generated at the ACLK pin.

FIGURE 48. Receive/Transmit timing in bidirectional mode



Note: The CLK signal shown is the worst-case of ACLK, BCLK, or SRTS generated at the ACLK pin.

PORT2 INTERFACE TIMING

This section includes a Port2 Interface timing table and these timing diagrams: Port2 Burst Write Timing (1 Word), Port2 Burst Write Timing (4 Words), Port2 Burst Read Timing (1 Word), and Port2 Burst Read Timing (2 Words).

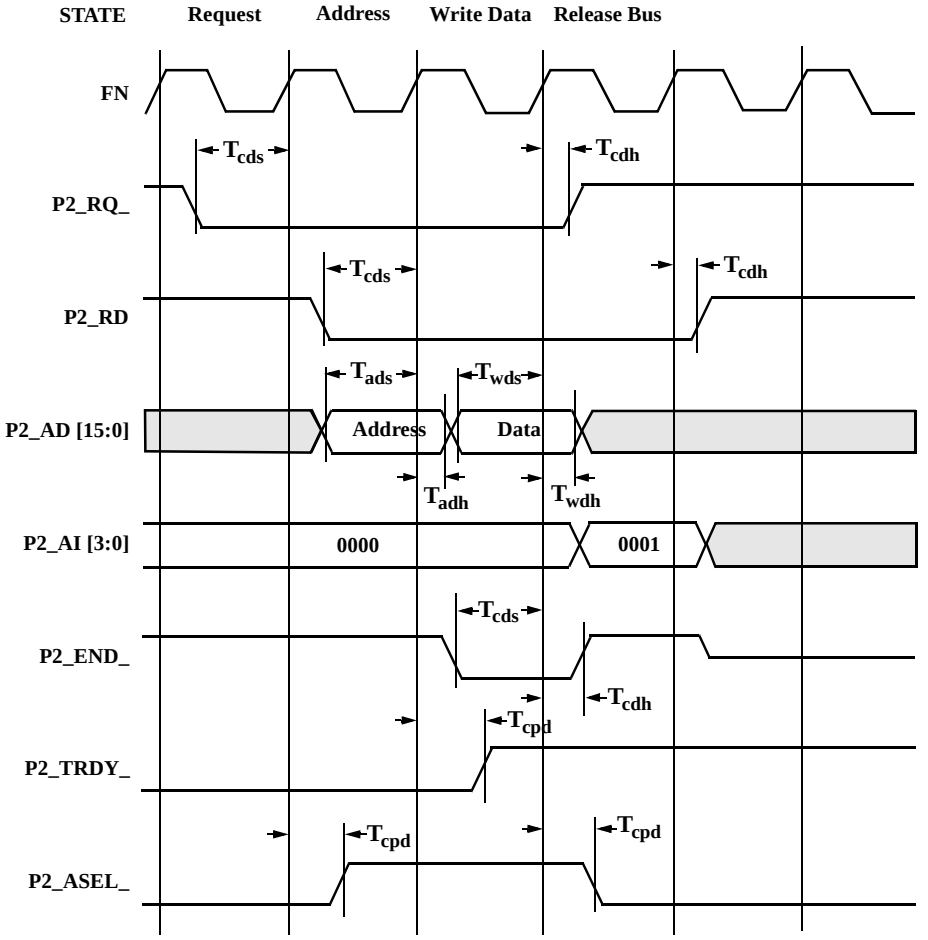
TABLE 24. Port2 Interface timing

Symbol	***Description***	***50 MHz (Note 1)***	
		min	***max***
Tads	Address set-up time - from valid address to rising FN (Note 2)	3.5	
Tadh	Address hold time - valid address after rising FN	0.5	-
Tcds	Control signal set-up time - from valid control signal to rising FN	6	-
Tcdh	Control signal hold time - valid control signal after rising FN	0.5	-
Twds	Write Data set-up time - from valid write data to rising FN	3.5	-
Twdh	Write Data hold time - valid write data after rising FN	0.5	-
The following two signals are driven and timed by the MXT3020:			
Tcpd	Control signal delay - from rising FN to control signal valid	4	10
Trpd	Read Data delay - from rising FN to read data valid	4	11

Notes:

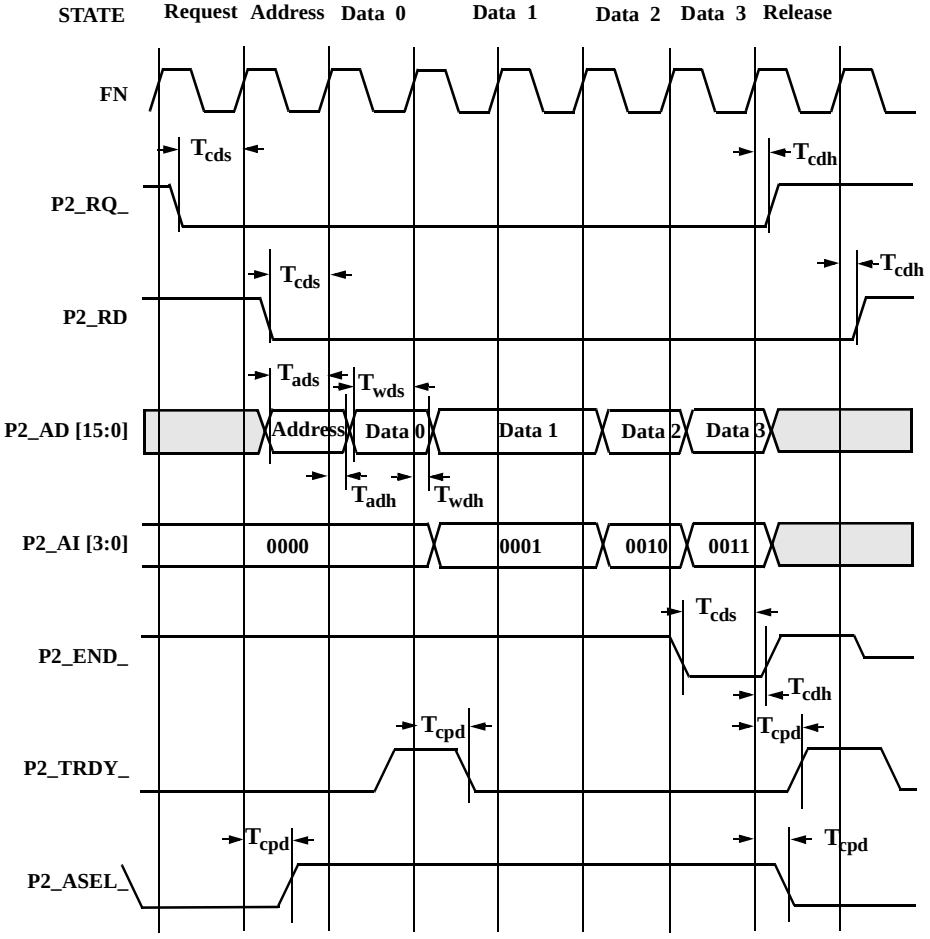
1. All units are nanoseconds (ns).
2. The Address Set-up Time for P2_A20 is 7.5 nanoseconds.
3. The numbers shown for 50 MHz operation also apply at lower operating speeds.

FIGURE 49. Port2 burst write timing (1 halfword)



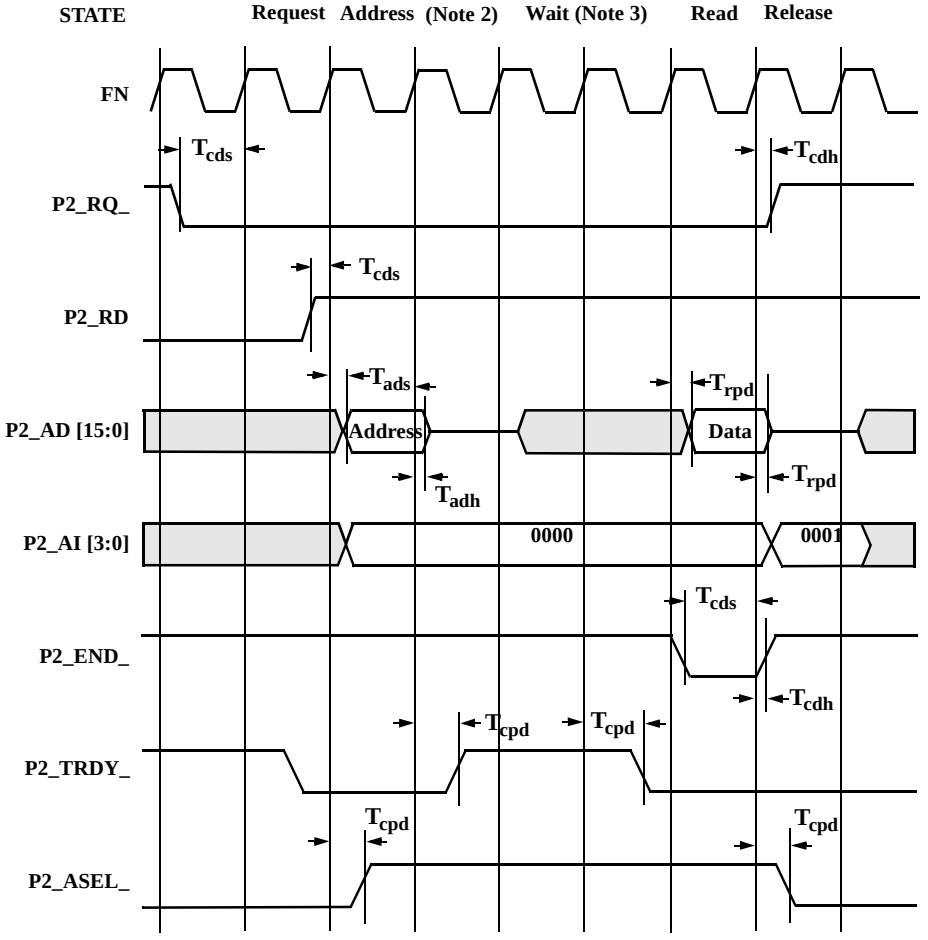
Note: The P2_BRST signal is not shown in the figure, and is asserted throughout the burst write transfer

FIGURE 50. Port2 burst write timing (4 halfwords)



Note: The P2_BRST signal is not shown in the figure, and is asserted throughout the burst write transfer

