PM7390

CONFIGURING THE PM7390 S/UNI MACH 48

PRELIMINARY

ISSUE 2: FEBRUARY 2001

PUBLIC REVISION HISTORY

Issue No.	Issue Date	Details of Change
1	Oct 2000	Document Created
2	Jan 2001	1. Added Section 7: “TCL Code examples”. 2. Section 6.1.11, Table 42: Change RCAS/TCAS Enable to all timeslots per the data sheet 3. Tables 34, 35, 36, 41, 54, 55, 56, 57, 92, 93,94,96 97: Changed headings from UL3/PL3 to ATM/POS to clarify functions. 4. Table 94 and Table 96: Corrected entries for POS and ATM traffic 5. Section 5.10: Added note that SIRP must be enabled for correct operation of the device 6. Section 6.3.5: Modified STS-1 example SDQ block pointer configuration to allow for dynamic reconfiguration to STS-3c. 7. Section 6.4.5: Modified SDQ block pointer configuration to allow for dynamic reconfiguration to STS-3c. 8. Section 6.4.6, Table 94: Corrected DS3 HDLC entries for offset 02 to 0x403 from 0x402 to allow bit swapping which is required for bit synchronous HDLC.
		9. Section 6.4.7, Table 96: Corrected DS3 HDLC entries for offset 03 to 0x7 from 0x3 to allow bit swapping which is required for bit synchronous HDLC. 10. Added Section 5.7.1: “FIFO Arrangement in the SDQs”

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 DEFINITIONS

RCAS	Receive Channel Assigner
TCAS	Transmit Channel Assigner
RXPHY	Receive UL3/PL3 Physical Layer Interface
TXPHY	Transmit UL3/PL3 Physical Layer Interface
RXSDQ	Receive Scalable Data Queue
TXSDQ	Transmit Scalable Data Queue
RCFP	Receive Cell and Frame Processor (STS-12c/STS-48c)
TCFP	Transmit Cell and Frame Processor (STS-12c/STS-48c)
RTDP	Receive Timeslice Datacom Processor (STS-3c/STS-1/DS-3)
TTDP	Transmit Timeslice Datacom Processor (STS-3c/STS-1/DS-3)
T8TE	Transmit 8B/10B Telecombus Encoder
R8TD	Receive 8B/10B Telecombus Decoder
SPLR	SMDS PLCP Layer Receiver
SPLT	SMDS PLCP Layer Transmitter

2 SCOPE

This document is intended as an introductory guide for hardware engineers to configure the S/UNI MACH48 device on a register access level.

3 OVERVIEW

The S/UNI MACH48 is an ATM and packet processor capable of processing data streams from STS-48/STM-16 with STS-1/STM-0/DS3 granularity for a total aggregate bandwidth of 2.488 Gbps. From the line side, the Serial (S-TCB) or Parallel Telecombus (P-TCB) interface takes SONET/SDH framed data and extracts ATM cells or POS packets for transmission onto a UL3/PL3 bus. The MACH48 can extract any combination of cells and packets. Towards the line side, the MACH48 maps cells and packets received from the UL3/PL3 bus into SONET/SDH frames. The SONET/SDH frames are then transmitted via the Parallel or Serial Telecombus.

This document outlines several basic setup examples involving channels of various bandwidths using either ATM or POS traffic. The following combinations are illustrated:

- 1 PHY: 1 x STS-48c/STM-
- 16 PHYS: 16 x STS-3c/STM-1
- 48 PHYS: 48 x STS-1/STM-0
- 25 PHYS: 1 x STS-12c/STM-4, 6 x STS-3c/STM-1, 9 x STS-1/STM-0, 9 x DS3

4 REFERENCES

1. PMC-1990823, S/UNI MACH48 Data Sheet Issue 3, May 2000.

5 REGISTER SETUP

The MACH48 can be set up in the following order:

- Configuration Register
- Timeslot Configuration
- Telecombus
- RXPHY Calendar
- Scalable Data Queue (SDQ)
- Receive Cell and Frame Processor (RCFP + RTDP)
- Transmit Cell and Frame Processor (TCFP + TTDP)
- SONET/SDH Inband Error Report Processor(SIRP)
- Receive Channel Assigner (RCAS)
- Transmit Channel Assigner (TCAS)

The following sections highlight registers of interest with specific reference to a particular setup configuration. Not all registers are necessarily listed.

5.1 Writing and Reading Indirect Registers

5.1.1 Writing Indirect Registers

Several blocks in the S/UNI MACH48 use indirect registers.

The algorithm for writing to an indirect register is illustrated below:

1. Poll the BUSY bit that is in the Indirect Address register that is to be written until it reads 0.

2. Write the appropriate indirect data and/or indirect configuration registers

3. Write the Indirect Register Address last with the RWB bit set to 0. This will initiate the write to that specific PHYID.

5.1.2 Reading Indirect Registers

To initiate a read from an indirect register, do the following:

1. Write the Indirect Address Register with the desired PHYID and the RWB bit set to 1. This will initiate the read.

2. Poll the BUSY bit that is in the base address of the register that is to be written until it reads 0.

3. Read the appropriate indirect data and/or indirect configuration registers.

Note: The indirect data and configuration registers should not be read except during a indirect read as they will be indeterminate values.

5.2 Valid Channel Assignments

The S/UNI MACH48 can service up to 48 PHYs of different bandwidths for a total of 2.488Gbps. The granularity of the channelization is STS-1/STM-0. Though the data streams can be channelized into virtually any combination of bandwidths, certain guidelines must be followed to ensure correct operation of the device. Specified channel assignments require specific FIFO and timeslot associations. The table below shows the different legal channel mappings. The three righmost columns are associated with example 4 in this document. Items in bold indicate the valid mappings for that example. Note that in the example, no lower bandwidth channel occupies a timeslot that is used by a higher

bandwidth channel (e.g. STS-3c E and F must occupy timeslots 0,4,8 and 2,6,10 respectively while the other STS-1/DS-3 channels may occupy the timeslots that remain).

Table 1 S/UNI MACH48 Legal Channel Mappings

Channel PHYID	STS-48c STM-16c	STS-12c STM-4c	STS-3c STM-1	STS-1 STM-0/ DS3	Mixed Example	FIFO NUM	Timeslot
0	a	a	a	a	STS-3c A	**0**	0
1	a	a	b	b	STS-3c B	**1**	1
2	a	a	c	c	STS-3c C	**2**	2
3	a	a	d	d	STS-3c D	**3**	3
4	a	a	a	e	STS-3c A	4	4
5	a	a	b	f	STS-3c B	5	5
6	a	a	c	g	STS-3c C	6	6
7	a	a	d	h	STS-3c D	7	7
8	a	a	a	I	STS-3c A	8	8
9	a	a	b	j	STS-3c B	9	9
10	a	a	c	k	STS-3c C	10	10
11	a	a	d	l	STS-3c D	11	11
12	a	b	e	m	STS-12c A	**16**	0
13	a	b	f	n	STS-12c A	17	1
14	a	b	g	o	STS-12c A	18	2
15	a	b	h	p	STS-12c A	19	3
16	a	b	e	q	STS-12c A	20	4
17	a	b	f	r	STS-12c A	21	5
18	a	b	g	s	STS-12c A	22	6
19	a	b	h	t	STS-12c A	23	7
20	a	b	e	u	STS-12c A	24	8
21	a	b	f	v	STS-12c A	25	9
22	a	b	g	w	STS-12c A	26	10
23	a	b	h	x	STS-12c A	27	11
24	a	c	I	y	DS-3 A	**32**	0
25	a	c	j	z	DS-3 B	**33**	1
26	a	c	k	aa	STS-1 A	**34**	2
27	a	c	l	ab	STS-1 B	**35**	3

Channel PHYID	STS-48c STM-16c	STS-12c STM-4c	STS-3c STM-1	STS-1 STM-0/ DS3	Mixed Example	FIFO NUM	Timeslot
28	a	c	l	ac	STS-1 C	**36**	4
29	a	c	j	ad	DS-3 C	**37**	5
30	a	c	k	ae	DS-3 D	**38**	6
31	a	c	l	af	STS-1 D	**39**	7
32	a	c	l	ag	DS-3 E	**40**	8
33	a	c	j	ah	STS-1 E	**41**	9
34	a	c	k	ai	DS-3 F	**42**	10
35	a	c	l	aj	STS-1 F	**43**	11
36	a	d	m	ak	STS-3c E	**48**	0
37	a	d	n	al	STS-1 G	**49**	1
38	a	d	o	am	STS-3c F	**50**	2
39	a	d	p	an	DS-3 G	**51**	3
40	a	d	m	ao	STS-3c E	52	4
41	a	d	n	ap	STS-1 H	**53**	5
42	a	d	o	aq	STS-3c F	54	6
43	a	d	p	ar	DS-3 H	**55**	7
44	a	d	m	as	STS-3c E	56	8
45	a	d	n	at	STS-1 I	**57**	9
46	a	d	o	au	STS-3c F	58	10
47	a	d	p	av	DS-3 I	**59**	11

5.3 Configuration Register

The Configuration register controls global reset and loopback functions.

Table 2 Configuration Register

Register Address	DESCRIPTION
0001h	S/UNI MACH48 Master Reset, Configuration, and Global Digital Loopback

5.4 Timeslot Configuration

These registers set up the system side timeslots. See Table 9 and 11 in the S/UNI MACH 48 Data Sheet for valid timeslot mappings.

Table 3 Timeslot Configuration Registers

Register Address	DESCRIPTION
0002h	Receive Timeslot Configuration #1
0003h	Receive Timeslot Configuration #2
0004h	Receive Timeslot Configuration #3
0005h	Receive Timeslot Configuration #4
0006h	Receive Timeslot Configuration #5
0007h	Receive Timeslot Configuration #6
0008h	Transmit Timeslot Configuration #1
0009h	Transmit Timeslot Configuration #2
000Ah	Transmit Timeslot Configuration #3
000Bh	Transmit Timeslot Configuration #4
000Ch	Transmit Timeslot Configuration #5
000Dh	Transmit Timeslot Configuration #6

5.5 Serial Telecombus

The Telecombus registers are only used when the line side Serial Telecombus is enabled. The base address for each Telecombus link is as follows:

Table 4 T8TE + R8TD Base Addresses

Base Address(T8TE)	LINK
1860h	TWRK[1]
1868h	TWRK[2]
1870h	TWRK[3]
1878h	TWRK[4]
1880h	TPROT[1]
1888h	TPROT[2]
1890h	TPROT[3]
1898h	TPROT[4]
1820h	RWRK[1]
1828h	RWRK[2]
1830h	RWRK[3]
1838h	RWRK[4]
1840h	RPROT[1]
1848h	RPROT[2]
1850h	RPROT[3]
1858h	RPROT[4]

Table 5 T8TE Registers

Register Address	DESCRIPTION
T8TE+00h	T8TE Control and Status
T8TE+01h	T8TE Interrupt Status
T8TE+02h	T8TE Telecombus Mode #1
T8TE+03h	T8TE Telecombus Mode #2
T8TE+04h	T8TE Test Pattern
T8TE+05h	T8TE Analog Control
T8TE+06h	T8TE DTB Bus

Table 6 R8TD Registers

Register Address	DESCRIPTION
R8TD+00h	R8TD Control and Status
R8TD +01h	Interrupt Status
R8TD +02h	Line Code Violation Count
R8TD +03h	Analog Control 1
R8TD +04h	Analog Control 2

Note: Proper operation of the MACH48 device requires RT8D + 03h Analog Control 1 to be written with CC34h.

5.6 RXPHY and TXPHY Setup

The RXPHY and TXPHY registers can be used for the PL3 or UL3 Receive interface. In PL3 mode, the RXPHY registers are used to set up the calendar attributes and burst size.

Table 7 RXPHY Registers

Register Address	DESCRIPTION
0040h	RXPHY Configuration
0041h	RXPHY Interrupt Status
0042h	RXPHY Interrupt Enable
0043h	RXPHY Indirect Burst Size
0044h	RXPHY Calendar Length
0045h	RXPHY Calendar Indirect Address Data
0046h	RXPHY Data Type Field

Table 8 TXPHY Registers

Register Address	DESCRIPTION
0048h	TXPHY Configuration
0049h	TXPHY Interrupt Status
004Ah	TXPHY Interrupt Enable
004Bh	TXPHY Data Type Field

The TPAHOLD bit in the TXPHY register (0048h) is explained in section 15.8.2 of the S/UNI MACH48 Data Sheet. It is set to 0 for all of the examples except the 48 x STS-1 example.

5.7 Scalable Data Queue (SDQ)

The SDQ registers are used to set up Receive and Transmit SDQ attributes.

Table 9 RXSDQ Registers

Register Address	DESCRIPTION
0050h	RXSDQ FIFO Reset
0051h	RXSDQ FIFO Interrupt Enable
0053h	RXSDQ FIFO Overflow Port and Interrupt Indication
0054h	RXSDQ FIFO EOP Error Port and Interrupt Indication
0055h	RXSDQ FIFO SOP Error Port and Interrupt Indication
0058h	RXSDQ FIFO Indirect Address
0059h	RXSDQ FIFO Indirect Configuration
005Ah	RXSDQ FIFO Indirect Data Available Threshold
005Bh	RXSDQ FIFO Indirect Cells and Packets Count
005Ch	RXSDQ FIFO Cells and Packets Accepted Aggregate Count (LSB)
005Dh	RXSDQ FIFO Cells and Packets Accepted Aggregate Count (MSB)
005Eh	RXSDQ FIFO Cells and Packets Dropped Aggregate Count

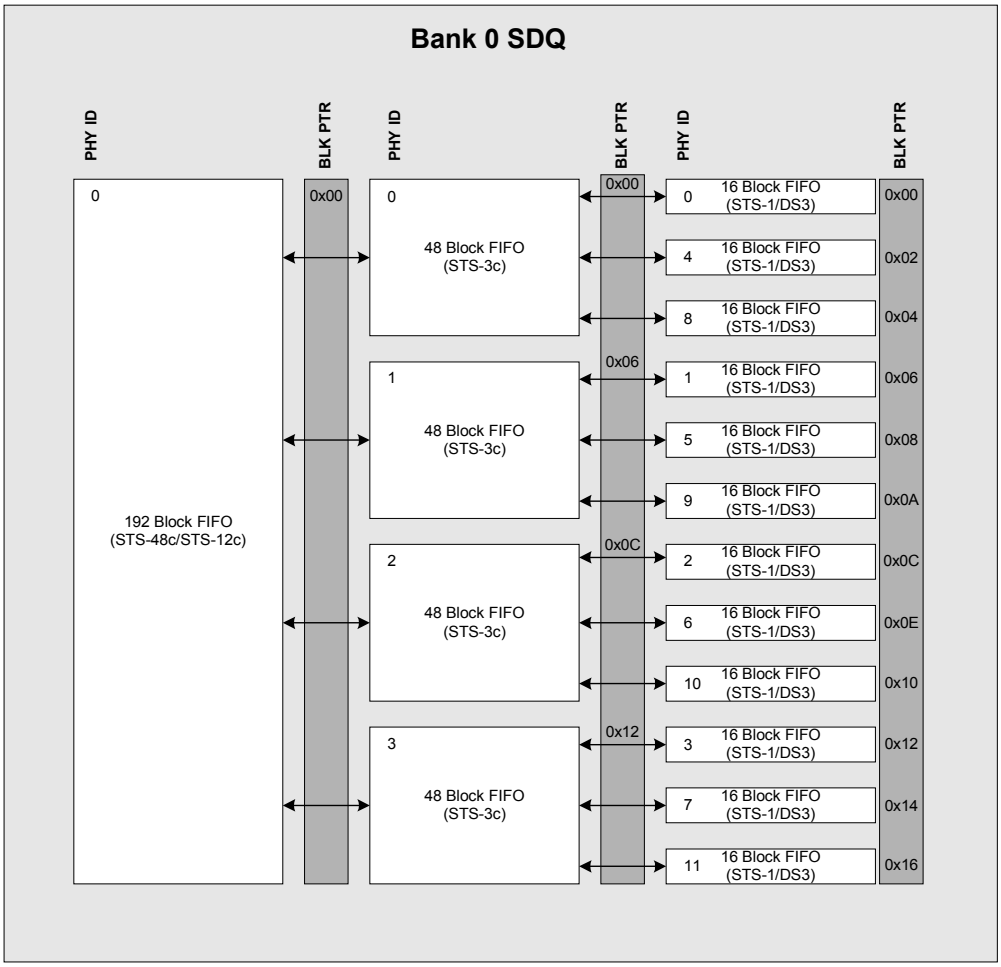
Table 10 TXSDQ Registers

Register Address	DESCRIPTION
0060h	TXSDQ FIFO Reset
0061h	TXSDQ FIFO Interrupt Enable
0063h	TXSDQ FIFO Overflow Port and Interrupt Indication
0064h	TXSDQ FIFO EOP Error Port and Interrupt Indication
0065h	TXSDQ FIFO SOP Error Port and Interrupt Indication
0068h	TXSDQ FIFO Indirect Address
0069h	TXSDQ FIFO Indirect Configuration
006Ah	TXSDQ FIFO Indirect Data Available Threshold
006Bh	TXSDQ FIFO Indirect Cells and Packets Count
006Ch	TXSDQ FIFO Cells and Packets Accepted Aggregate Count (LSB)
006Dh	TXSDQ FIFO Cells and Packets Accepted Aggregate Count (MSB)
006Eh	TXSDQ FIFO Cells and Packets Dropped Aggregate Count

5.7.1 FIFO Arrangement in the SDQs

The figure below shows the bank 0 SDQ that would service PHY 0 for STS-12c, PHYs 0-3 for STS-3c and PHYs 0 to 11 for STS-1. If the PHYs are to be dynamically reconfigured, then the FIFOs must arranged such that only those being reconfigured will be affected. A way to do this is to locate STS-1 FIFOs next to their corresponding members for a larger bandwidth FIFO. As an example, STS-3c channel (phyid) 0 is comprised of timeslots 1, 5 and 9 in STS-12 stream 0. If these were STS-1 channels, they would correspond to channels 0, 4 and 8, respectively. When the STS-1 channels are aggregated into the STS-3c, the STS-1 FIFOs can be absorbed to create a single FIFO. In order to do this, the three STS-1 FIFOs must be located consecutively.

Figure 1 - Bank 0 SDQ FIFO Arrangement



5.7.2 Data/Buffer Available Threshold

The Data Available Threshold and Buffer Available Threshold parameters are explained in Section 14.15 and 14.16 of the S/UNI MACH48 Data Sheet. The equation below outlines the criteria for setting BT and DT.

(DT[7..0] +1) + (BT[4..0] < FIFO SIZE

For the RXSDQ the DT setting is linked to the BURST SIZE with the following criteria.

(BT[4..0] +1) </= BURST SIZE

It is important to note that for UL3, DT[7..0], BT[4..0] and BURST SIZE are set to 0003h (4 blocks= 1 ATM cell) for all FIFO sizes.

Table 11 RXSDQ DT Settings per FIFO

Mode	Bandwidth	Fifo Size	DT(7..0)	BURST SIZE
UL3	STS-48c/STS-12c	192	3	3
	STS-3c	48	3	3
	STS-1	16	3	3
PL3	STS-48c/STS-12c	192	95	3
	STS-3c	48	23	3
	STS-1	16	7	3

It is recommended that the DT setting for the TXSDQ be 1/2 to 2/3 the size of the FIFO. In the examples the values used are 1/2 the size of the FIFO when in PL3 mode.

Table 12 TXSDQ BT/DT Settings per FIFO

Mode	Bandwidth	Fifo Size	BT(4..0)	DT(7..0)
UL3	STS-48c/STS-12c	192	3	3
	STS-12c	192	3	3
	STS-1	16	3	3
PL3	STS-48c/STS-12c	192	7	95
	STS-3c	48	7	23
	STS-1	16	7	7

5.8 Receive Cell and Frame Processor (RCFP + RTDP)

There are two types of Receive Cell and Frame Processor blocks, RCFP (Receive Cell and Frame Processor) and RTDP (Receive Timeslice Datacom Processor). The RCFP blocks are used for STS-12c/STM-4c and STS-48c/STM-16c channels while the RTDP blocks are used for STS-3c/STM-1, STS-1/STM-0, and DS-3 bandwidth channels.

Table 13 RCFP + RTDP Base Addresses

Register Address	DESCRIPTION
0070h	RCFP Configuration #1 (RCFP0_BASE)
0080h	RCFP Configuration #2 (RCFP1_BASE)
0090h	RCFP Configuration #3 (RCFP2_BASE)
00A0h	RCFP Configuration #4 (RCFP3_BASE)
00F0h	RTDP Indirect Channel Select #1 (RTDP0_BASE)
0110h	RTDP Indirect Channel Select #2 (RTDP1_BASE)
0130h	RTDP Indirect Channel Select #3 (RTDP2_BASE)
0150h	RTDP Indirect Channel Select #4 (RTDP3_BASE)

Table 14 RTDP Registers

Register Address	DESCRIPTION
RTDP_BASEx + 01h	RTDP Indirect Configuration
RTDP_BASEx + 02h	RTDP Indirect Minimum Packet Length and Bit Order
RTDP_BASEx + 03h	RTDP Indirect Maximum Packet Length
RTDP_BASEx + 04h	RTDP Indirect LCD Count Threshold
RTDP_BASEx + 05h	RTDP Indirect Idle Cell Header and Mask
(x=1 to 4)	

Table 15 RCFP Registers

Register Address	DESCRIPTION
RCFP_BASEx + 01h	RCFP Interrupt Enable
RCFP_BASEx + 02h	RCFP Interrupt Indication and Status
RCFP_BASEx + 03h	RCFP Minimum Packet Length

Register Address	DESCRIPTION
RCFP_BASEx + 04h	RCFP Maximum Packet Length
RCFP_BASEx + 05h	RCFP LCD Count Threshold
RCFP_BASEx + 06h	RCFP Idle Cell Header and Mask
RCFP_BASEx + 07h	RCFP Receive Byte/Idle Cell Counter (LSB)
RCFP_BASEx + 08h	RCFP Receive Byte/Idle Cell Counter
RCFP_BASEx + 09h	RCFP Receive Byte/Idle Cell Counter (MSB)
RCFP_BASEx + 0Ah	RCFP Packet/Cell Counter (LSB)
RCFP_BASEx + 0Bh	RCFP Packet/Cell Counter (MSB)
RCFP_BASEx + 0Ch	RCFP Receive Erred FCS/HCS Counter
RCFP_BASEx + 0Dh	RCFP Receive Aborted Packet Counter
RCFP_BASEx + 0Eh	RCFP Receive Minimum Length Packet Error Counter
RCFP_BASEx + 0Fh	RCFP Receive Maximum Length Packet Error Counter
(x=1 to 4)	

5.9 Transmit Cell and Frame Processor (TCFP + TTDP)

Like the Receive side, the Transmit side has two types of cell and frame processor blocks. The Transmit Cell and Frame Processor (TCFP) is used to process STS-12c/STM-4c and STS-48c/STM-16c data streams. The Transmit Timeslice Datacom Processor (TTDP) is used to process STS-3c/STM-1 and lower bandwidths.