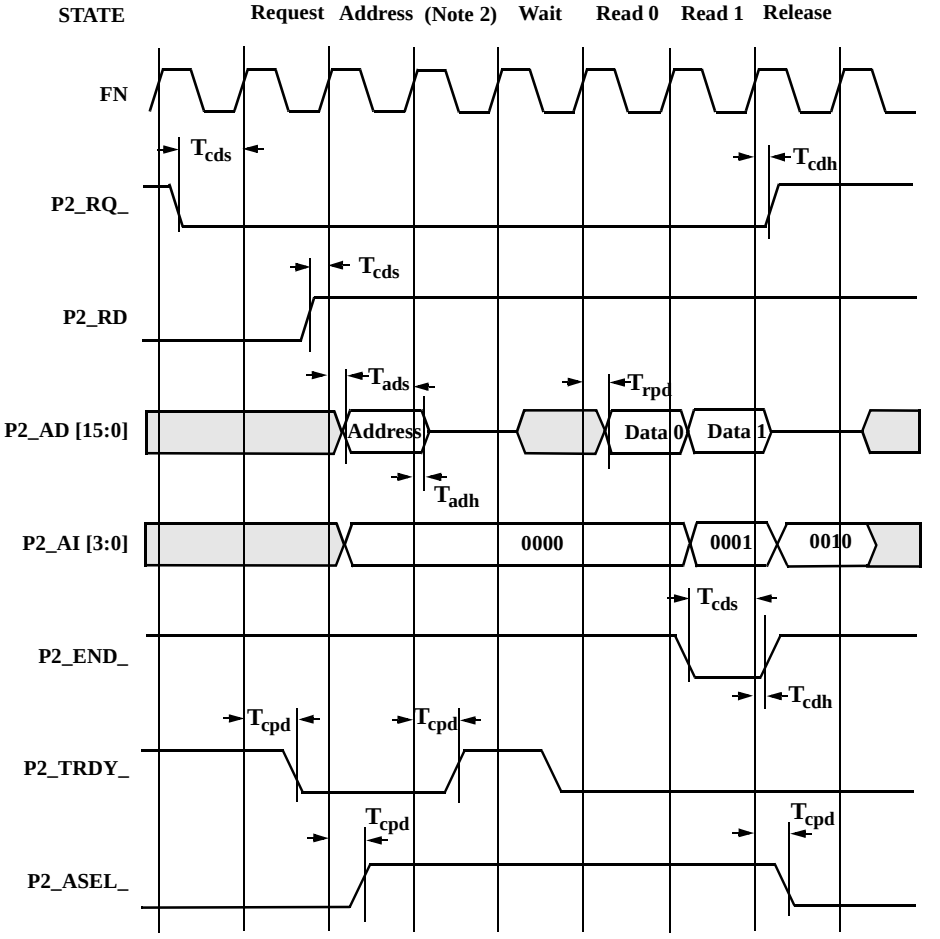
FIGURE 51. Port2 burst read timing (1 halfword)



Notes:

1. The P2_BRST signal is not shown in the figure, and is asserted throughout the burst read transfer.
2. The transfer state immediately after the address state is the bus turnaround state.
3. While the figure above shows two wait states, there can be any number of wait states.

FIGURE 52. Port2 burst read timing (2 halfwords)



Notes:

1. The P2_BRST signal is not shown in the figure, and is asserted throughout the burst read transfer.
2. The transfer state immediately after the address state is the bus turnaround state.
3. While the figure above shows one wait state, there can be any number of wait states.

SCATTER/GATHER MEMORY INTERFACE

This section includes a memory interface timing table and timing diagrams for word writes, burst byte writes, and burst byte reads.

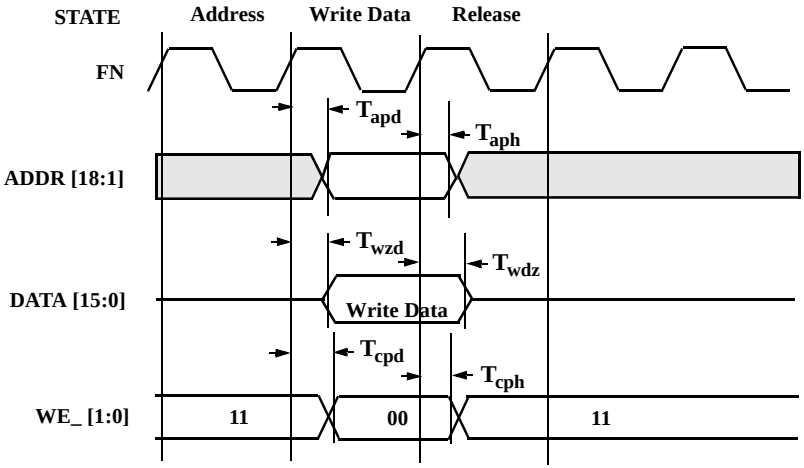
TABLE 25. Scatter/Gather Memory Interface Timing Table

Symbol	*Description*	*50 MHz (Note)*	
		min	*max*
Tcpd	Control Delay - from rising FN to control valid	3.5	10.5
Tcph	Control Hold - valid control after rising FN	3.0	-
Tapd	Address Delay - from rising FN to address valid	3.5	10.5
Taph	Address Hold - valid address after rising FN	3.0	-
Twzd	Transmit Data Enable Delay from rising FN to write data valid (See Figure 53 and Figure 54.)	3.5	10.5
Twpd	Write Data Delay - from rising FN to write data valid (See Figure 54.)	3	9
Twdz	Write Data Disable Delay - from rising FN to write data tri-state (See Figure 53 and Figure 54.)	4	11
Trds	Read Data Setup - from valid read data to rising FN	2.5	-
Trdh	Read Data Hold - valid read data after rising FN	0.5	-

Notes: 1.All units are nanoseconds (ns).

2.The numbers shown for 50 MHz operation also apply at lower operating speeds.

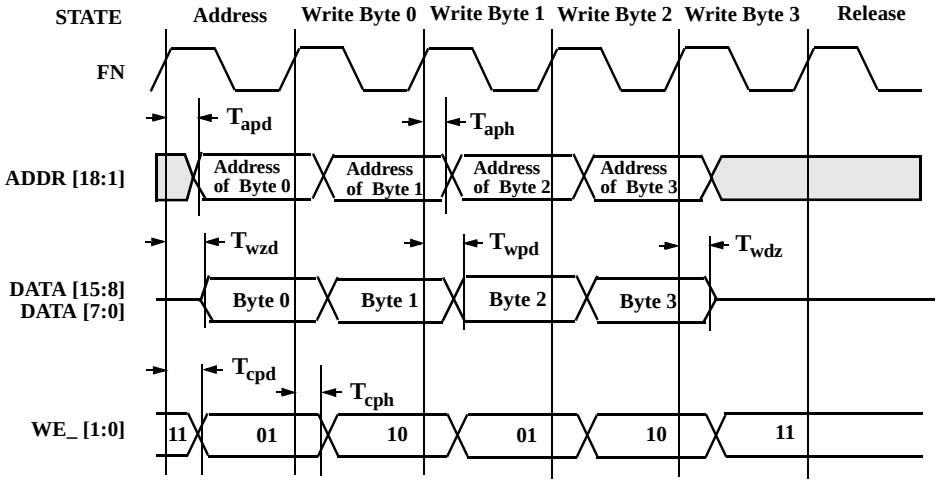
FIGURE 53. Scatter/Gather Memory Halfword Write Timing



Notes:

1. All signals shown (except the chip clock, FN) are prefixed by either GATH_ or SCAT_ depending upon whether they are used with the Gather Memory or Scatter Memory respectively.
2. The Output Enable signals (OE_ [1:0]) are not shown and are in the unasserted state throughout these write operations.

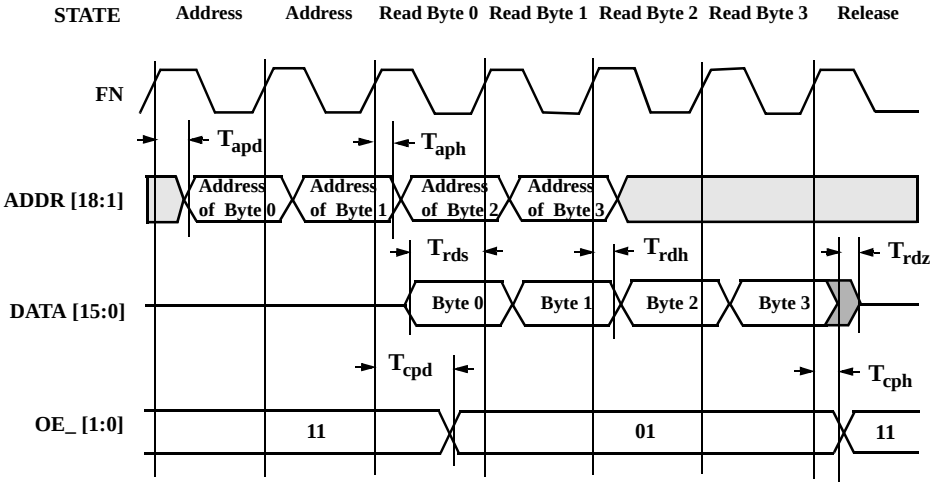
FIGURE 54. Scatter/Gather Memory Burst Write (Byte) Timing



Notes:

1. All signals shown (except the chip clock, FN) are prefixed by either GATH_ or SCAT_ depending upon whether they are used with the Gather Memory or Scatter Memory respectively.
2. The Output Enable signals (OE_ [1:0]) are not shown and are in the unasserted state throughout these write operations.
3. For simplicity, the figure shows a burst write of only four bytes; burst transfers of arbitrary length are possible, retaining the same timing as that shown.
4. Since the Scatter/Gather memories are random access memories, the addresses used do not need to be in sequential order.

FIGURE 55. Scatter/Gather Memory Burst Read (Byte/Halfword) Timing



Notes:

1. All signals shown (except the chip clock, FN) are prefixed by either GATH_ or SCAT_ depending upon whether they are used with the Gather Memory or Scatter Memory respectively.
2. The Write Enable signals (WE_ [1:0]) are not shown and are in the unasserted state throughout these read operations.
3. For simplicity, the figure shows a burst read of only four bytes; burst transfers of arbitrary length are possible, retaining the same timing as that shown.
4. Since the Scatter/Gather memories are random access memories, the addresses used do not need to be in sequential order.

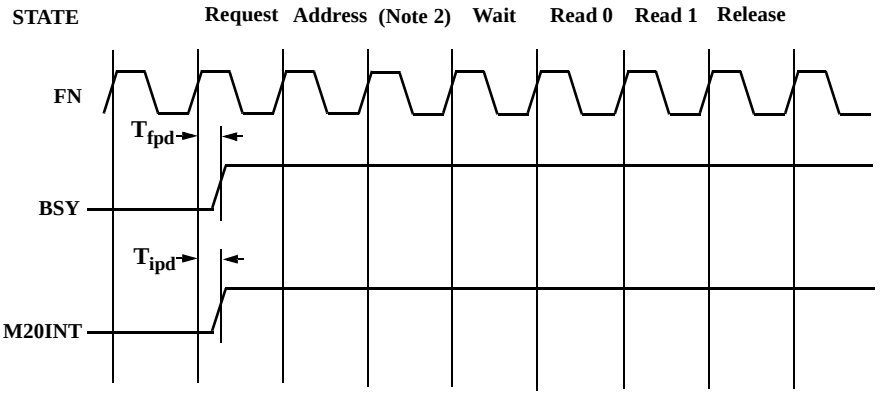
STATUS INTERFACE

This section includes a status timing table and timing diagrams for the GTBRBSY and STBRBSY flags, plus the M20INT indication. For further information about the GTBRBSY and STBRBSY flags, see “Interface Pins” on page 71

TABLE 26. Status interface timing table

Symbol	*Description*	*50 MHz (Note)*	
		min	*max*
Tfpd	BSY Flag (GTBRBSY or STBRBSY) Delay - valid flag after rising FN	4	11.25
Tipd	M20INT Delay - valid indication after rising FN	3.5	9

FIGURE 56. Status interface timing



SCSA BUS TIMING

This section includes the SCSA bus timing for the recommended 2/4 MHz and 8 MHz implementations. Both timing tables and timing diagrams are provided. Figure 57 shows the reference SCSA interface circuit on which the timing is based.