Table 16 TCFP + TDFP Base Addresses

Register Address	DESCRIPTION
00B0h	TCFP Configuration #1
00C0h	TCFP Configuration #2
00D0h	TCFP Configuration #3
00E0h	TCFP Configuration #4
0170h	TTDP Indirect Channel Select #1
0180h	TTDP Indirect Channel Select #2
0190h	TTDP Indirect Channel Select #3
01A0h	TTDP Indirect Channel Select #4

Table 17 TCFP Registers

Register Address	DESCRIPTION
TCFPx_BASE+01h	TCFP Interrupt Indication
TCFPx_BASE+02h	TCFP Idle/Unassigned ATM Cell Header
TCFPx_BASE+03h	TCFP Diagnostics
TCFPx_BASE+04h	TCFP Transmit Cell/Packet Counter (LSB)
TCFPx_BASE+05h	TCFP Transmit Cell/Packet Counter (MSB)
TCFPx_BASE+06h	TCFP Transmit Byte Counter (LSB)
TCFPx_BASE+07h	TCFP Transmit Byte Counter
TCFPx_BASE+08h	TCFP Transmit Byte Counter (MSB)
TCFPx_BASE+09h	TCFP Aborted Packet Counter
x = 1 to 4	

Table 18 TTDP Registers

Register Address	DESCRIPTION
TTDPx_BASE+01h	TTDP Indirect Configuration
TTDPx_BASE+02h	TTDP Indirect Idle/Unassigned ATM Cell Header
TTDPx_BASE+03h	TTDP Indirect Diagnostics
TTDPx_BASE+04h	TTDP Indirect Transmit Cell/Packet Counter (LSB)
TTDPx_BASE+05h	TTDP Indirect Transmit Cell/Packet Counter (MSB)
TTDPx_BASE+06h	TTDP Indirect Transmit Byte Counter (LSB)
TTDPx_BASE+07h	TTDP Indirect Transmit Byte Counter (MSB)
TTDPx_BASE+08h	TTDP Indirect Aborted Packet Counter
TTDPx_BASE+09h	TTDP CRC Error Mask
TTDPx_BASE+0Ah	TTDP Interrupt Enable 1
TTDPx_BASE+0Bh	TTDP Interrupt Enable 2
TTDPx_BASE+0Ch	TTDP Interrupt 1
TTDPx_BASE+0Dh	TTDP Interrupt 2
TTDPx_BASE+0Eh	TTDP Transmit Off
x = 1 to 4	

5.10 SONET/SDH Inband Error Report Processor

The SONET/SDH Inband Error Report Processor (SIRP) maps remote error indications (REI, RDI) from the G1 byte of the receive SONET Path Overhead into the transmit SONET Path Overhead. It is important to note that the SIRP must be enabled for all provisioned channels in order for the S/UNI MACH48 to function correctly. When the SIRP is enabled the J1 byte is passed correctly to the downstream blocks.

Table 19 SIRP Base Addresses

Register Address	DESCRIPTION
14C0h	SIRP1_BASE
14D0h	SIRP2_BASE
14E0h	SIRP3_BASE
14F0h	SIRP4_BASE

Table 20 SIRP Registers

Register Address	DESCRIPTION
SIRPx_BASE + 00h	SIRP Timeslot #0 Configuration
SIRPx_BASE + 01h	SIRP Timeslot #1 Configuration
SIRPx_BASE + 02h	SIRP Timeslot #2 Configuration
SIRPx_BASE + 03h	SIRP Timeslot #3 Configuration
SIRPx_BASE + 04h	SIRP Timeslot #4 Configuration
SIRPx_BASE + 05h	SIRP Timeslot #5 Configuration
SIRPx_BASE + 06h	SIRP Timeslot #6 Configuration
SIRPx_BASE + 07h	SIRP Timeslot #7 Configuration
SIRPx_BASE + 08h	SIRP Timeslot #8 Configuration
SIRPx_BASE + 09h	SIRP Timeslot #9 Configuration
SIRPx_BASE + 0Ah	SIRP Timeslot #A Configuration
SIRPx_BASE + 0Bh	SIRP Timeslot #B Configuration
SIRPx_BASE + 0Ch	SIRP LCD RDI Value Register
x = 1 to 4	

5.11 Receive Channel Assigner (RCAS)

The Receive Channel Assigner groups STS-1/STM-0 timeslots into channels. A more detailed description of its operation can be found in the S/UNI MACH48 Data Sheet, Section 11.13. The RCAS registers are directly accessed and use the following registers.

Table 21 RCAS Base Addresses

Register Address (RCAS12)	DESCRIPTION
01B0h	RCAS_BASE #1
01C0h	RCAS_BASE #2
01D0h	RCAS_BASE #3
01E0h	RCAS_BASE #4

Table 22 RCAS Registers

Register Address	DESCRIPTION
RCAS12+00	RCAS Channel Disable
RCAS12+01	RCAS Channel Loopback Enable
RCAS12+(02-0D)	RCAS Timeslot Configuration #0-11

5.12 Transmit Channel Assigner (TCAS)

The Transmit Channel Assigner maps channel numbers from the UL3/PL3 interface into the STS-1 timeslots. The TCAS registers are directly accessed and use the following registers.

Table 23 TCAS Base Addresses

Register Address	DESCRIPTION
01F0h	TCAS_BASE #1
0200h	TCAS_BASE #2
0210h	TCAS_BASE #3
0220h	TCAS_BASE #4

Table 24 TCAS Registers

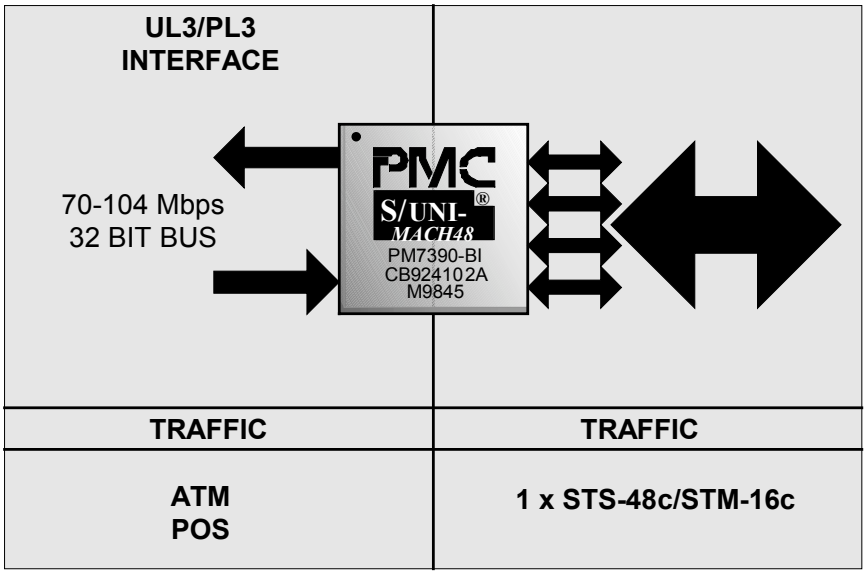
Register Address	DESCRIPTION
TCAS12+00h	TCAS Channel Configuration
TCAS12+(02h-0Dh)	TCAS Timeslot Configuration #0-11

6 EXAMPLES

6.1 Example 1: Single PHY, STS-48c

In this example, the MACH48 device is configured to receive a single PHY STS-48c. The example sets up Serial Telecombus mode and allows the choice of either UL3 or PL3 mode.

Figure 2 - MACH48 STS-48c Example



6.1.1 Set Configuration Register

Table 25 Config Register (STS-48c), S-TCB

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0001h	Master Reset, Configuration and Global Digital Loopback	015Ch	01DCh

Table 26 Config Register (STS-48c), DLOOP

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0001h	Master Reset, Configuration and Global Digital Loopback	015Eh	01DCh

6.1.2 Set Timeslot and Delay Registers (STS-48c)

For STS-48c/STM-16c, the timeslot registers are ignored.

Set the RJ0DLY, this value will need to be modified if the S-TCB is used, depending on the system configuration.

Table 27 RX S-TCB Synchronization Delay

ADDRESS	DESCRIPTION	VALUE
0011h	Receive Serial Telecombus Syncronization Delay	807Fh

Set the IWTI, IPTI,OPTI and OWTI modes in the Miscellaneous Register to STS-48c mode.

Table 28 Miscellaneous Register 0012h

ADDRESS	DESCRIPTION	VALUE
0012h	Miscellaneous	80FAh

Note: This is the default value at startup

6.1.3 Set Serial Telecombus Registers

For Serial Telecombus Mode, set up the working and protect links for High Order Path Termination (HPT) mode.

Set TMODEx[1..0] => 01 for HPT mode.

Table 29 S-TCB Register Setup

ADDRESS	DESCRIPTION	VALUE
T8TE1_BASE (1860h) + 0002h	T8TE1 Telecombus Mode #1	5555h
T8TE1_BASE (1860h) + 0003h	T8TE1 Telecombus Mode #2	0055h
T8TE2_BASE (1868h) + 0002h	T8TE2 Telecombus Mode #1	5555h
T8TE2_BASE (1868h) + 0003h	T8TE2 Telecombus Mode #2	0055h
T8TE3_BASE (1870h) + 0002h	T8TE3 Telecombus Mode #1	555h
T8TE3_BASE (1870h) + 0003h	T8TE3 Telecombus Mode #2	0055h
T8TE4_BASE (1878h) + 0002h	T8TE4 Telecombus Mode #1	5555h
T8TE4_BASE (1878h) + 0003h	T8TE4 Telecombus Mode #2	0055h
T8TE5_BASE (1880h) + 0002h	T8TE5 Telecombus Mode #1	5555h
T8TE5_BASE (1880h) + 0003h	T8TE5 Telecombus Mode #2	0055h
T8TE6_BASE (1888h) + 0002h	T8TE6 Telecombus Mode #1	5555h
T8TE6_BASE (1888h) + 0003h	T8TE6 Telecombus Mode #2	0055h
T8TE7_BASE (1890h) + 0002h	T8TE7 Telecombus Mode #1	5555h
T8TE7_BASE (1890h) + 0003h	T8TE7 Telecombus Mode #2	0055h
T8TE8_BASE (1898h) + 0002h	T8TE8 Telecombus Mode #1	5555h
T8TE8_BASE (1898h) + 0003h	T8TE8 Telecombus Mode #2	0055h

The following register must also be set in the S/UNI MACH48 for proper operation.

Table 30 R8TDx Analog Control Setting

ADDRESS	DESCRIPTION	VALUE
R8TDx_BASE+03h	R8TDx Analog Control 1	CC34h
(x=1 to 8)		

For loopback (Serial Telecombus Diagnostic loopback), the loopback bits must be set. The DLBEN bits are located in the R8TD blocks.

Table 31 S-TCB Diagnostic Loopback

ADDRESS	DESCRIPTION	VALUE
R8TD1_BASE (1820h)	R8TD1 Control and Status	8000h
R8TD2_BASE (1828h)	R8TD2 Control and Status	8000h
R8TD3_BASE (1830h)	R8TD3 Control and Status	8000h

ADDRESS	DESCRIPTION	VALUE
R8TD4_BASE (1838h)	R8TD4 Control and Status	8000h
R8TD5_BASE (1840h)	R8TD5 Control and Status	8000h
R8TD6_BASE (1848h)	R8TD6 Control and Status	8000h
R8TD7_BASE (1850h)	R8TD7 Control and Status	8000h
R8TD8_BASE (1858h)	R8TD8 Control and Status	8000h

6.1.4 RXPHY Calendar (STS-48c/STM-16c)

For the STS-48c example, there is only one entry in the calendar when the POS mode is selected. The calendar length requires a direct write while the calendar entry requires an indirect register write.

Table 32 Calendar Length Value (1 x STS-48c)

ADDRESS	DESCRIPTION	VALUE
0044h	RXPHY Calendar Length	0000h

An example write for a calendar entry is illustrated below:

1. Poll the BUSY bit in register 0045h: Bit 15.

2. When BUSY bit is 0, write 0 to CALENDAR ADDRESS[6..0] and CALENDAR DATA[5..0] and CONFIG_RWB in register 0045h.

The TXPHY is set up in a similar manner.

6.1.5 Scalable Data Queue (STS48c/STM-16c)

Both the RXSDQ and the TXSDQ are set up in a similar manner. For the STS-48c example there is only one entry in the SDQ. The entry will have the following attributes.

Table 33 Example SDQ Entry Setup for PHYID 0(STS-48c)

Description	Value(DEC)	Value (BIN)	Register	Note
PHYID	0	000000b	0058h	PHYID for the STS-48c channel
FIFO NUMBER	0	000000b	0059h	Associates a FIFO# with a PHYID
FIFO BLOCK SIZE	3	11b	0059h	A Block size of 192 (as per Data Sheet Section 14)

Description	Value(DEC)	Value (BIN)	Register	Note
DATA AVAILABLE THRESHOLD	3 or 95	00000011b or 01011111b	005A	= 3 for ATM and variable for POS depending on required burst size, in this example=95.

These values are written to the RXSDQ in the following order:

1. Poll the BUSY bit in register 0058h until it reads 0.
2. Write the data to the addresses as shown in Table 34.
3. Write the Indirect Register Address with the RWB bit set to 0 and the PHYID. This will initiate the write of the indirect data registers.

Writing to the TXSDQ is done in the same manner.

Table 34 SDQ Register Setup (STS-48c)

ADDRESS	DESCRIPTION	ATM	POS
0059h	RXSDQ FIFO Indirect Configuration	80C0h	C0C0h
005Ah	RXSDQ FIFO Indirect Data AvailableThreshold	0303h	0703h
0069h	TXSDQ FIFO Indirect Configuration	80C0h	C0C0h
006Ah	TXSDQ FIFO Inidrect Data/Buffer Available Threshold	0303h	5F07h

6.1.6 Receive Cell and Frame Processor (STS48c)

In STS-48c/STM-4c mode, only the least significant RCFP base address is used (0070h). The RTDP blocks are not used. Note, at this point the processor is not provisioned. This will be done after all other registers are set.

Table 35 RCFP0 Register (STS-48c)

ADDRESS	DESCRIPTION	ATM	POS
0070h	RCFP Configuration Register #1	0110h	0510h

6.1.7 Transmit Cell and Frame Processor (STS48c)

The TCFP is set up identically to the RCFP.

Table 36 TCFP0 Register (STS-48c)

ADDRESS	DESCRIPTION	ATM	POS
00B0h	TCFP Configuration Register #1	0080h	0180h

6.1.8 Receive Channel Assigner (STS48c)

Set the Timeslot mode and provision each channel (0 to 11). The data sheet stipulates that if the MACH48 is set up for STS-48c, all timeslots should be provisioned and enabled for channel 0.

For this example: TsxMode = 000 for STS-12c/STM-4c

TSx_PROV is set to 1

The registers should be written as follows:

Table 37 RCAS Register Setup (STS-48c)

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE (0x01B0h) + (0002h+x)	RCAS0 Timeslot Configuration #0-11	0010h
RCAS1_BASE (0x01C0h) + (0002h+x)	RCAS1 Timeslot Configuration #0-11	0010h
RCAS2_BASE (0x01D0h) + (0002h+x)	RCAS2 Timeslot Configuration #0-11	0010h
RCAS3_BASE (0x01E0h) + (0002h+x)	RCAS3 Timeslot Configuration #0-11	0010h
x = 0h to Bh		

If channel based diagnostic loopback is required, the following registers should be written:

Table 38 RCAS Loopback (STS-48c)

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE (0x01B0h) + (0001h)	RCAS0 Channel Loopback Enable	1001h
RCAS1_BASE (0x01C0h) + (0001h)	RCAS1 Channel Loopback Enable	1001h
RCAS2_BASE (0x01D0h) + (0001h)	RCAS2 Channel Loopback Enable	1001h
RCAS3_BASE (0x01E0h) + (0001h)	RCAS3 Channel Loopback Enable	1001h

6.1.9 Transmit Channel Assigner (STS48c/STM-16c)

Set the Timeslot mode and provision each channel (0 to 11). Like the RCAS, all timeslots should be provisioned and enabled for channel 0.

For this example: TsxMode = 000 for STS-12c/STM-4c

TSx_PROV is set to 1

The registers should be written as follows:

Table 39 TCAS Register Setup (STS-48c)

ADDRESS	DESCRIPTION	VALUE
TCAS0_BASE (0x01F0h) + (0002h+x) where x = 0 to Bh	TCAS0 Timeslot Configuration #0-11	0010h
TCAS1_BASE (0x0200h) + (0002h+x) where x = 0 to Bh	TCAS1 Timeslot Configuration #0-11	0010h
TCAS2_BASE (0x0210h) + (0002h+x) where x = 0 to Bh	TCAS2 Timeslot Configuration #0-11	0010h
TCAS3_BASE (0x0220h) + (0002h+x) where x = 0 to Bh	TCAS3 Timeslot Configuration #0-11	0010h

6.1.10 SONET/SDH Inband Error Report Processor

For the STS-48c example, only the SIRP1_BASE is provisioned.

Table 40 STS-48c SIRP Setup

ADDRESS	DESCRIPTION	VALUE
SIRP1_BASE+00h	SIRP Timeslot Configuration #0	0007h

6.1.11 Enable Blocks (STS48c/STM-16c)

Once all of the registers are set up as required, the blocks can be enabled.

Table 41 Provisioning RCFP/TCFP Blocks

ADDRESS	DESCRIPTION	ATM	POS
0070h	RCFP Configuration Register	0111h	0511h

00B0h	TCFP Configuration Register	0081h	0181h

In the STS-48c mode, all of the channels are provisioned and enabled.

Table 42 Enabling RCAS/TCAS Blocks

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE	RCAS12 Channel Disable	0000h
RCAS1_BASE	RCAS12 Channel Disable	0000h
RCAS2_BASE	RCAS12 Channel Disable	0000h
RCAS3_BASE	RCAS12 Channel Disable	0000h
TCAS0_BASE	TCAS12 Channel Configuration	0000h
TCAS1_BASE	TCAS12 Channel Configuration	0000h
TCAS2_BASE	TCAS12 Channel Configuration	0000h
TCAS3_BASE	TCAS12 Channel Configuration	0000h

The SDQ FIFO are enable as follows:

Table 43 Enabling RXSDQ/TXSDQ Blocks

ADDRESS	DESCRIPTION	VALUE
0050h	RXSDQ FIFO Reset	0000h
0060h	TXSDQ FIFO Reset	0000h

The RXPHY and TXPHY is enabled. For the STS-48c mode, TPAHOLD is set to 0 since the FIFO size is large (192 blocks).

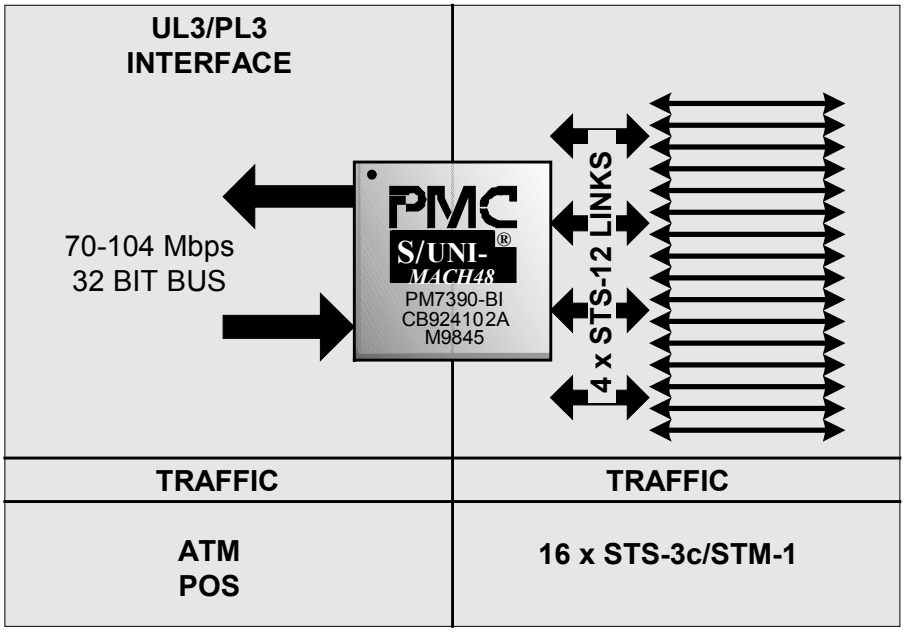
Table 44 Enabling RXPHY/TXPHY Blocks

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0040h	RXPHY Configuration	0000h	0000h
0048h	TXPHY Configuration	0000h	0040h

6.2 Example 2: 16 x STS-3c Channels

In this example, the MACH48 is setup up to have 16 STS-3c PHYs. The PHYS are numbered from 0 to 3, 12 to 15, 24 to 27, and 36 to 39.

Figure 3 - MACH48 STS-3c Example



6.2.1 Set Configuration Register

The setup is for Serial Telecombus and either UL3 or PL3

Table 45 Config Register (16 x STS-3c), S-TCB

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0001h	Master Reset, Configuration and Global Digital Loopback	0150h	01D0h

For Diagnostic loopback, the register is set up as follows:

Table 46 Config Register (16 x STS-3c), DLOOP

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0001h	Master Reset, Configuration and Global Digital Loopback	0152h	01D2h

6.2.2 Set Timeslot and Delay Registers (16 x STS-3c)

Table 47 Timeslot Registers (16 x STS-3c)

ADDRESS	DESCRIPTION	VALUE
0002h-000Dh	RX/TX Timeslot Configuration #0-11	5555h

Set the RJ0DLY, this value will need to be modified depending on the system configuration.

Table 48 RX S-TCB Synchronization Delay

ADDRESS	DESCRIPTION	VALUE
0011h	Receive Serial Telecombus Synchronization Delay	007Fh

Set the IWTI, IPTI, OPTI and OWTI modes in the Miscellaneous Register to bypass.

Table 49 Miscellaneous Register 0012h

ADDRESS	DESCRIPTION	VALUE
0012h	Miscellaneous	8055h

6.2.3 Set Serial Telecombus Registers (STS-3c)

Similarly to Example 1, the MACH48 is set up in Serial Telecombus Mode. Set up the working and protect links for High Order Path Termination (HPT) mode. The R8TDx +03h must also be set to CC34h.

Set TMODEx[1..0] => 01 for HPT mode.

6.2.4 Set Up RXPHY Calendar (16 x STS-3c)

For this example, there are 48 entries in the calendar when POS mode is selected.