FIGURE 57. Computer telephony interface reference circuit

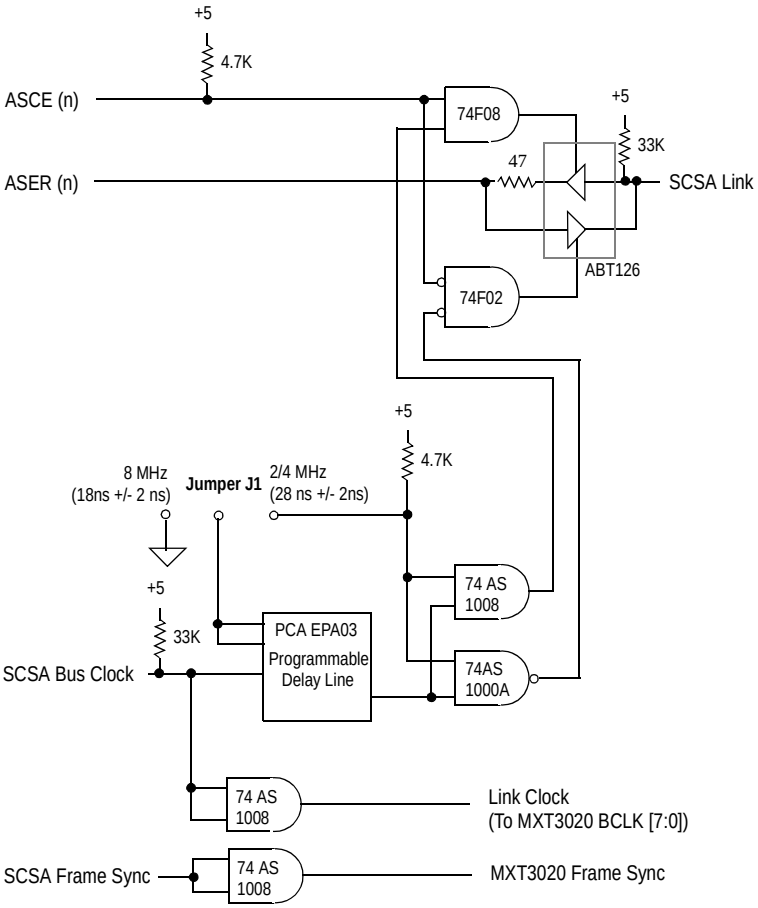


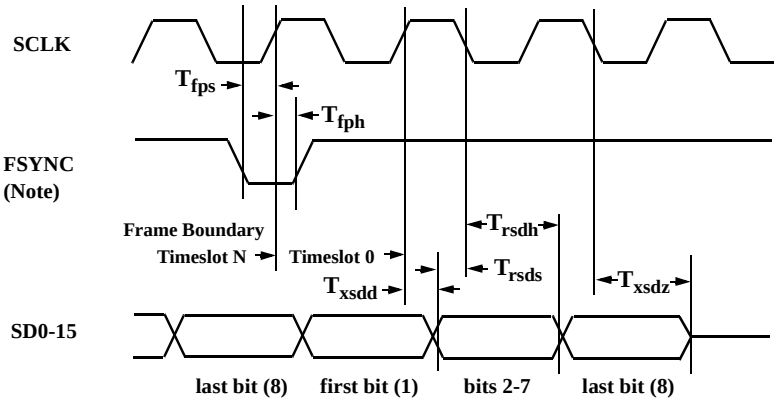
TABLE 27. MXT3020 SCSA Bus Timing (2/4 MHz)

Symbol	*Description*	*50 MHz (Note)*	
		min	*max*
Tfps	Rx Frame Sync Setup Time - from valid FSYNC to rising edge of SCLK	3	-
Tfph	Rx Frame Sync Hold Time - FSYNC valid from rising edge of SCLK	3	-
Txsdd	Tx Output Data Delay - from rising edge of SCLK to valid output or output enabled on the bus	29	47
Trsds	Rx Input Data Setup Time - from valid data to falling edge of SCLK	6	-
Trsdh	Rx Input Data Hold Time - valid data after falling edge of SCLK	8	-
Txsdz	Tx Output Disable Delay - from the SCLK falling edge to output disabled	29	47

Notes: 1.All units are nanoseconds (ns).

2.The numbers shown for 50 MHz operation also apply at lower operating speeds.

FIGURE 58. SCSA Bus Timing



Note: The FSYNC pin on the MXT3020 can be programmed (via the Circuit Interface Configuration Register) to accept either a positive or a negative assertion FSYNC signal.

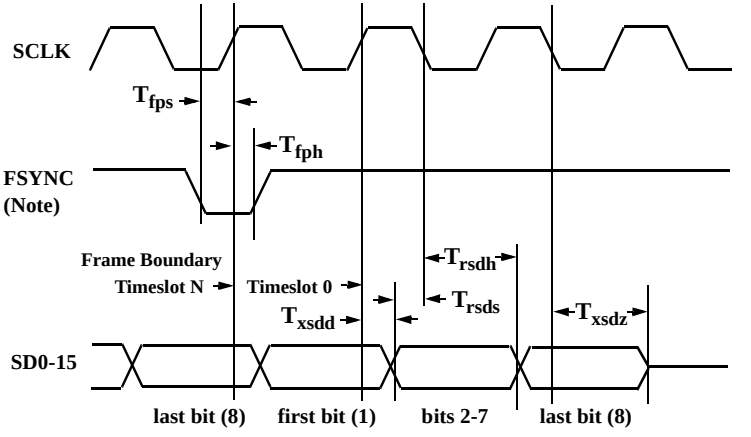
TABLE 28. MXT3020 SCSA Bus Timing (8 MHz)

Symbol	*Description*	*50 MHz (Note)*	
		min	*max*
Tfps	Rx Frame Sync Setup Time - from valid FSYNC to rising edge of SCLK	3	-
Tfph	Rx Frame Sync Hold Time - FSYNC valid from rising edge of SCLK	3	-
Txsdd	Tx Output Data Delay - from rising edge of SCLK to valid output or output enabled on the bus	19	37
Trsds	Rx Input Data Setup Time - from valid data to falling edge of SCLK	6	-
Trsdh	Rx Input Data Hold Time - valid data after falling edge of SCLK	8	-
Txsdz	Tx Output Disable Delay - from the SCLK falling edge to output disabled	19	37

Notes: 1.All units are nanoseconds (ns).

2.The numbers shown for 50 MHz operation also apply at lower operating speeds.

FIGURE 59. SCSA Bus Timing



Note: The FSYNC pin on the MXT3020 can be programmed (via the Circuit Interface Configuration Register) to accept either a positive or negative assertion FSYNC signal.

MVIP BUS TIMING

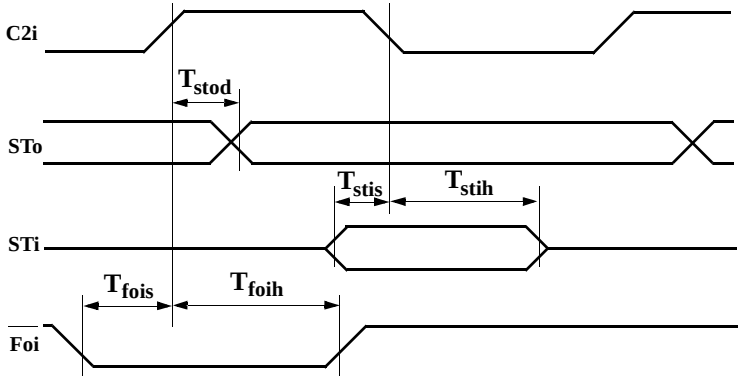
This section includes the MVIP bus timing for the recommended 2 and 4 MHz implementations. Both a timing table and timing diagrams are provided.

TABLE 29. MXT3020 MVIP Bus Timing (2/4 MHz)

Symbol	*Description*	*50 MHz(Note)*	
		min	*max*
Tfois	FOi Setup Time - from valid FSYNC to rising edge of C2i	3	-
Tfoih	FOi Hold Time - FSYNC valid from rising edge of C2i	3	-
Tstod	STo Data Delay - from rising edge of C2i to valid output or output enabled on the bus	29	47
Tstis	STi Data Setup Time - from valid data to falling edge of C2i	6	-
Tstih	Rx Input Data Hold Time - valid data after falling edge of C2i	8	-

Notes: 1.All units are nanoseconds (ns).
2.The numbers shown for 50 MHz operation also apply at lower operating speeds.

FIGURE 60. MVIP BUS Timing (2/4 MHz)



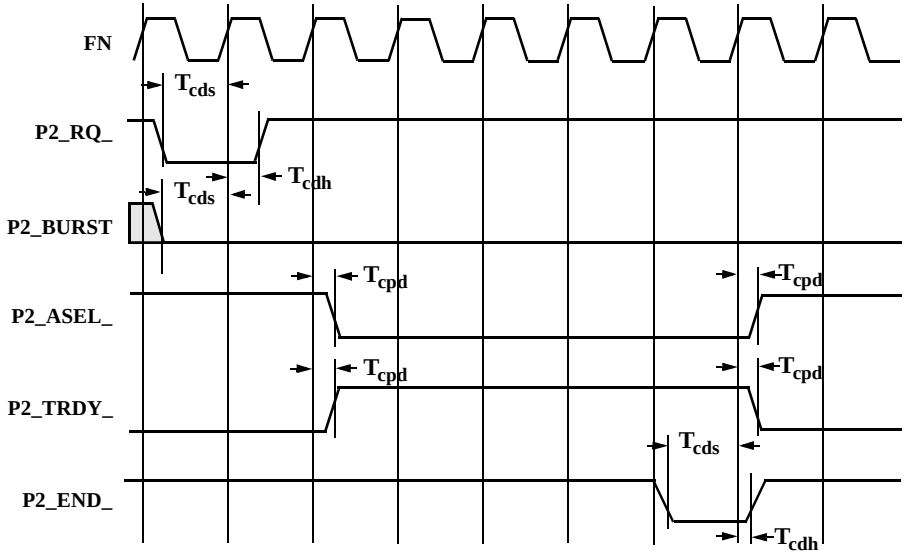
Note: The FSYNC pin on the MXT3020 can be programmed (via the Circuit Interface Configuration Register) to accept either a positive or a negative assertion FSYNC signal.