Table 50 Calendar Register Values (16 x STS-3c)

ADDRESS	DESCRIPTION	VALUE
0044h	RXPHY Calendar Length	002Fh

Table 51 RXPHY Calendar (16 x STS-3c)

Calendar Entry	Channel #	Calendar Entry	Channel #	Calendar Entry	Channel #
0	0	**16**	0	**32**	0
1	12	**17**	12	**33**	12
2	24	**18**	24	**34**	24
3	36	**19**	36	**35**	36
4	1	**20**	1	**36**	1
5	13	**21**	13	**37**	13
6	25	**22**	25	**38**	25
7	37	**23**	37	**39**	37
8	2	**24**	2	**40**	2
9	14	**25**	14	**41**	14
10	26	**26**	26	**42**	26
11	38	**27**	38	**43**	38
12	3	**28**	3	**44**	3
13	15	**29**	15	**45**	15
14	27	**30**	27	**46**	27
15	39	**31**	39	**47**	39

An example write to calendar entry 15 is illustrated below:

1. Poll the BUSY bit in register 0045h: Bit 15.

2. When BUSY bit is 0, write 15 to CALENDAR ADDRESS[6..0] and 39 to CALENDAR DATA[5..0] and CONFIG_RWB in register 0045h. (0F27h)

6.2.5 Scalable Data Queue (16 x STS-3c)

Both the RXSDQ and TXSDQ should be set up according to the following table.

Table 52 FIFO Setup (16 x STS-3c)

Channel # PHYID[5:0]	BW	Size (Block)	Size (Cell)	FIFO_BS [1:0]	Bank	FIFO# [5:0]	Block_PTR [4:0]
							(hex)
0	STS-3c	48	12	2	0	0	00
1						1	06
2						2	0C
3						3	12
12	STS-3c	48	12	2	1	16	00
13						17	06
14						18	0C
15						19	12
24	STS-3c	48	12	2	2	32	00
25						33	06
26						34	0C
27						35	12
36	STS-3c	48	12	2	3	48	00
37						49	06
38						50	0C
39						51	12

The following is an example of how an SDQ entry is performed. The table below shows the required parameters used to set up the PHYID = 14 entry in Table 52.

Table 53 Example SDQ Entry Setup for PHYID 14(16 x STS-3c)

Description	Value(DEC)	Value (BIN)	Register	Note
PHYID	14	00000Eb	0058h	PHYID for the STS-3c channel
FIFO NUMBER	17	010001b	0059h	Associates a FIFO# with a PHYID
FIFO BLOCK SIZE	2	10b	0059h	A Block size of 48 (as per Data Sheet Section 14)
DATA AVAILABLE THRESHOLD	3 or 23	00000011b or 00100011b	005A	= 3 for ATM and variable for POS depending on required burst size, in this example=23.

These values are written to the RXSDQ in the following order:

1. Poll the BUSY bit in register 0058h until it reads 0.
2. Write the data to the addresses as shown in Table 54.
3. Write the Indirect Register Address with the RWB bit set to 0 and the PHYID. This will initiate the write of the indirect data registers.

Writing to the TXSDQ is done in the same manner.

Table 54 SDQ Register Setup for PHYID 14 STS-3c Entry

ADDRESS	DESCRIPTION	ATM VALUE	POS VALUE
0059h	RXSDQ FIFO Indirect Configuration	928Ch	D28Ch
005Ah	RXSDQ FIFO Indirect Data AvailableThreshold	0303h	0703h
0069h	TXSDQ FIFO Indirect Configuration	928Ch	D28Ch
006Ah	TXSDQ FIFO Inidrect Data Available Threshold	0303h	2307h

6.2.6 Receive Timeslice Datacom Processor

In this example (16 x STS-3c), the RCFP blocks are not used. The table below shows the values that are to be written into the RTDP indirect registers for each of the 16 channels used.

RTDP_BASE + 01h: Enable Channel, Select POS/ATM, STRIP_SEL = 1.

RTDP_BASE + (02h – 05h) are set to default values.

Table 55 RTDP Register Setup (16 x STS-3c)

Channel # PHYID[5:0]	RTDP BASE	OFFSET					
		01		02	03	04	05
		POS	ATM				
0	**00F0h**	0511	0111	0402	0300	0168	01FF
1							
2							
3							
12	**0110h**	0511	0111	0402	0300	0168	01FF
13							
14							
15							
24	**0130h**	0511	0111	0402	0300	0168	01FF
25							
26							
27							

Channel # PHYID[5:0]	RTDP BASE	OFFSET					
		01		02	03	04	05
36	0150h	0511h	0111h	0402h	0300h	0168h	01FFh
37							
38							
39							

The example below will write the RTDP setup for PHYID = 27.

1. Poll the BUSY bit in register: 0130h until it reads 0.

2. Write the above indirect registers except the Indirect Address Register.

3. Write the Indirect Register Address (0130h) last with the RWB bit set to 0 and the CHAN[3..0] set to 3 (0003h). This will initiate the write to Channel 3 which corresponds to PHYID.= 27 in the third RTDP block.

6.2.7 Transmit Timeslice Datacom Processor

Similarly to the RTDP, the TTDP is set up as follows.

Table 56 TTDP Register Setup (16 x STS-3c)

Channel # PHYID[5:0]	TTDP BASE	OFFSET			
		01h		02h	03h
		POS	ATM		
0	0170	0301	0101	016A	0003
1					
2					
3					
12	0180	0301	0101	016A	0003
13					
14					
15					
24	0190	0301	0101	016A	0003
25					
26					

Channel # PHYID[5:0]	TTDP BASE	OFFSET			
		01h		02h	03h
		POS	ATM		
27					
36	01A0	0301	0101	016A	0003
37					
38					
39					

The following is an example of how an TTDP entry is performed. The table below shows the required register values to set up the PHYID = 26 entry in Table 56.

Table 57 TTDP Register Setup for PHYID 26 STS-3c Entry

ADDRESS	DESCRIPTION	ATM VALUE	POS VALUE
TTDP2_BASE + 00h	TTDP Indirect Channel Select	0002h	0002h
TTDP2_BASE + 01h	TTDP Indirect Configuration	0101h	0301h
TTDP2_BASE + 02h	TTDP Indirect Idle/Unassigned ATM Cell Header	016Ah	016Ah
TTDP2_BASE + 03h	TTDP Indirect Diagnostics	0003h	0003h

1. Poll the BUSY bit in register: 0190h until it reads 0.

2. Write the above indirect registers except the Indirect Address Register.

3. Write the Indirect Register Address (0190h) last with the RWB bit set to 0 and the CHAN[3..0] set to 2 (0002h). This will initiate the write to Channel 2 which corresponds to PHYID= 26 in the third RTDP block.

6.2.8 Receive Channel Assigner (16 x STS-3c)

For this example: TsxMode = 001 for STS-3c/STM-1

TSx_PROV is set to 1

The registers should be written as follows to each RCAS base address.

Table 58 RCAS Register Setup (16 x STS-3c)

ADDRESS	DESCRIPTION	VALUE
RCAS_BASE + 0002h	RCAS Timeslot Configuration #0	0030h
RCAS_BASE + 0003h	RCAS Timeslot Configuration #1	0031h
RCAS_BASE + 0004h	RCAS Timeslot Configuration #2	0032h
RCAS_BASE + 0005h	RCAS Timeslot Configuration #3	0033h
RCAS_BASE + 0006h	RCAS Timeslot Configuration #4	0030h
RCAS_BASE + 0007h	RCAS Timeslot Configuration #5	0031h
RCAS_BASE + 0008h	RCAS Timeslot Configuration #6	0032h
RCAS_BASE + 0009h	RCAS Timeslot Configuration #7	0033h
RCAS_BASE + 000Ah	RCAS Timeslot Configuration #8	0030h
RCAS_BASE + 000Bh	RCAS Timeslot Configuration #9	0031h
RCAS_BASE + 000Ch	RCAS Timeslot Configuration #10	0032h
RCAS_BASE + 000Dh	RCAS Timeslot Configuration #11	0033h

If channel based diagnostic loopback is required, the following registers should be written:

Table 59 RCAS Loopback Settings (16 x STS-3c)

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE (0x01B0h) + (0001h)	RCAS0 Channel Loopback Enable	000Fh
RCAS1_BASE (0x01C0h) + (0001h)	RCAS1 Channel Loopback Enable	000Fh
RCAS2_BASE (0x01D0h) + (0001h)	RCAS2 Channel Loopback Enable	000Fh
RCAS3_BASE (0x01E0h) + (0001h)	RCAS3 Channel Loopback Enable	000Fh

6.2.9 Transmit Channel Assigner (16 x STS-3c)

Set the Timeslot mode and provision each channel (0 to 11).

TsxMode = 001 for STS-3c/STM-1,TSx_PROV is set to 1

Therefore the registers should be written as follows:

Table 60 TCAS Configuration Registers (16 x STS-3c)

ADDRESS	DESCRIPTION	VALUE
TCAS_BASE + 0002h	TCAS Timeslot Configuration #0	0030h

ADDRESS	DESCRIPTION	VALUE
TCAS_BASE + 0003h	TCAS Timeslot Configuration #1	0031h
TCAS_BASE + 0004h	TCAS Timeslot Configuration #2	0032h
TCAS_BASE + 0005h	TCAS Timeslot Configuration #3	0033h
TCAS_BASE + 0006h	TCAS Timeslot Configuration #4	0030h
TCAS_BASE + 0007h	TCAS Timeslot Configuration #5	0031h
TCAS_BASE + 0008h	TCAS Timeslot Configuration #6	0032h
TCAS_BASE + 0009h	TCAS Timeslot Configuration #7	0033h
TCAS_BASE + 000Ah	TCAS Timeslot Configuration #8	0030h
TCAS_BASE + 000Bh	TCAS Timeslot Configuration #9	0031h
TCAS_BASE + 000Ch	TCAS Timeslot Configuration #10	0032h
TCAS_BASE + 000Dh	TCAS Timeslot Configuration #11	0033h

6.2.10 SONET/SDH Inband Error Processor (16 x STS-3c)

For the 16 x STS-3c example, only the first four timeslots of each SIRP are provisioned.

Table 61 - 16 x STS-3c SIRP Configuration

ADDRESS	DESCRIPTION	VALUE
SIRPx_BASE + 00h	SIRP Timeslot #0 Configuration	0001h
SIRPx_BASE + 01h	SIRP Timeslot #1 Configuration	0001h
SIRPx_BASE + 02h	SIRP Timeslot #2 Configuration	0001h
SIRPx_BASE + 03h	SIRP Timeslot #3 Configuration	0001h
x= 1 to 4		

6.2.11 Enable Blocks (16 x STS-3c)

Once all of the registers are set up as required, the blocks can be enabled.