MXT3020 ASSISTANCE TO NON-BURST DEVICES

The P2 Bus will support both burst transfer and non-burst transfer devices within the same address space. Transfers to/from the two different device types are differentiated by the state of the P2_BURST signal. The signal is asserted for transfers addressed to a burst device and is negated for transfers addressed to a non-burst device.

The MXT3020 contains logic which will recognize the initiation of a non-burst transfer when P2RQ_ is asserted and P2BURST is not asserted. It will respond to this condition by driving P2ASEL_ to the negated state and driving P2TRDY_ to the asserted state until it detects the assertion of P2END_. Since the MXT3020 provides this signalling sequence (Figure 61), non-burst devices can be attached to the P2 Bus with little, if any, additional logic.

FIGURE 61. Timing for MXT3020 Assistance to Non-Burst Devices



Note: For timings, see Table 24, “Port2 Interface timing,” on page 144

MXT3020 RESET TIMING

This section includes a MXT3020 reset timing table and diagram.

TABLE 30. MXT3020 Reset Timing

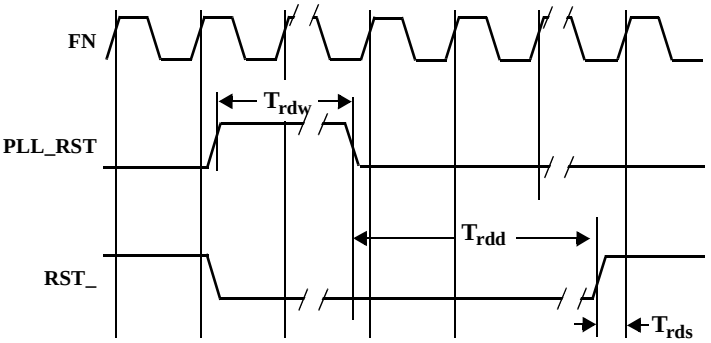
	33 MHz	*40 MHz*	*50 MHz*	
Par				***Description***
Trdw	1 clock cycle	1 clock cycle	1 clock cycle	Minimum number of clock cycles that PLL_Reset must be asserted for use by the MXT3020. Other devices may require a longer assertion.
Trdd	150 clock cycles	150 clock cycles	150 clock cycles	Minimum number of additional clock cycles that RST_ must be held low after PLL_Reset has been deasserted.
Trds	5 ns	5 ns	5 ns	Minimum input setup time to rising CLK for removal of reset signal.

Notes:

1, All units are nanoseconds (ns).

2. While RST_ can be asserted asynchronously, it must be deasserted synchronous to FN.

FIGURE 62. MXT3020 Reset Timing



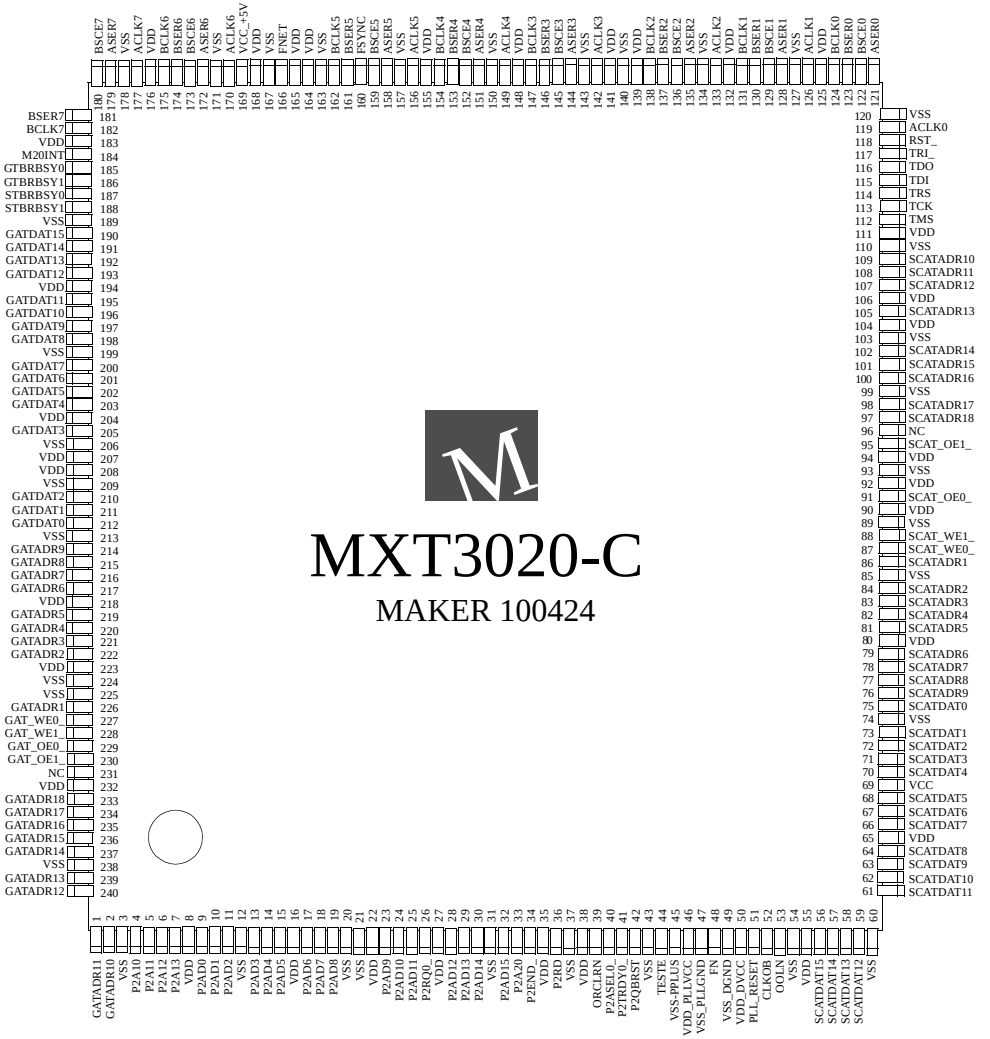
CHAPTER 9 *Pin Information*

This chapter provides information on the MXT3020 pinouts. The information includes pin diagrams, signal descriptions, and pin listings.

MXT3020 PINOUT

Figure 63 provides a diagram of the MXT3020 pinout.

FIGURE 63MXT3020 package/pin diagram



Note: To conserve space in this figure, SCAT_ADRS signals are shown as SCATADR, SCAT_DAT signals as SCATDAT, GATH_ADRS signals as GATADR, GATH_DAT signals as GATDAT, GATH_WE signals as GAT_WE, and GATH_OE signals as GAT_OE. The official signal names used in Table 38 supersede the abbreviations used in this figure.

MXT3020 SIGNAL DESCRIPTIONS

- Port2
- Scatter/Gather Memory Interface
- Circuit Interface
- Miscellaneous signals, such as clock, control and test
- Power and ground pins

TABLE 31. Port 2 Interface Signal Description

Pin #	*Symbol*	*I/O*	*Name*	*Description*
32, 30, 29, 28, 25, 24, 23, 19, 18, 17, 15, 14, 13, 11, 10, 9	P2AD[15:0]	I/O1	Port 2 Address/ Data [15:0]	This is a 16-bit multiplexed address and bi-directional data bus. Data is transferred to/from the MXT3020 over P2AD[15:0]
7, 6, 5, 4	P2AI[3:0]	IN2	Port 2 Address Index Bus	In burst mode P2AI[3:0] represent the four least significant address bits of the 20-bit address. In non-burst mode P2AI[3:2] represent the two most significant address bits of the 18-bit address.
26	P2RQ_	IN2	Port 2 Request	This signal indicates that commands are in the active stage of the MXT3010 Port 2 DMA command queue.
36	P2RD	IN2	Port 2 Read / Write Select	The MXT3010 drives this signal during a DMA transfer to the MXT3020. This signal indicates a write (0) transfer or a read (1) transfer.
34	P2END_	IN2	Port 2 End	The MXT3010 asserts P2END_ (0) to indicate the last cycle of the transfer.
42	P2QBRST	IN2	Port 2 Burst	This signal indicates burst (1) or non-burst (0) transfer mode.
41	P2TRDY_	I/O1	Port 2 Target Ready	P2TRDY_ is used by the MXT3020 to insert wait states. This signal is also used in conjunction with P2ASEL_ to deselect (i.e., tri-state) the MXT3010.
40	P2ASEL_	I/O1	Port 2 Address Select	The MXT3020 drives this signal and selects address or data cycles. This signal is also used in conjunction with P2TRDY_ to deselect (i.e., tri-state) the MXT3010.
33	P2A20	IN2	Port2 Address Bit 20	P2A20 is used to select the 1 Mbyte region in Port2 address space in which the MXT3020 resides. When P2A20 is low, the lower 1 Mbyte region is selected; high selects the upper 1 Mbyte region.

TABLE 32. Scatter/Gather Memory Interface signal description

Pin #	***Symbol***	***I/O***	***Name***	***Description***
97, 98, 100, 101, 102, 105, 107, 108, 109, 76, 77, 78, 79, 81, 82, 83, 84, 86	SCAT_ADDR [18:1]	OUT 1	Scatter Memory Address Lines	Address bus to external synchronous SRAM for Scatter Memory.
56, 57, 58, 59, 61, 62, 63, 64, 66, 67, 68, 70, 71, 72, 73, 75	SCAT_DATA [15:0]	I/O 3	Scatter Memory Data Lines	Data bus to external synchronous SRAM for Scatter Memory.
88,87	SCAT_WE_ [1:0]	OUT 1	Scatter Memory Write Enables	Write enables to external synchronous SRAM for Scatter Memory.
95,91	SCAT_OE_ [1:0]	OUT1	Scatter Memory Output Enables	Output enables to external synchronous SRAM for Scatter Memory.
233, 234, 235, 236, 237, 239, 240, 1, 2, 214, 215, 216, 217, 219, 220, 221, 222, 226	GATH_ADDR [18:1]	OUT 1	Gather Memory Address Lines	Address bus to external synchronous SRAM for Gather Memory.
190, 191, 192, 193, 195, 196, 197, 198, 200, 201, 202, 203, 205, 210, 211, 212	GATH_DATA [15:0]	I/O 3	Gather Memory Data Lines	Data bus to external synchronous SRAM for Gather Memory.
228, 227	GATH_WE_ [1:0]	OUT 1	Gather Memory Write Enables	Write enables to external synchronous SRAM for Gather Memory.
230, 229	GATH_OE_ [1:0]	OUT 1	Gather Memory Output Enables	Output enables to external synchronous SRAM for Gather Memory.
96, 231	No Connect			Formerly SCAT_CE_ and GATH_CE_

Note: SCAT_WE_, SCAT_OE_, GATH_WE_, and GATH_OE_ are active low signals.

TABLE 33. Circuit Interface signal description

Pin #	*Symbol*	*I/O*	*Name*	*Description*
179, 172, 158, 151, 144, 135, 128, 121	ASER [7:0]	I/O 2	Serial Data A [7:0]	In **UNI**-directional mode ASER [7:0] are the Tx Data output signals for the circuit interfaces. In **BI**-directional mode ASER [7:0] are Tx and Rx bi-directional data lines.
181, 174, 161, 153, 146, 137, 130, 123	BSER [7:0]	I/O 2	Serial Data B [7:0]	In **UNI**-directional mode BSER [7:0] are the Rx Data input signals for the circuit interface. In **BI**-directional mode BSER [7:0] are Tx and Rx bi-directional data lines.
177, 170, 156, 149, 142, 133, 126, 119	ACLK [7:0]	I/O 2	Serial Clock A [7:0]	In **UNI**-directional mode ACLK [7:0] are the Tx clocks for ASER[7:0]. ACLKn can be programmed to be an input or an output. In **BI**-directional mode ACLK can be programmed to be the Tx clock for ASER[7:0] and BSER[7:0] or a tri-state enable for ASER [7:0]. See “Link Configuration register” on page 12.
182,175,162, 154,147,138, 131,124	BCLK[7:0]	IN 1	Serial Clock B [7:0]	BCLK [7:0] are the Rx clocks for ASER [7:0] and BSER [7:0] for both UNI and BI modes.
180, 173, 159, 152, 145, 136, 129, 122	BSCE(7:0)	I/O 2	Serial Control B	In **UNI**-directional mode BSCE (7:0) are the frame sync signals for the circuit interfaces. In **BI**-directional mode BSCE (7:0) are tri-state enables for BSER (7:0).
160	FSYNC	IN 1	Frame Sync	Common frame sync input for all links when FS_MODE bit is 0.

TABLE 34. Miscellaneous clock, control, and test signal descriptions

Pin #	*Symbol*	*I/O*	*Name*	*Description*
184	M20INT	OUT 1	MXT3020 Interrupt	This line is asserted whenever the MXT3020 completes a scatter or gather operation. This signal is held asserted until the TBR_BSY flags are read via a P2 Non-burst DMA.
185	GTBRBSY0	OUT 1	Gather TBR_BSY0 (Status register bit 0)	This pin is asserted when Gather Task Buffer 0 is busy.
186	GTBRBSY1	OUT 1	Gather TBR_BSY1 (Status register bit 1)	This pin is asserted when Gather Task Buffer 1 is busy.
187	STBRBSY0	OUT 1	Scatter TBR_BSY0 (Status register bit 0)	This pin is asserted when Scatter Task Buffer 0 is busy.
188	STBRBSY1	OUT 1	Scatter TBR_BSY1 (Status register bit 1)	This pin is asserted when Scatter Task Buffer 1 is busy.
166	FNET	IN 1	Network Clock	Network clock used for SRTS
118	RST_	IN 1	Reset	Reset for MXT3020. Active low.
117	TRI_	IN 1		Tri-state pin for tester. (0) tri-states
116	TDO	OUT 1	Test Data Out	JTAG Test Data Output
115	TDI	IN 1	Test Data In	JTAG Test Data Input
114	TRS	IN 1	Test Reset	JTAG Test Reset Input
113	TCK	IN 1	Test Clock	JTAG Test Clock Input
112	TMS	IN 1	Test Mode Select	JTAG Test Mode Select Input
53	OOLN	OUT 1	Out Of Lock	OOLN is asserted (0) when the MXT3020's internal PLL is out of lock. OOLN is negated (1) when the PLL is locked. This pin is only valid when ORCLRN is asserted low.
52	CLKOB	OUT 1	Clock Output B	CLKOB is the clock output for tester
44	TESTE	IN 3	Test Pin	TESTE is a test pin for the PLL. When (0) test mode is disabled.
39	ORCLRN	IN 3		When this pin is asserted (0), the OOLN pin indicates whether the PLL is locked or out of lock. This pin should normally be pulled high.

TABLE 34. Miscellaneous clock, control, and test signal descriptions

Pin #	*Symbol*	*I/O*	*Name*	*Description*
48	FN	IN 4	System Clock	System Clock (typically the same System Clock used by the MXT3010)
51	PLL_RESET	IN 3	PLL Reset	PLL Reset Input

TABLE 35. Power and Ground pin descriptions

Pin #	*Symbol*	*I/O*	*Name*	*Description*
3, 12, 20, 21, 31, 37, 43, 54, 60, 74, 85, 89, 93, 99, 103, 110, 120, 127, 134, 140, 143, 150, 157, 163, 167, 171, 178, 189, 199, 206, 209, 213, 224, 225, 238	VSS	GND	Ground	These pins provide ground return paths for the various power supply inputs.
8, 16, 22, 27, 35, 38, 55, 65, 80, 90, 92, 94, 104, 106, 111, 125, 132, 139, 141, 148, 155, 164, 165, 168, 176, 183, 194, 204, 207, 208, 223, 218, 232	VDD	VDD	3.3 Volt supply	These pins each require +3.3 VDC (+/- 5%) power supply input. They are used to supply current to the 3.3 volt output buffers and the core logic of the device.
45	VSS_PPLUS	GND	PLL GND	Shield for PLL
46	VDD_PLLVCC	+3.3V	PLL VCC	Analog power for PLL
47	VSS_PLLGND	GND		Analog ground for PLL
49	VSS_DGND	GND		Digital ground for PLL

TABLE 35. Power and Ground pin descriptions

Pin #	***Symbol***	***I/O***	***Name***	***Description***
50	VDD_DVCC	+3.3V		Digital power for PLL
69, 169	VCC	+5V or +3.3V		+5 VDC (+/-10%) or +3.3 VDC (+/-5%) Can be connected to +3.3 VDC if 5 volt tolerant I/O's are not required.

JTAG/PLL MISCELLANEOUS PIN TERMINATIONS

The table below indicates how unused pins on the MXT3020 should be terminated for normal operation.

TABLE 36. Unused pin termination - specific pins

Pin Name	*Pin #*	*Termination*
ORCLRN	39	Pull Up[a]
TESTE	44	Pull Down[b]
TRI_	117	Pull Up
TCK	113	Pull Up
TRS	114	Pull Down

a. A resistor in the range of 4.7K ohms to 10Kohms between the pin and +3.3V

b. A resistor in the range of 120 ohms to 1K ohms between the pin and GND.

If a pin is not utilized within a design, the following guidelines should be followed:

TABLE 37. Unused pin termination - general pins

Pin Type	*Comment*	*Termination*
Output (OUT1)	These are internally terminated	None required
Input (IN 1, IN 2, IN 3, IN 4)	If asserted state is logic low (NAME_)	Pull Up[a]
Input (IN 1, IN 2, IN 3, IN 4)	If asserted state is logic high (NAME)	Pull Down[b]
Input/Output (I/O 1, I/O 2, I/O 3)	If firmware enables as output	See “Output” above
Input/Output (I/O 1, I/O 2, I/O 3)	If firmware enables as input	See “Input” above

a. A resistor in the range of 4.7K ohms to 10Kohms between the pin and +3.3V

b. A resistor in the range of 120 ohms to 1K ohms between the pin and GND

PIN LISTING

Table 38 provides the pin listings for the MXT3020. Table 39 on page 175 provides descriptions of the pin types listed in this table.

.

TABLE 38. Pin Listing

Pin	*Pin Label*	*Pin Type*	*Pin*	*Pin Label*	*Pin Type*
1	GATH_ADRS11	OUT 1	27	VDD	+3.3V
2	GATH_ADRS10	OUT 1	28	P2AD12	I/O 1
3	VSS	GND	29	P2AD13	I/O 1
4	P2AI0	IN 2	30	P2AD14	I/O 1
5	P2AI1	IN 2	31	VSS	GND
6	P2AI2	IN 2	32	P2AD15	I/O 1
7	P2AI3	IN 2	33	P2A20	IN 2
8	VDD	+3.3V	34	P2END_	IN 2
9	P2AD0	I/O 1	35	VDD	+3.3V
10	P2AD1	I/O 1	36	P2RD	IN 2
11	P2AD2	I/O 1	37	VSS	GND
12	VSS	GND	38	VDD	+3.3V
13	P2AD3	I/O 1	39	ORCLRN	IN 3
14	P2AD4	I/O 1	40	P2ASEL_	I/O 1
15	P2AD5	I/O 1	41	P2TRDY_	I/O 1
16	VDD	+3.3V	42	P2QBRST	IN 2
17	P2AD6	I/O 1	43	VSS	GND
18	P2AD7	I/O 1	44	TESTE	IN 3
19	P2AD8	I/O 1	45	VSS_PPLUS	GND
20	VSS	GND	46	VDD_PLLVCC	+3.3V1[a]
21	VSS	GND	47	VSS_PLLGND	GND
22	VDD	+3.3	48	FN	IN 4
23	P2AD9	I/O 1	49	VSS_DGND	GND
24	P2AD10	I/O 1	50	VDD_DVCC	+3.3V
25	P2AD11	I/O 1	51	PLL_RESET	IN 3
26	P2RQ0_	IN 2	52	CLKOB	OUT 1

TABLE 38. Pin Listing

Pin	*Pin Label*	*Pin Type*	*Pin*	*Pin Label*	*Pin Type*
53	OOLN	OUT 1	87	SCAT_WE0_	OUT 1
54	VSS	GND	88	SCAT_WE1_	OUT 1
55	VDD	+3.3V	89	VSS	GND
56	SCAT_DAT15	I/O 3	90	VDD	+3.3V
57	SCAT_DAT14	I/O 3	91	SCAT_OE0_	OUT 1
58	SCAT_DAT13	I/O 3	92	VDD	+3.3V
59	SCAT_DAT12	I/O 3	93	VSS	GND
60	VSS	GND	94	VDD	+3.3V
61	SCAT_DAT11	I/O 3	95	SCAT_OE1_	OUT 1
62	SCAT_DAT10	I/O 3	96	No connect	
63	SCAT_DAT9	I/O 3	97	SCAT_ADRS18	OUT 1
64	SCAT_DAT8	I/O 3	98	SCAT_ADRS17	OUT 1
65	VDD	+3.3V	99	VSS	GND
66	SCAT_DAT7	I/O 3	100	SCAT_ADRS16	OUT 1
67	SCAT_DAT6	I/O 3	101	SCAT_ADRS15	OUT 1
68	SCAT_DAT5	I/O 3	102	SCAT_ADRS14	OUT 1
69	VCC	+5V/3.3V	103	VSS	GND
70	SCAT_DAT4	I/O 3	104	VDD	+3.3V
71	SCAT_DAT3	I/O 3	105	SCAT_ADRS13	OUT 1
72	SCAT_DAT2	I/O 3	106	VDD	+3.3V
73	SCAT_DAT1	I/O 3	107	SCAT_ADRS12	OUT 1
74	VSS	GND	108	SCAT_ADRS11	OUT 1
75	SCAT_DAT0	OUT 1	109	SCAT_ADRS10	OUT 1
76	SCAT_ADRS9	OUT 1	110	VSS	GND
77	SCAT_ADRS8	OUT 1	111	VDD	+3.3V
78	SCAT_ADRS7	OUT 1	112	TMS	IN 1
79	SCAT_ADRS6	OUT 1	113	TCK	IN 1
80	VDD	3.3V	114	TRS	IN 1
81	SCAT_ADRS5	OUT 1	115	TDI	IN 1
82	SCAT_ADRS4	OUT 1	116	TDO	OUT 1
83	SCAT_ADRS3	OUT 1	117	TRI_	IN 1
84	SCAT_ADRS2	OUT 1	118	RST_	IN 1
85	VSS	GND	119	ACLK0	I/O 2
86	SCAT_ADRS1	OUT 1	120	VSS	GND