Table 62 Enabling RCAS/TCAS Blocks

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE	RCAS12 Channel Disable	0FF0h
RCAS1_BASE	RCAS12 Channel Disable	0FF0h
RCAS2_BASE	RCAS12 Channel Disable	0FF0h
RCAS3_BASE	RCAS12 Channel Disable	0FF0h
TCAS0_BASE	TCAS12 Channel Configuration	0FF0h
TCAS1_BASE	TCAS12 Channel Configuration	0FF0h
TCAS2_BASE	TCAS12 Channel Configuration	0FF0h
TCAS3_BASE	TCAS12 Channel Configuration	0FF0h

Table 63 Enabling RXSDQ/TXSDQ Blocks

ADDRESS	DESCRIPTION	VALUE
0050h	RXSDQ FIFO Reset	0000h
0060h	TXSDQ FIFO Reset	0000h

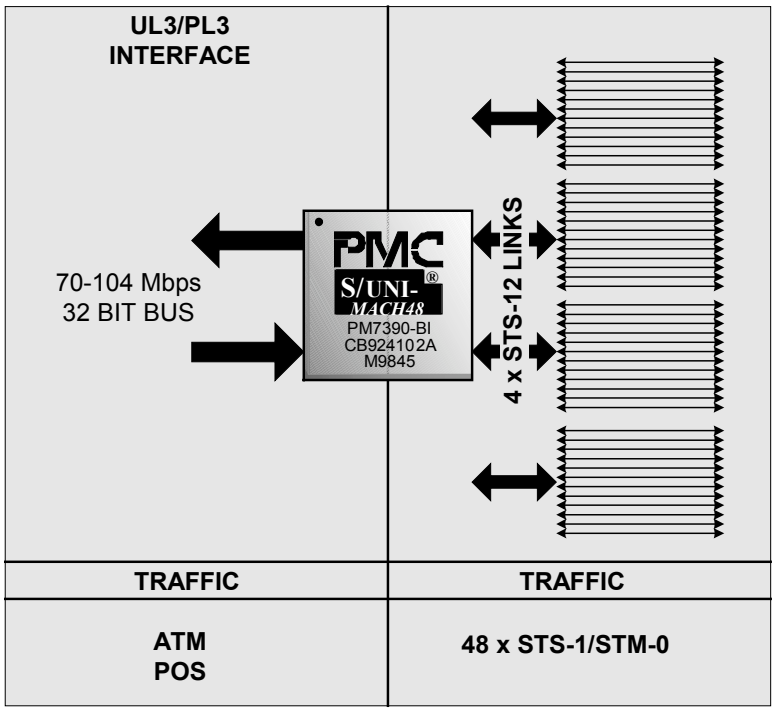
Table 64 Enabling RXPHY/TXPHY Blocks

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0040h	RXPHY Configuration	0000h	0000h
0048h	TXPHY Configuration	0000h	0040h

6.3 Example 3: 48 x STS-1

In this example, the MACH48 is setup up to have 48 PHYs, each at STS-1/STM-0 bandwidth. The PHYS are numbered from 0 to 47.

Figure 4 - MACH48 STS-1 Example



6.3.1 Set Configuration Register

The setup is for Serial Telecombus and either UL3 or PL3

Table 65 Config Register (48 x STS-1), S-TCB

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0001h	Master Reset, Configuration and Global Digital Loopback	0150h	01D0h

For Diagnostic loopback, the register is set up as follows:

Table 66 Config Register (48 x STS-1), DLOOP

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0001h	Master Reset, Configuration and Global Digital Loopback	0152h	01D2h

6.3.2 Set Timeslot and Delay Registers (48 x STS-1)

Both the RX and TX Timeslot configuration registers are written with the same values.

Table 67 Timeslot Registers (48 x STS-1)

ADDRESS	DESCRIPTION	VALUE
0002h-000Dh	RX/TX Timeslot Configuration	AAAAh

Set the RJ0DLY, this value will need to be modified depending on the system configuration.

Table 68 RX S-TCB Synchronization Delay

ADDRESS	DESCRIPTION	VALUE
0011h	Receive Serial Telcombus Synchronization Delay	007Fh

Set the IWTI, IPTI,OPTI and OWTI modes in the Miscellaneous Register bypass mode.

Table 69 Miscellaneous Register 0012h

ADDRESS	DESCRIPTION	VALUE
0012h	Miscellaneous	8055h

6.3.3 Set Serial Telecombus Registers

Similarly to Example 1, Section 6.1.3 the MACH48 is set up in Serial Telecombus Mode. Set up the working and protect links for High Order Path Termination (HPT) mode. The R8TDx +03h must also be set to CC34h.

Set TMODEx[1..0] => 01 for HPT mode.

6.3.4 Set Up RXPHY Calendar

The Calendar for this example has a length of 48. Each channel from 0 to 47 is polled with a uniform probability of being selected.

Table 70 Calendar Length Register (48 x STS-1)

ADDRESS	DESCRIPTION	VALUE
0044h	RXPHY Calendar Length	002Fh

The table can be set up as follows:

Table 71 - RXPHY Calendar Setup (48 x STS-1)

Calendar Entry	Channel #	Calendar Entry	Channel #	Calendar Entry	Channel #
0	0	**16**	4	**32**	8
1	12	**17**	16	**33**	20
2	24	**18**	28	**34**	32
3	36	**19**	40	**35**	44
4	1	**20**	5	**36**	9
5	13	**21**	17	**37**	21
6	25	**22**	29	**38**	33
7	37	**23**	41	**39**	45
8	2	**24**	6	**40**	10
9	14	**25**	18	**41**	22
10	26	**26**	30	**42**	34
11	38	**27**	42	**43**	46
12	3	**28**	7	**44**	11
13	15	**29**	19	**45**	23

Calendar Entry	Channel #	Calendar Entry	Channel #	Calendar Entry	Channel #
14	27	**30**	31	**46**	35
15	39	**31**	43	**47**	47

6.3.5 Scalable Data Queue (48 x STS-1)

Both the RXSDQ and TXSDQ are set up according to the following table. The block pointers are arranged such that any reconfiguration to a higher bandwidth channel will not affect unintended channels.

Table 72 48 x STS-1 FIFO Setup

Channel # PHYID[5:0]	BW	Size (Block)	Size (Cell)	FIFO_BS [1:0]	Bank	FIFO# [5:0]	Block_PTR [4:0]
0	STS-1	16	4	1	0	0	00
1						1	06
2						2	0C
3						3	12
4						4	02
5						5	08
6						6	0E
7						7	14
8						8	04
9						9	0A
10						10	10
11						11	16
12	STS-1	16	4	1	1	16	00
13						17	06
14						18	0C
15						19	12
16						20	02
17						21	08
18						22	0E
19	STS-1	16	4	1	1	23	14

Channel # PHYID[5:0]	BW	Size (Block)	Size (Cell)	FIFO_BS [1:0]	Bank	FIFO# [5:0]	Block_PTR [4:0]
20						24	04
21						25	0A
22						26	10
23						27	16
24	STS-1	16	4	1	2	32	00
25						33	06
26						34	0C
27						35	12
28						36	02
29						37	08
30						38	0E
31						39	14
32						40	04
33						41	0A
34						42	10
35						43	16
36	STS-1	16	4	1	3	48	00
37						49	06
38						50	0C
39						51	12
40						52	02
41						53	08
42						54	0E
43						55	14
44						56	04
45						57	0A
46						58	10
47						59	16

The table below shows the required parameters used to set up the PHYID = 36 entry in Table 72.

Table 73 - Example SDQ Entry Setup for PHYID 36(48 x STS-1)

Description	Value(DEC)	Value (BIN)	Register	Note
PHYID	36	100100b	0058h	PHYID for the STS-1 channel
FIFO NUMBER	48	110000b	0059h	Associates a FIFO# with a PHYID
FIFO BLOCK SIZE	1	01b	0059h	A Block size of 48 (as per Data Sheet Section 14)
DATA AVAILABLE THRESHOLD	3 or 7	00000011b or 00000111b	005A	= 3 for ATM and variable for POS depending on required burst size, in this example=7.

These values are written to the RXSDQ in the following order:

1. Poll the BUSY bit in register 0058h until it reads 0.
2. Write the data to the addresses as shown in Table 74.
3. Write the Indirect Register Address with the RWB bit set to 0 and the PHYID. This will initiate the write of the indirect data registers.

Writing to the TXSDQ is done in the same manner.

Table 74 SDQ Register Setup for PHYID 36 STS-1 Entry

ADDRESS	DESCRIPTION	ATM VALUE	POS VALUE
0059h	RXSDQ FIFO Indirect Configuration	B040h	F040h
005Ah	RXSDQ FIFO Indirect Data AvailableThreshold	0303h	0703h
0069h	TXSDQ FIFO Indirect Configuration	B040h	F040h
006Ah	TXSDQ FIFO Inidrect Data Available Threshold	0303h	0707h

6.3.6 Receive Timeslice Datacom Processor

The RTDP is set up according to the following table.

Table 75 RTDP Register Setup (48 x STS-1)

Channel #	RTDP	OFFSET					
PHYID[5:0]	BASE	01		02	03	04	05
		POS	ATM				
0	00F0h	0511h	0111h	0402h	0300h	0168h	01FFh
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12	0110h	0511h	0111h	0402h	0300h	0168h	01FFh
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23	0130h	0511h	0111h	0402h	0300h	0168h	01FFh
24							
25							

Channel #	RTDP	OFFSET					
PHYID[5:0]	BASE	01		02	03	04	05
		POS	ATM				
26	0130h	0511h	0111h	0402h	0300h	0168h	01FFh
27							
28							
29							
30							
31							
32							
33							
34							
35	0150H	0511h	0111h	0402h	0300	0168h	01FFh
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							

6.3.7 Transmit Timeslice Datacom Processor

The TTDP is set up identically to the RTDP above.

6.3.8 Receive Channel Assigner

TSxMode = 010 for STS-1/STM-0, TSx_PROV is set to 1

The registers should be written as follows for each RCAS block:

Table 76 RCAS Register Setup (48 x STS-1)

ADDRESS	DESCRIPTION	VALUE
RCAS_BASE + 0002h	RCAS Timeslot Configuration #0	0050h
RCAS_BASE + 0003h	RCAS Timeslot Configuration #1	0051h
RCAS_BASE + 0004h	RCAS Timeslot Configuration #2	0052h
RCAS_BASE + 0005h	RCAS Timeslot Configuration #3	0053h
RCAS_BASE + 0006h	RCAS Timeslot Configuration #4	0054h
RCAS_BASE + 0007h	RCAS Timeslot Configuration #5	0055h
RCAS_BASE + 0008h	RCAS Timeslot Configuration #6	0056h
RCAS_BASE + 0009h	RCAS Timeslot Configuration #7	0057h
RCAS_BASE + 000Ah	RCAS Timeslot Configuration #8	0058h
RCAS_BASE + 000Bh	RCAS Timeslot Configuration #9	0059h
RCAS_BASE + 000Ch	RCAS Timeslot Configuration #10	005Ah
RCAS_BASE + 000Dh	RCAS Timeslot Configuration #11	005Bh

If channel based diagnostic loopback is required, the following registers should be written:

Table 77 RCAS Base Register Values for Loopback(48 x STS-1)

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE (0x01B0h) + (0001h)	RCAS0 Channel Loopback Enable	0FFFh
RCAS1_BASE (0x01C0h) + (0001h)	RCAS1 Channel Loopback Enable	0FFFh
RCAS2_BASE (0x01D0h) + (0001h)	RCAS2 Channel Loopback Enable	0FFFh
RCAS3_BASE (0x01E0h) + (0001h)	RCAS3 Channel Loopback Enable	0FFFh

6.3.9 Transmit Channel Assigner

Set the Timeslot mode and provision each channel.

TSxMode = 010 for STS-1/STM-0, TSx_PROV is set to 1

The registers are written as follows:

Table 78 TCAS Configuration Registers (48 x STS-1)

ADDRESS	DESCRIPTION	VALUE
TCAS_BASE + 0002h	TCAS Timeslot Configuration #0	0050h
TCAS_BASE + 0003h	TCAS Timeslot Configuration #1	0051h
TCAS_BASE + 0004h	TCAS Timeslot Configuration #2	0052h
TCAS_BASE + 0005h	TCAS Timeslot Configuration #3	0053h
TCAS_BASE + 0006h	TCAS Timeslot Configuration #4	0054h
TCAS_BASE + 0007h	TCAS Timeslot Configuration #5	0055h
TCAS_BASE + 0008h	TCAS Timeslot Configuration #6	0056h
TCAS_BASE + 0009h	TCAS Timeslot Configuration #7	0057h
TCAS_BASE + 000Ah	TCAS Timeslot Configuration #8	0058h
TCAS_BASE + 000Bh	TCAS Timeslot Configuration #9	0059h
TCAS_BASE + 000Ch	TCAS Timeslot Configuration #10	005Ah
TCAS_BASE + 000Dh	TCAS Timeslot Configuration #11	005Bh

6.3.10 SONET/SDH Inband Error Processor (48 x STS-1)

For the 48 x STS-1 example, all 48 timeslots are provisioned.