TABLE 38. Pin Listing

Pin	*Pin Label*	*Pin Type*	*Pin*	*Pin Label*	*Pin Type*
121	ASER0	I/O 2	155	VDD	+3.3V
122	BSCE0	I/O 2	156	ACLK5	I/O 2
123	BSER0	I/O 2	157	VSS	GND
124	BCLK0	IN 1	158	ASER5	I/O 2
125	VDD	+3.3V	159	BSCE5	I/O 2
126	ACLK1	I/O 2	160	FSYNC	IN 1
127	VSS	GND	161	BSER5	I/O 2
128	ASER1	I/O 2	162	BCLK5	IN 1
129	BSCE1	I/O 2	163	VSS	GND
130	BSER1	I/O 2	164	VDD	+3.3V
131	BCLK1	IN 1	165	VDD	+3.3V
132	VDD	+3.3V	166	FNET	IN 1
133	ACLK2	I/O 2	167	VSS	GND
134	VSS	GND	168	VDD	+3.3V
135	ASER2	I/O 2	169	VCC	5V/3.3V
136	BSCE2	I/O 2	170	ACLK6	I/O 2
137	BSER2	I/O 2	171	VSS	GND
138	BCLK2	IN 1	172	ASER6	I/O 2
139	VDD	+3.3V	173	BSCE6	I/O 2
140	VSS	GND	174	BSER6	I/O 2
141	VDD	+3.3V	175	BCLK6	IN 1
142	ACLK3	I/O 2	176	VDD	+3.3V
143	VSS	GND	177	ACLK7	I/O 2
144	ASER3	I/O 2	178	VSS	GND
145	BSCE3	I/O 2	179	ASER7	I/O 2
146	BSER3	I/O 2	180	BSCE7	I/O 2
147	BCLK3	IN 1	181	BSER7	I/O 2
148	VDD	+3.3V	182	BCLK7	IN 1
149	ACLK4	I/O 2	183	VDD	+3.3V
150	VSS	GND	184	M20INT	OUT 1
151	ASER4	I/O 2	185	GTBRBSY0	OUT 1
152	BSCE4	I/O 2	186	GTBRBSY1	OUT 1
153	BSER4	I/O 2	187	STBRBSY0	OUT 1
154	BCLK4	IN 1	188	STBRBSY1	OUT 1

TABLE 38. Pin Listing

Pin	*Pin Label*	*Pin Type*	*Pin*	*Pin Label*	*Pin Type*
189	VSS	GND	215	GATH_ADRS8	OUT 1
190	GATH_DAT15	I/O 3	216	GATH_ADRS7	OUT 1
191	GATH_DAT14	I/O 3	217	GATH_ADRS6	OUT 1
192	GATH_DAT13	I/O 3	218	VDD	+3.3V
193	GATH_DAT12	I/O 3	219	GATH_ADRS5	OUT 1
194	VDD	+3.3V	220	GATH_ADRS4	OUT 1
195	GATH_DAT11	I/O 3	221	GATH_ADRS3	OUT 1
196	GATH_DAT10	I/O 3	222	GATH_ADRS2	OUT 1
197	GATH_DAT9	I/O 3	223	VDD	+3.3V
198	GATH_DAT8	I/O 3	224	VSS	GND
199	VSS	GND	225	VSS	GND
200	GATH_DAT7	I/O 3	226	GATH_ADRS1	OUT 1
201	GATH_DAT6	I/O 3	227	GATH_WE0_	OUT 1
202	GATH_DAT5	I/O 3	228	GATH_WE1_	OUT 1
203	GATH_DAT4	I/O 3	229	GATH_OE0_	OUT 1
204	VDD	+3.3V	230	GATH_OE1_	OUT 1
205	GATH_DAT3	I/O 3	231	No connect	
206	VSS	GND	232	VDD	+3.3V
207	VDD	+3.3V	233	GATH_ADRS18	OUT 1
208	VDD	+3.3V	234	GATH_ADRS17	OUT 1
209	VSS	GND	235	GATH_ADRS16	OUT 1
210	GATH_DAT2	I/O 3	236	GATH_ADRS15	OUT 1
211	GATH_DAT1	I/O 3	237	GATH_ADRS14	OUT 1
212	GATH_DAT0	I/O 3	238	VSS	GND
213	VSS	GND	239	GATH_ADRS13	OUT 1
214	GATH_ADRS9	OUT 1	240	GATH_ADRS12	OUT 1

a. See “VDD_PLLVCC decoupling” on page 180.

Table 39 provides descriptions for the pin types referred to in Table 38 on page 171.

TABLE 39. Pin Type Descriptions

Pin type	***Description***
IN 1	Input buffer, 5V tolerant, 3.3V Schmitt Input
IN 2	Input buffer, 5V tolerant, 3.3V TTL Input
IN 3	Input buffer, 3.3V CMOS Input
IN 4	Input buffer, 5V tolerant, 3.3V CMOS Input, PLL REFCLK
OUT 1	Tri-state-able output buffer, 4 ma source, Noise-isolated, 3.3V TTL
I/O 1	Bi-directional buffer, 4 ma source, 5V tolerant, 3.3V TTL Input
I/O 2	Bi-directional buffer, 8 ma source, 5V tolerant, 3.3V TTL Input
I/O 3	Bi-directional buffer, 4 ma source, 3.3V TTL Input, not 5V tolerant
+3.3V	+3.3 VDC (±5%)
+3.3V1	+3.3 VDC (±5%) Power for PLL; see “VDD_PLLVCC decoupling” on page 180
+5V/3.3V	+5 VDC (±10%). This supply can be 3.3 Volts if 5 Volt tolerant I/O’s aren’t necessary.
GND	Ground

For the proper treatment of pins that are not utilized within a design, see Table 37, “Unused pin termination - general pins,” on page 170.

CHAPTER 10 *Electrical Parameters*

This chapter provides information about the electrical parameters of the MXT3020. The following topics are included:

- MXT3020 Operating conditions and maximum ratings
- MXT3020 Power sequencing
- MXT3020 Phase Lock Loop (PLL) implementation

MXT3020 MAXIMUM RATINGS AND OPERATING CONDITIONS

TABLE 40. Absolute maximum ratings[1] (V_{SS} = 0V)

Symbol	*Parameter*	*Min*	*Max*	*Units*
VDD3	3.3 volt supply	-0.3	3.63	V
VDD5	I/O supply[a]	VDD3	7.0	V
V_{IN}	Input voltage (IN1, IN2, IN3, IN4, I/O1, I/02)	-0.5	VDD5 + 0.3	V
I_{CLAMP}	Input clamp current	-10	10	ma
T_A	Operating free-air temperature range	0	70	°C
T_{STG}	Storage temperature range	-40	125	°C

a. VDD5 must be 5 volts only if 5-volt tolerant I/Os are required. If such I/Os are not required, VDD5 can be 3.3V.

TABLE 41. Recommended operating conditions

Symbol	*Parameter*	*Min*	*Max*	*Units*
VDD3	3.3 volt supply (5% tolerance)	3.135	3.47	V
V_{IH}	High-level input voltage[a]	2	VDD5+.3	V
V_{IL}	Low-level input voltage[b]	-0.3	0.8	V
I_{OH} & I_{OL}	Output current - OUT1		4	ma
	Output current - I/O1		4	ma
	Output current - I/O2		8	ma
T_J	Operating junction temperature	0	125	°C

a. Characterized with VDD3 10% tolerance range (2.97V - 3.63V).

b. Characterized with VDD3 10% tolerance range (2.97V - 3.63V).

1. Stresses beyond the “Absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only. Operation at conditions beyond the indicated “Recommended operating conditions” is not recommended and may adversely affect device reliability.

DC electrical characteristics

TABLE 42. DC Electrical characteristics

Symbol	*Parameter*	*Min*	*Max*	*Units*
ICC3	3.3 volt supply current (33 MHz)		212	mA
	3.3 volt supply current (40 MHz)		258	mA
	3.3 volt supply current (50 MHz)		303	mA
	3.3 volt supply current (60 MHz)		379	mA
ICC5[a]	5 volt supply current		10	mA
V_{OH}	VDD = min, IOH = max	2.4		V
V_{OL}	VDD = min, IOH = max		0.4	V
C_I	Pin capacitance	Typ	6	pF
L_I	Pin inductance	8	10	nH

a. The VDD circuit is a clamping diode providing overshoot protection. ICC5 = Iclamp = 10 ma. The total ICC for the MXT3020 is ICC3 + ICC5.

MXT3020 POWER SEQUENCING

With the exception of optional use of VDD5 as a bias voltage for diodes in the output driver section, the MXT3020 uses a single voltage, +3.3 VDC ±5%. Therefore, there is no need to follow multiple voltage power sequencing rules.

MXT3020 PLL CONSIDERATIONS

Overview

The MXT3020 has an internal Phase Lock Loop (PLL) which it uses to generate the on-chip clock. This PLL allows the on- chip clock tree delay to be neutralized, and optimum performance of the IC to be obtained. The on-chip PLL can be affected by external circuit noise, so careful circuit design must be employed to optimize the performance of the PLL.

Degradation of the PLL performance manifests itself as jitter. This jitter is measured as the timing variation of the chip's internal clock to a stable reference clock supplied to the chip on the FN pin (pin 48). The internal clock can be observed directly; any jitter on the internal clock will show up as jitter on the CLKOB signal (pin 52). Jitter will cause a variation in the timing of the chip relative to the board clock. The timing variation will affect setup and hold timing and erode timing margins at the chip interface.

The following sections cover circuit design issues which affect the operation of the PLL. Key areas of interest are de-coupling, creating a quiet PLL VDD, and ensuring a good PCB layout of the PLL area.

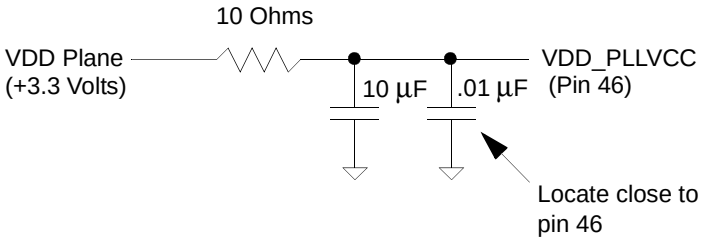
VDD_PLLVCC decoupling

The PLL has a separate power pin labeled VDD_PLLVCC (pin 46). This pin must be supplied with a very stable voltage level and should be well decoupled. The current draw of this pin is very low, 2.5 mA nominal. The low current draw allows the voltage to be isolated from the 3.3V power plane with a resistor. Due to the low current draw, a resistor is recommended for isolating this supply from the main power plane. The VDD_ PLLVCC pin should also be bypassed with a combination of a 10µF tantalum cap and a 0.01µF ceramic cap as shown in Figure 64.

If the VDD_PLLVCC pin is supplied voltage from a linear regulator, the designer must ensure that enough current is being drawn to keep the regulator in regulation. The output of a linear regulator is essentially noise free.

While not required, it is highly recommended that VSS_PLLGND have its own via.

FIGURE 64Generating a quiet VDD_PLLVCC



General decoupling

The MXT3020 must be properly decoupled to ensure clean PLL operation. The PLL is most sensitive to noise on the VDD supply. VDD noise contains both low frequency and high frequency components. Power supply switching noise or insufficient bulk decoupling causes low frequency VDD noise. The switching of the digital logic drivers causes high frequency noise. Both of these noise sources must be taken into account to ensure optimum performance.

The MXT3020 is designed in a high speed CMOS process. The device has 3.3V output drivers, and these output drivers require good decoupling to ensure optimum performance. There should be sixteen high frequency decoupling caps between the 3.3V plane and the ground plane. The preferred value is 0.01µF. These should be arrayed around the chip, with four decoupling caps installed on each side. Additionally, there should be a minimum 20µF of bulk decoupling on the supply voltage (VDD) nearby to the chip. This can be a single 22µF tantalum capacitor, or preferably a pair of 10µF tantalum capacitors.

In a switching power supply environment, it is beneficial to filter the switching noise. This can be accomplished by filtering the MXT3020's VDD with a ferrite bead. The ferrite bead works in conjunction with the bulk decoupling capacitors to effectively filter the power supply switching noise. The ferrite bead must be sized to handle the current draw of the entire chip. An appropriate part is the FairRite 2743021446 surface mount ferrite bead.

FIGURE 65MXT3020 decoupling capacitor location

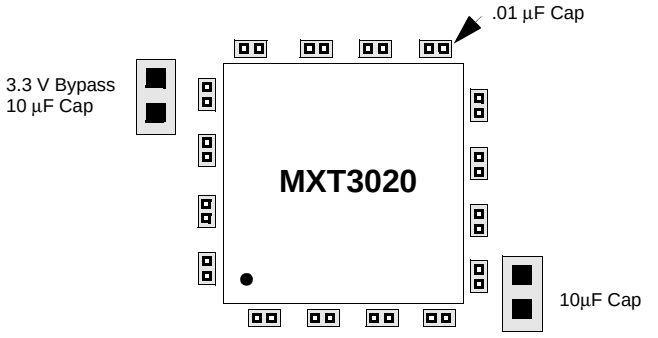


Figure 65 shows the optimal location of the decoupling capacitors around the MXT3020. This diagram depicts the location of 0805 size 0.01µF capacitors under the chip pin pads on the bottom side of the board. The capacitors are located close to the associated power pins. The capacitor should share a common via with the power pin of the chip with a short and wide etch. The same should be done with the ground connections. Etch connections between the devices and the planes should be made as short as possible.

Reference clock jitter

The PLL of the MXT3020 locks the internal chip clock to the reference clock supplied to the device. The PLL will not necessarily be able to track jitter which is on the reference clock. If there is significant jitter on the reference, and the chip clock does not track it, the jitter will cause a reduction in timing margin at the chip interface. If reference clock jitter is present, the system timing budget should be derated by that amount.

Jitter on the reference clock can be caused by power supply noise affecting components of the clock generation and distribution circuit. One potential source of jitter is power supply noise or poor decoupling of crystal oscillators. Noise on the oscillator power pin, whether from the board or self-induced, can convert to timing jitter at the

oscillator output. Some devices are better than others in this aspect of operation. To reduce this noise source, ensure that the oscillator is well decoupled according to the manufacturer's specifications.

The distribution of the reference clock can also introduce clock jitter. Designs that use dividers in the reference clock path must avoid the possibility of simultaneous switching jitter, which can occur in synchronous counters. PLL clock buffers can also be a source of jitter, as these devices are generally susceptible to power supply noise, and can convert this noise to timing jitter.

CHAPTER 11 *Mechanical and Thermal Information*

This chapter provides information on the MXT3020 mechanical and thermal properties.

MXT3020 MECHANICAL/THERMAL INFORMATION

The MXT3020 is packaged in a 240-pin enhanced quad flat-pack. Figure 66 shows the pin configuration and package dimensions.

FIGURE 66MXT3020 package/pin diagram - top view

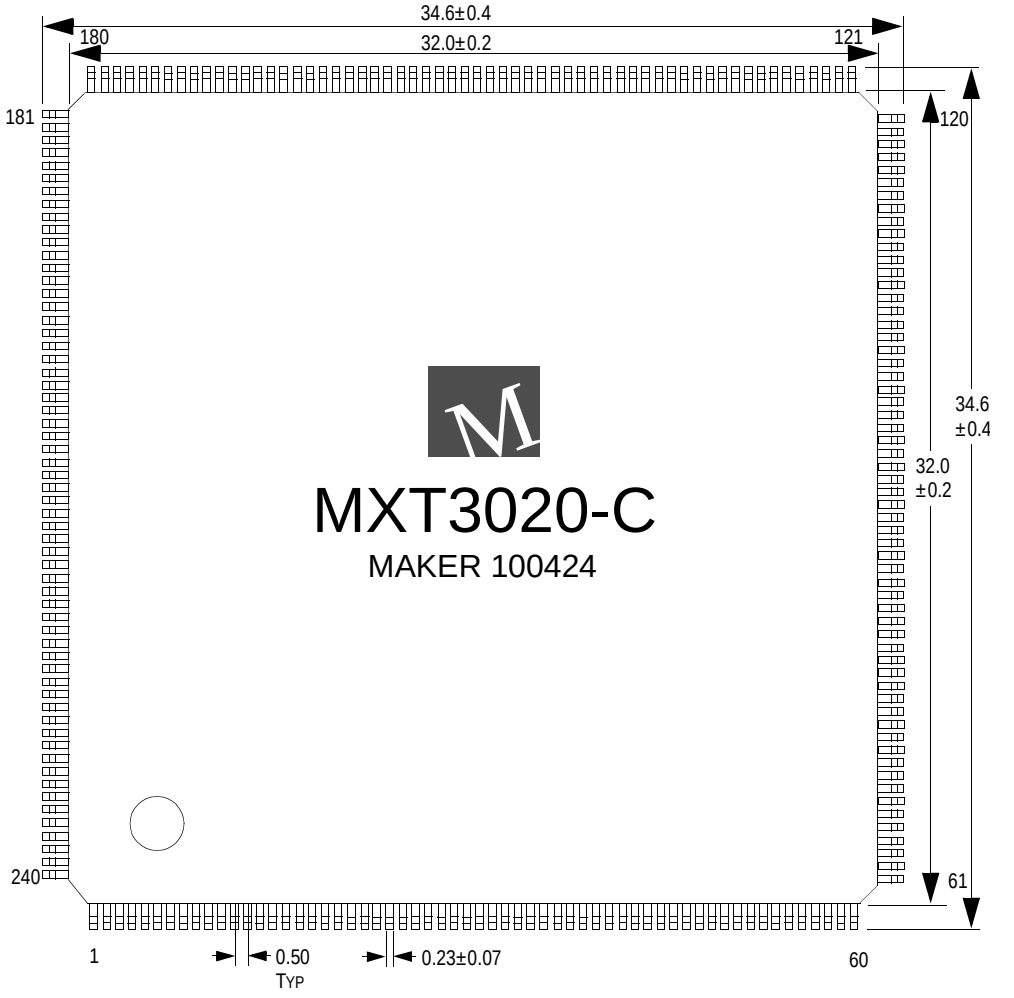


FIGURE 67MXT3020 package/pin diagram - side view

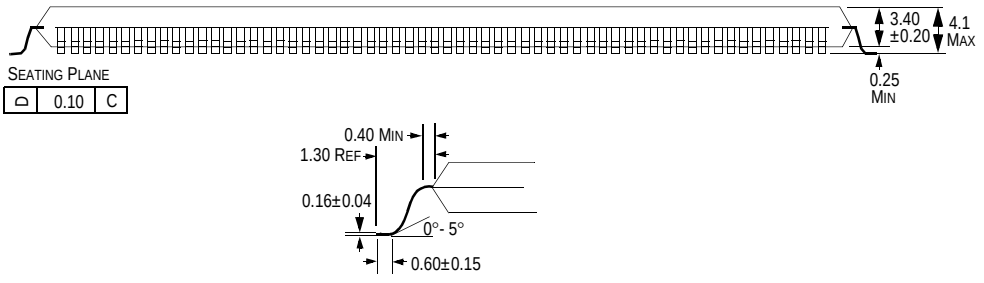


TABLE 43. MXT3020 package summary

Package			***θjc (×C/W)***	***θja (×C/W)***	
Package Type	***Body size (mm)***	***Lead pitch (mm)***		****Still Air***	****Air Flow***
PQFP 240	32.0 x 32.0 x 3.6	0.5	6.5	30	20

* These numbers will vary depending on the board stack-up and orientation. All airflow numbers are quoted with 1m/sec of air flow over the device.

Storage conditions

The MXT3020 is a level 3 IAW IPC-SM-786A or JESD 22-A112 device. The MXT3020's safe floor life (out of bag) prior to solder reflow is 1 week at ≤ 30°C/60% RH.