Table 79 - 48 x STS-1 SIRP Configuration

ADDRESS	DESCRIPTION	VALUE
SIRPx_BASE + 00h	SIRP Timeslot #0 Configuration	0007h
SIRPx_BASE + 01h	SIRP Timeslot #1 Configuration	0007h
SIRPx_BASE + 02h	SIRP Timeslot #2 Configuration	0007h
SIRPx_BASE + 03h	SIRP Timeslot #3 Configuration	0007h
SIRPx_BASE + 04h	SIRP Timeslot #4 Configuration	0007h
SIRPx_BASE + 05h	SIRP Timeslot #5 Configuration	0007h
SIRPx_BASE + 06h	SIRP Timeslot #6 Configuration	0007h
SIRPx_BASE + 07h	SIRP Timeslot #7 Configuration	0007h
SIRPx_BASE + 08h	SIRP Timeslot #8 Configuration	0007h
SIRPx_BASE + 09h	SIRP Timeslot #9 Configuration	0007h
SIRPx_BASE + 0Ah	SIRP Timeslot #A Configuration	0007h
SIRPx_BASE + 0Bh	SIRP Timeslot #B Configuration	0007h
x= 1 to 4		

6.3.11 Enable Blocks (48 x STS-1)

Once all of the registers are set up as required, the blocks can be enabled.

Table 80 Enabling RCAS/TCAS Blocks

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE	RCAS12 Channel Disable	0000h
RCAS1_BASE	RCAS12 Channel Disable	0000h
RCAS2_BASE	RCAS12 Channel Disable	0000h
RCAS3_BASE	RCAS12 Channel Disable	0000h
TCAS0_BASE	TCAS12 Channel Configuration	0000h
TCAS1_BASE	TCAS12 Channel Configuration	0000h
TCAS2_BASE	TCAS12 Channel Configuration	0000h
TCAS3_BASE	TCAS12 Channel Configuration	0000h

Table 81 Enabling RXSDQ/TXSDQ Blocks

ADDRESS	DESCRIPTION	VALUE
0050h	RXSDQ FIFO Reset	0000h
0060h	TXSDQ FIFO Reset	0000h

For this example, TPAHOLD = 1.

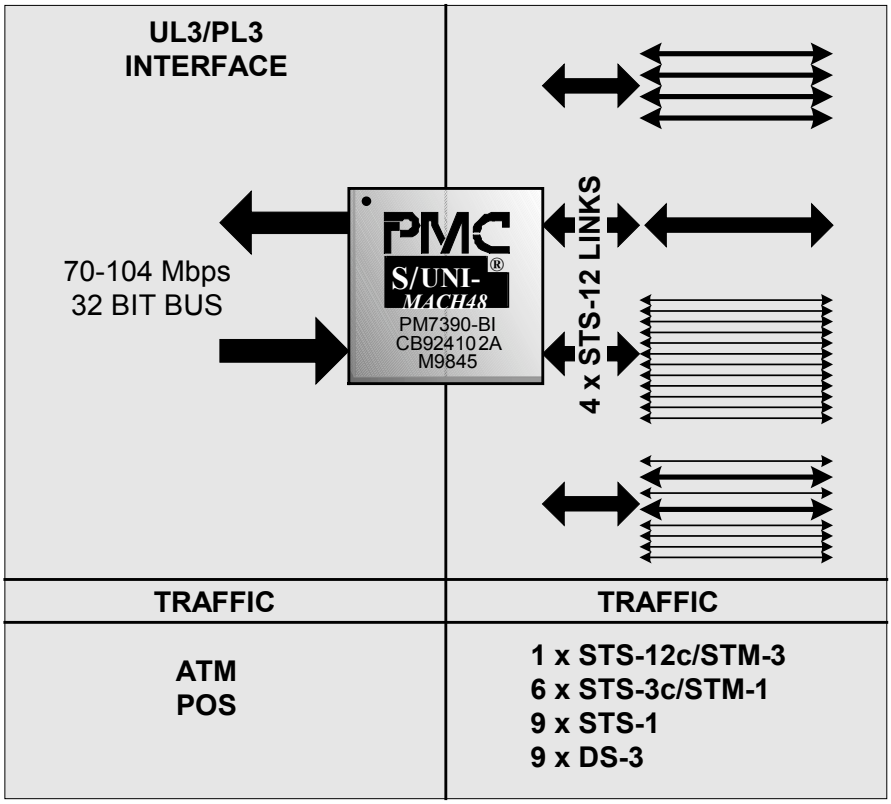
Table 82 Enabling RXPHY/TXPHY Blocks

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0040h	RXPHY Configuration	0000h	0000h
0048h	TXPHY Configuration	0000h	00C0h

6.4 Example 4: 25 PHYs/ Mixed Rate

This example shows how mixed rate channels can be implemented.

Figure 5 - Mixed Rate Example



6.4.1 Set Configuration Register

The setup is for Serial Telecombus and either UL3 or PL3

Table 83 Config Register (Mixed Rate), S-TCB

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0001h	Master Reset, Configuration and Global Digital Loopback	0150h	01D0h

For Diagnostic loopback, the register is set up as follows:

Table 84 Config Register (Mixed Rate), DLOOP

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0001h	Master Reset, Configuration and Global Digital Loopback	0152h	01D2h

6.4.2 Set Timeslot and Delay Registers (Mixed)

The timeslot registers are set up according to their bandwidth.

Table 85 Mixed Rate Timeslot Registers

ADDRESS	DESCRIPTION	VALUE
0002h	Receive Timeslot Configuration #1	5555h
0003h	Receive Timeslot Configuration #2	0055h
0004h	Receive Timeslot Configuration #3	0000h
0005h	Receive Timeslot Configuration #4	AAAAh
0006h	Receive Timeslot Configuration #5	99AAh
0007h	Receive Timeslot Configuration #6	9999h
0008h	Transmit Timeslot Configuration #1	5555h
0009h	Transmit Timeslot Configuration #2	0055h
000Ah	Transmit Timeslot Configuration #3	0000h
000Bh	Transmit Timeslot Configuration #4	AAAAh
000Ch	Transmit Timeslot Configuration #5	99AAh
000Dh	Transmit Timeslot Configuration #6	9999h

Set the RJ0DLY, this value will need to be modified depending on the system configuration.

Table 86 RX S-TCB Synchronization Delay

ADDRESS	DESCRIPTION	VALUE
0011h	Receive Serial Telcombus Synchronization Delay	007Fh

Set the IWTI, IPTI, OPTI and OWTI modes in the Miscellaneous Register to bypass.

Table 87 Miscellaneous Register 0012h

ADDRESS	DESCRIPTION	VALUE
0012h	Miscellaneous	8055h

6.4.3 Set Serial Telecombus Registers

Similarly to Example 1, Section 6.1.3, the MACH48 is set up in Serial Telecombus Mode. Set up the working and protect links for High Order Path Termination (HPT) mode. The R8TDx +03h must also be set to CC34h.

Set TMODEx[1..0] => 01 for HPT mode.

6.4.4 Set Up RXPHY Calendar (Mixed Rate)

The calendar length for this example is 48 entries.

Table 88 Calendar Length Register (Mixed Rate)

ADDRESS	DESCRIPTION	VALUE
0044h	RXPHY Calendar Length	002Fh

The table is set up such that it creates a weighted probability of servicing a channel depending on its bandwidth. The table below shows one possible implementation.

Table 89 RXPHY Calendar Setup (Mixed Rate)

Calendar Entry	Channel #	Calendar Entry	Channel #	Calendar Entry	Channel #
0	12	**16**	12	**32**	12
1	0	**17**	0	**33**	0

Calendar Entry	Channel #	Calendar Entry	Channel #	Calendar Entry	Channel #
2	1	**18**	1	**34**	1
3	2	**19**	2	**35**	2
4	12	**20**	12	**36**	12
5	3	**21**	3	**37**	3
6	36	**22**	36	**38**	36
7	38	**23**	38	**39**	38
8	12	**24**	12	**40**	12
9	24	**25**	30	**41**	37
10	25	**26**	31	**42**	39
11	26	**27**	32	**43**	41
12	12	**28**	12	**44**	12
13	27	**29**	33	**45**	43
14	28	**30**	34	**46**	45
15	29	**31**	35	**47**	47

6.4.5 Set Up SDQ (Mixed Rate)

The table below shows how this example is set up with respect to FIFO block sizes.

Table 90 25 x Mixed Rate Example FIFO Setup

Channel # PHYID[5:0]	BW	Size (Block)	Size (Cell)	FIFO_BS [1:0]	Bank	FIFO# [5:0]	Block_PTR [4:0]
							(hex)
0	STS-3c	48	12	2	0	0	00
1						1	06
2						2	0C
3						3	12
12	STS-12c	192	48	3	1	16	00
24	DS-3	16	4	1	2	32	00
25	DS-3					33	06
26	STS-1					34	0C

Channel # PHYID[5:0]	BW	Size (Block)	Size (Cell)	FIFO_BS [1:0]	Bank	FIFO# [5:0]	Block_PTR [4:0]
							(hex)
27	STS-1					35	12
28	STS-1					36	02
29	DS-3					37	08
30	DS-3					38	0E
31	STS-1	16	4	1	2	39	14
32	DS-3					40	04
33	STS-1					41	0A
34	DS-3					42	10
35	STS-1					43	16
36	STS-3c	48	12	2	3	48	00
37	STS-1	16	4	1	3	49	06
38	STS-3c	48	12	2	3	50	0C
39	DS-3					51	12
41	STS-1					53	08
43	DS-3	16	4	1	3	55	14
45	STS-1					57	0A
47	DS-3					59	16

The following is an example of how an SDQ entry is performed for a STS-12c connection. The table below shows the required register parameters to set up the PHYID = 12 FIFO.

Table 91 Example SDQ Entry Setup for PHYID 12 (STS-12c)

Description	Value(DEC)	Value (BIN)	Register	Note
PHYID	12	001100b	0058h	PHYID for the STS-12c channel
FIFO NUMBER	16	010000b	0059h	Associates a FIFO# with a PHYID
FIFO BLOCK SIZE	3	11b	0059h	A Block size of 192 (as per Data Sheet Section 14)
DATA AVAILABLE THRESHOLD	3 or 95	00000011b or 01011111b	005A	= 3 for UL3 and variable for PL3 depending on required burst size, in this example=95.

These values are written to the RXSDQ in the following order:

1. Poll the BUSY bit in register 0058h until it reads 0.

2. Write the data to the addresses as shown in Table 92.

3. Write the Indirect Register Address with the RWB bit set to 0 and the PHYID. This will initiate the write of the indirect data registers.

Writing to the TXSDQ is done in the same manner.

Table 92 SDQ Register Setup for PHYID 12 STS-12c Entry

ADDRESS	DESCRIPTION	ATM VALUE	POS VALUE
0059h	RXSDQ FIFO Indirect Configuration	90C0h	D0C0h
005Ah	RXSDQ FIFO Indirect Data AvailableThreshold	0303h	0703h
0069h	TXSDQ FIFO Indirect Configuration	90C0h	D0C0h
006Ah	TXSDQ FIFO Inidrect Data Available Threshold	0303h	5F07h

6.4.6 Receive Cell and Frame Processor

Channel 12 is setup as an STS-12c/STM-4 data stream.

Table 93 RCFP1 Register (STS-12c)

ADDRESS	DESCRIPTION	ATM VALUE	POS VALUE
0080h	RCFP Configuration Register #1	0111h	0511h

The remaining channels are set up according to the table below.