APPENDIX A *Acronyms*

Acronym	*Definition*
AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACR	Allowed Cell Rate
ATM	Asynchronous Transfer Mode
CAM	Content Addressable Memory
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CI	Congestion Indicator
CLP	Cell Loss Priority
CPCS	Common Part Convergence Sublayer
CPI	Common Part Identifier
CRC	Cyclic Redundancy Check
CSS	Cell Scheduling System
DMA	Direct Memory Access
E1	European 2.048 Mbps rate TDM system
EFCI	Explicit Forward Congestion Indicator
ESS	External State Signals
FIFO	First In First Out
GCRA	Generic Cell Rate Algorithm
GFC	General Flow Control

Acronym	*Definition*
HEC	Header Error Control
ICS	Interchip Communication System
IFO	Instruction Field Options
J2	96-channel TDM system used by Japan Telephone
MIB	Management Information Base
MVIP	Muti-Vendor Integration Protocol™
OAM	Operations and Management
PCR	Peak Cell Rate
PDU	Physical Data Unit
PHY	Physical Layer
PIT	Programmable Interval Timers
PTI	Payload Type Identifier
RAM	Random Access Memory
RM	Resource Management
RX	Receive
SAR	Segmentation and Reassembly
SCSA	Signal Computing System Architecture, ANSI standard
SDU	Service Data Unit
SHFM	Store Halfword to Fast Memory
SRAM	Random Access Memory
SRTS	Synchronous Residual Time Stamp
SWAN	Soft-Wired ATM Network
TDM	Time Division Multiplexing
T1	24-channel TDM system used in North America
TX	Transmit
UBR	Undefined Bit Rate
UDT	Unstructured Data Transfer
UU	User-to-User
VBR	Variable Bit Rate
VC	Virtual Channel
VCI	Virtual Channel Identifier
VP	Virtual Path
VPI	Virtual Path Identifier

APPENDIX B *Registered Decoder PAL Source Code*

```
;pal2v_translate_on
; Tpd = #10
; Tco = #8
;pal2v_translate_off
;

;PALASM Design Description
;
Title           mxt3020_dec.pds
Date            6/24/98
Revision        4.0
Author          Joseph Tompkins
Company         Maker Communications
Pattern         A

CHIP DECODE     PAL26V12

; Revision History
;
; Rev 1.0    Initial release
; Rev 2.0    Changed to PAL26V12.
;            Added inputs: m20_int, m21_int, m22_int, m23_int
;            Added output: m20int
; Rev 3.0    Changed to MXT3020 select to be a non-burst read
;            instead of a write. Select register has been
;            moved to 0x600000 from 0x200000.
; Rev 4.0    "sel_code0" was combinatorial instead of registered because
;            of a typo in the clock definition statement. The definition
```

```
;            for sel_code0 was written as "sel_code1.clkf = clk" instead
;            of "sel_code0.clkf = clk"
;
;=============================================================================
;              Non-Burst Decode Logic
;
;             P2 Non-Burst Address Map
;
;       MXT3010                     P2 Bus {P2AI[3:2],P2AD[15:0]}
;       Address                     Address
;
;      0x800000 +------------------+ 0x40000          ^
;               | mxt3020_sel_reg  |                  |
;      0x600000 +------------------+ 0x30000          |
;               | mxt3020_bsy_reg  |
;      0x400000 +------------------+ 0x20000       512 Kbytes
;               |    not used      |
;      0x200000 +------------------+ 0x10000          |
;               |    not used      |                  |
;      0x000000 +------------------+                  v
;
;             MXT3020 Select Registers
;
;      0x800000 +------------------+  0x40000
;               |                  |
;               |                  |
;               |                  |
;               |    not used      |
;               |                  |
;               |                  |
;               |                  |
;      0x600080 +------------------+  0x40004
;               | Select MXT3020 #3|
;      0x600060 +------------------+  0x40003
;               | Select MXT3020 #2|
;      0x600040 +------------------+  0x40002
;               | Select MXT3020 #1|
;      0x600020 +------------------+  0x40001
;               | Select MXT3020 #0|
;      0x600000 +------------------+  0x40000
;
;        MXT3010                     P2 Bus
;        Address                     Address
```

```
;
;
;
;          MXT3020 TBR Busy Flag Register
;
;        MXT3010                    P2 Bus
;        Address                    Address
;               15                 0
;               +---+----+----+----+
;      0x400000 |#3 | #2 | #1 | #0 |  0x20000
;               +---+----+----+----+
;                |    |    |    |
;                |    |    |    +-- [3:0]   = {STBR_BSY[1:0], GTBR_BSY[1:0]}
;                |    |    +------- [7:4]   = {STBR_BSY[1:0], GTBR_BSY[1:0]}
;                |    +------------ [11:8]  = {STBR_BSY[1:0], GTBR_BSY[1:0]}
;                +----------------- [15:12] = {STBR_BSY[1:0], GTBR_BSY[1:0]}
;
;
;============================================================================

;== Inputs ==;

pin     1  clk
pin     2  p2_ai3       ; Port 2 Address Index [3]
pin     3  p2_ai2       ; Port 2 Address Index [2]
pin     4  p2_req_      ; Port 2 Request (low)
pin     5  p2_rd        ; Port 2 Read/Write_ (high)
pin     6  p2_qbrst     ; Port 2 Burst/NonBurst_ (high)
;pin    7  vcc          ; power pin
pin     8  p2_ad1       ; Port 2 Address/Data line [1]
pin     9  reset_       ; Reset (low)
pin    10  p2_end_      ; Port 2 End (low)
;pin   11  nc_11        ; no connect
pin    12  p2_asel_     ; Port 2 Address Phase Select (low)
pin    13  p2_ad0       ; Port 2 Address/Data line [0]
pin    14  m20_int      ; MXT3020 #0 interrrupt
;pin   21  gnd_21       ; ground pin
pin    25  m23_int      ; MXT3020 #3 interrrupt
pin    26  m22_int      ; MXT3020 #2 interrrupt
pin    27  m21_int      ; MXT3020 #1 interrrupt
;pin   28  nc_28        ; no connect

;
```

```
;== Outputs ==;

pin    15  mxt3020_sel2_  registered   ; MXT3020 #2 select (low)
pin    16  mxt3020_sel1_  registered   ; MXT3020 #1 select (low)
pin    17  mxt3020_sel0_  registered   ; MXT3020 #0 select (low)
pin    18  sel_code1      registered   ; mxt3020_sel encode bit (internal)
pin    19  sel_code0      registered   ; mxt3020_sel encode bit (internal)
pin    20  mxt3020_sel3_  registered   ; MXT3020 #3 select (low)
pin    22  m20int         combinatorial; MXT3020 Interrupt
;pin   23  io_23                       ; unused io
;pin   24  io_24                       ; unused io

equations

;================================================================================
; The MXT3020 Interrupt output is the logical OR of the interrupts from the 4
MXT3020
; chips on the motherboard. The output is tied to the ICSI_C pin of the MXT3010-
EP.
; The interrupt asserts when a gather or scatter completes inside an MXT3020.
;================================================================================

m20int = m20_int + m21_int + m22_int + m23_int

;================================================================================
; The MXT3020 selects are generated by a registered 2-to-4 decoder. The decoder
; can be accessed by the MXT3010 via the Port 2 Bus using a non-burst read.
; The most-significant two Port2 non-burst address bits (P2_AI[3:2]) are
; decoded to select one of four regions of PORT 2 non-burst space. If the fourth
; region is accessed then the MXT3020 Select Register is selected. The least-
; significant two bits of the non-burst address are then captured by the Select
; Register logic as "SEL_CODE[1:0]". Only READ operations are decoded.
;
;  P2_REQ_ P2_ASEL_ P2_RD P2_QBRST P2_AI[3:2] P2_AD[1:0]   SEL_CODE[1:0]
; ------------------------------------------------------------+-------------
;     0       0       1       0       11          00       |  00 <reset value
;     0       0       1       0       11          01       |  01
;     0       0       1       0       11          10       |  10
;     0       0       1       0       11          11       |  11
;    |__________|     |       |      |__|                   |
;
;    Address Phase   Read   Non-Burst Fourth      MXT3020   No connect, used
;                                     Region      Select    internally in PAL
;The first term in each equation is the set term. All others are hold terms.
```

```
;=============================================================================

sel_code1.clkf = clk

sel_code1 =
    (/p2_req_   * /p2_asel_ * p2_rd * /p2_qbrst  *  p2_ai3 *  p2_ai2 * p2_ad1
* reset_) +
    (/p2_req_   * /p2_asel_ * p2_rd * /p2_qbrst  * /p2_ai3 * /p2_ai2 * sel_code1
* reset_) +
    (/p2_req_   * /p2_asel_ * p2_rd * /p2_qbrst  *  p2_ai3 * /p2_ai2 * sel_code1
* reset_) +
    (/p2_req_   * /p2_asel_ * p2_rd * /p2_qbrst  * /p2_ai3 *  p2_ai2 * sel_code1
* reset_) +
     ( p2_req_   * sel_code1 * reset_) +
     ( p2_asel_  * sel_code1 * reset_) +
     (/p2_rd     * sel_code1 * reset_) +
     ( p2_qbrst  * sel_code1 * reset_)

sel_code0.clkf = clk

sel_code0 =
     (/p2_req_   * /p2_asel_ * p2_rd * /p2_qbrst  *  p2_ai3 *  p2_ai2 * p2_ad0
* reset_) +
    (/p2_req_   * /p2_asel_ * p2_rd * /p2_qbrst  * /p2_ai3 * /p2_ai2 * sel_code0
* reset_) +
    (/p2_req_   * /p2_asel_ * p2_rd * /p2_qbrst  *  p2_ai3 * /p2_ai2 * sel_code0
* reset_) +
    (/p2_req_   * /p2_asel_ * p2_rd * /p2_qbrst  * /p2_ai3 *  p2_ai2 * sel_code0
* reset_) +
     ( p2_req_   * sel_code0 * reset_) +
     ( p2_asel_  * sel_code0 * reset_) +
     (/p2_rd     * sel_code0 * reset_) +
     ( p2_qbrst  * sel_code0 * reset_)


;=============================================================================
; The SEL_CODE[1:0] bits are used to assert one of the four MXT3020 select
; lines on the next P2 Burst Space read or write operation. Until a burst space
; operation is detected the previously asserted MXT3020 select will be
; maintained. At reset, mxt3020_sel0_ is asserted and all other selects are
; deasserted.
;
;=============================================================================
;
```

```
mxt3020_sel3_.clkf = clk

mxt3020_sel3_ =
  /((/p2_req_  * /p2_asel_ *  p2_qbrst *  sel_code1 *  sel_code0 * reset_) +
     ( p2_req_   * /mxt3020_sel3_ * reset_) +
    ( p2_asel_  * /mxt3020_sel3_ * reset_) +
     (/p2_qbrst  * /mxt3020_sel3_ * reset_))

mxt3020_sel2_.clkf = clk

mxt3020_sel2_ =
  /((/p2_req_  * /p2_asel_ *  p2_qbrst *  sel_code1 * /sel_code0 * reset_) +
     ( p2_req_   * /mxt3020_sel2_ * reset_) +
    ( p2_asel_  * /mxt3020_sel2_ * reset_) +
     (/p2_qbrst  * /mxt3020_sel2_ * reset_))


mxt3020_sel1_.clkf = clk

mxt3020_sel1_ =
  /((/p2_req_  * /p2_asel_ *  p2_qbrst * /sel_code1 *  sel_code0 * reset_) +
     ( p2_req_   * /mxt3020_sel1_ * reset_) +
    ( p2_asel_  * /mxt3020_sel1_ * reset_) +
     (/p2_qbrst  * /mxt3020_sel1_ * reset_))

mxt3020_sel0_.clkf = clk

mxt3020_sel0_ =
  /((/p2_req_  * /p2_asel_ *  p2_qbrst * /sel_code1 * /sel_code0 * reset_) +
     ( p2_req_   * /mxt3020_sel0_ * reset_) +
    ( p2_asel_  * /mxt3020_sel0_ * reset_) +
     (/p2_qbrst  * /mxt3020_sel0_ * reset_) +
     (/reset_))
```

Index