Table 94 RTDP Register Setup (6 x STS-3c, 9 x STS-1, 9 x DS3)

<table>
<tr><th>Channel #</th><th>RTDP</th><th colspan="6">OFFSET</th></tr>
<tr><th>PHYID[5:0]</th><th>BASE</th><th colspan="2">01</th><th>02</th><th>03</th><th>04</th><th>05</th></tr>
<tr><th></th><th></th><th>POS</th><th>ATM</th><th></th><th></th><th></th><th></th></tr>
<tr><td>0</td><td rowspan="4">00F0h</td><td rowspan="4">0511h</td><td rowspan="4">0111h</td><td rowspan="4">0402h</td><td rowspan="4">0300h</td><td rowspan="4">0168h</td><td rowspan="4">01FFh</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>12</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr>
<tr><td>24</td><td rowspan="3">0130h</td><td>0D11h</td><td>N/A</td><td>0403h</td><td rowspan="3">0300h</td><td rowspan="3">0168h</td><td rowspan="3">01FFh</td></tr>
<tr><td>25</td><td>N/A</td><td rowspan="2">0111h</td><td rowspan="2">0402h</td></tr>
<tr><td>26</td><td>0511h</td></tr>
</table>

Channel #	RTDP	OFFSET				
PHYID[5:0]	BASE	01		02	03	04	05
		POS	ATM				
27	0130h	0511h	0111h	0402h	0300h	0168h	01FFh
28							
29		N/A	0111h				
30							
31		0511h	0111h				
32		N/A	0111h				
33		0511h	0111h				
34		0D11h	N/A	0403h			
35		0511h	0111h	0402h			
36	0150h	0511h	0111h	0402h	0300h	0168h	01FFh
37							
38							
39		N/A	0111h				
41		0511h	0111h				
43							
45		N/A	0111h				
47		0D11h	N/A	0403h			

6.4.7 Transmit Cell and Frame Processor

Similarly to the RCFP, the TCFP is used to set up Channel 12.

Table 95 TCFP Register Setup (Mixed Rate)

ADDRESS	DESCRIPTION	VALUE
TCFP1_BASE + 00h	TCFP1 Configuration	0181h

The TTDP is set up according to the table below.

Table 96 TTDP Register Setup (Mixed Rate)

Channel # PHYID[5:0]	TTDP BASE	OFFSET			
		01h		02h	03h
		POS	ATM		
0	0170h	0301h	0101h	016Ah	0003h
1					
2					
3					
12	N/A	N/A	N/A	N/A	N/A
24	0190h	0701h	N/A	016Ah	0007h
25		N/A	0101h		0003h
26		0301h			
27					
28					
29		N/A			
30					
31		0301h			
32		N/A			
33		0301h			
34		0701h	N/A		0007h
35		0301h	0101h		0003h
36	01A0h	0301h	0101h	016Ah	0003h
37					
38					
39		N/A			
41		0301h			
43					
45		N/A			
47		0701h	N/A		0007h

The following is an example of how a TTDP entry is performed. The table below shows the required register values to set up the PHYID = 37 entry in Table 96.

Table 97 TTDP Register Setup for PHYID 37 STS-1 Entry

ADDRESS	DESCRIPTION	ATM VALUE	POS VALUE
TTDP2_BASE + 00h	TTDP Indirect Channel Select	0001h	0001h
TTDP2_BASE + 01h	TTDP Indirect Configuration	0101h	0301h
TTDP2_BASE + 02h	TTDP Indirect Idle/Unassigned ATM Cell Header	016Ah	016Ah
TTDP2_BASE + 03h	TTDP Indirect Diagnostics	0003h	0003h

6.4.8 Receive Channel Assigner

TSxMode = 001 for STS-3c/STM-1, TSx_PROV is set to 1

The registers should be written as follows for the RCAS0 block:

Table 98 RCAS0 Register Setup

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE + 0002h	RCAS Timeslot Configuration #0	0030h
RCAS0_BASE + 0003h	RCAS Timeslot Configuration #1	0031h
RCAS0_BASE + 0004h	RCAS Timeslot Configuration #2	0032h
RCAS0_BASE + 0005h	RCAS Timeslot Configuration #3	0033h
RCAS0_BASE + 0006h	RCAS Timeslot Configuration #4	0030h
RCAS0_BASE + 0007h	RCAS Timeslot Configuration #5	0031h
RCAS0_BASE + 0008h	RCAS Timeslot Configuration #6	0032h
RCAS0_BASE + 0009h	RCAS Timeslot Configuration #7	0033h
RCAS0_BASE + 000Ah	RCAS Timeslot Configuration #8	0030h
RCAS0_BASE + 000Bh	RCAS Timeslot Configuration #9	0031h
RCAS0_BASE + 000Ch	RCAS Timeslot Configuration #10	0032h
RCAS0_BASE + 000Dh	RCAS Timeslot Configuration #11	0033h

TsxMode = 000 for STS-12c/STM-4, TSx_PROV is set to 1

The registers should be written as follows for the RCAS1 block:

Table 99 RCAS1 Register Setup

ADDRESS	DESCRIPTION	VALUE
RCAS1_BASE + 0002h	RCAS Timeslot Configuration #0	0010h
RCAS1_BASE + 0003h	RCAS Timeslot Configuration #1	0010h
RCAS1_BASE + 0004h	RCAS Timeslot Configuration #2	0010h

ADDRESS	DESCRIPTION	VALUE
RCAS1_BASE + 0005h	RCAS Timeslot Configuration #3	0010h
RCAS1_BASE + 0006h	RCAS Timeslot Configuration #4	0010h
RCAS1_BASE + 0007h	RCAS Timeslot Configuration #5	0010h
RCAS1_BASE + 0008h	RCAS Timeslot Configuration #6	0010h
RCAS1_BASE + 0009h	RCAS Timeslot Configuration #7	0010h
RCAS1_BASE + 000Ah	RCAS Timeslot Configuration #8	0010h
RCAS1_BASE + 000Bh	RCAS Timeslot Configuration #9	0010h
RCAS1_BASE + 000Ch	RCAS Timeslot Configuration #10	0010h
RCAS1_BASE + 000Dh	RCAS Timeslot Configuration #11	0010h

TSxMode = 011 for DS3 HDLC

TSxMode = 100 for DS3 Direct Mapped ATM

TSxMode = 010 for STS-1/STM-0

TSx_PROV is set to 1

The registers should be written as follows for the RCAS2 block:

Table 100 RCAS2 Register Setup

ADDRESS	DESCRIPTION	VALUE
RCAS2_BASE + 0002h	RCAS Timeslot Configuration #0	0070h
RCAS2_BASE + 0003h	RCAS Timeslot Configuration #1	0091h
RCAS2_BASE + 0004h	RCAS Timeslot Configuration #2	0052h
RCAS2_BASE + 0005h	RCAS Timeslot Configuration #3	0053h
RCAS2_BASE + 0006h	RCAS Timeslot Configuration #4	0054h
RCAS2_BASE + 0007h	RCAS Timeslot Configuration #5	00B5h
RCAS2_BASE + 0008h	RCAS Timeslot Configuration #6	00B6h
RCAS2_BASE + 0009h	RCAS Timeslot Configuration #7	0057h
RCAS2_BASE + 000Ah	RCAS Timeslot Configuration #8	0098h
RCAS2_BASE + 000Bh	RCAS Timeslot Configuration #9	0059h
RCAS2_BASE + 000Ch	RCAS Timeslot Configuration #10	007Ah
RCAS2_BASE + 000Dh	RCAS Timeslot Configuration #11	005Bh

TSxMode = 010 for STS-1/STM-0

TSxMode = 101 for DS3 PLCP ATM

TSx_PROV is set to 1

The registers should be written as follows for the RCAS3 block:

Table 101 RCAS3 Register Setup

ADDRESS	DESCRIPTION	VALUE
RCAS3_BASE + 0002h	RCAS Timeslot Configuration #0	0030h
RCAS3_BASE + 0003h	RCAS Timeslot Configuration #1	0051h
RCAS3_BASE + 0004h	RCAS Timeslot Configuration #2	0032h
RCAS3_BASE + 0005h	RCAS Timeslot Configuration #3	00B3h
RCAS3_BASE + 0006h	RCAS Timeslot Configuration #4	0030h
RCAS3_BASE + 0007h	RCAS Timeslot Configuration #5	0055h
RCAS3_BASE + 0008h	RCAS Timeslot Configuration #6	0032h
RCAS3_BASE + 0009h	RCAS Timeslot Configuration #7	0097h
RCAS3_BASE + 000Ah	RCAS Timeslot Configuration #8	0030h
RCAS3_BASE + 000Bh	RCAS Timeslot Configuration #9	0059h
RCAS3_BASE + 000Ch	RCAS Timeslot Configuration #10	0032h
RCAS3_BASE + 000Dh	RCAS Timeslot Configuration #11	007Bh

If channel based diagnostic loopback is required, the following registers should be written:

Table 102 Settings for Loopback

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE (0x01B0h) + (0001h)	RCAS0 Channel Loopback Enable	000Fh
RCAS1_BASE (0x01C0h) + (0001h)	RCAS1 Channel Loopback Enable	1001h
RCAS2_BASE (0x01D0h) + (0001h)	RCAS2 Channel Loopback Enable	0FFFh
RCAS3_BASE (0x01E0h) + (0001h)	RCAS3 Channel Loopback Enable	0AAFh

6.4.9 Transmit Channel Assigner

The TCAS registers should be setup identically to the RCAS registers.

Table 103 TCAS0 Register Setup

ADDRESS	DESCRIPTION	VALUE
TCAS0_BASE + 0002h	TCAS0 Configuration #0	0030h
TCAS0_BASE + 0003h	TCAS0 Configuration #1	0031h
TCAS0_BASE + 0004h	TCAS0 Configuration #2	0032h
TCAS0_BASE + 0005h	TCAS0 Configuration #3	0033h
TCAS0_BASE + 0006h	TCAS0 Configuration #4	0030h
TCAS0_BASE + 0007h	TCAS0 Configuration #5	0031h
TCAS0_BASE + 0008h	TCAS0 Configuration #6	0032h
TCAS0_BASE + 0009h	TCAS0 Configuration #7	0033h
TCAS0_BASE + 000Ah	TCAS0 Configuration #8	0030h
TCAS0_BASE + 000Bh	TCAS0 Configuration #9	0031h
TCAS0_BASE + 000Ch	TCAS0 Configuration #10	0032h
TCAS0_BASE + 000Dh	TCAS0 Configuration #11	0033h

Table 104 TCAS1 Register Setup

ADDRESS	DESCRIPTION	VALUE
TCAS1_BASE + 0002h	TCAS1 Configuration #0	0010h
TCAS1_BASE + 0003h	TCAS1 Configuration #1	0010h
TCAS1_BASE + 0004h	TCAS1 Configuration #2	0010h
TCAS1_BASE + 0005h	TCAS1 Configuration #3	0010h
TCAS1_BASE + 0006h	TCAS1 Configuration #4	0010h
TCAS1_BASE + 0007h	TCAS1 Configuration #5	0010h
TCAS1_BASE + 0008h	TCAS1 Configuration #6	0010h
TCAS1_BASE + 0009h	TCAS1 Configuration #7	0010h
TCAS1_BASE + 000Ah	TCAS1 Configuration #8	0010h
TCAS1_BASE + 000Bh	TCAS1 Configuration #9	0010h
TCAS1_BASE + 000Ch	TCAS1 Configuration #10	0010h
TCAS1_BASE + 000Dh	TCAS1 Configuration #11	0010h

Table 105 TCAS2 Register Setup

ADDRESS	DESCRIPTION	VALUE
TCAS2_BASE + 0002h	TCAS2 Configuration #0	0070h
TCAS2_BASE + 0003h	TCAS2 Configuration #1	0091h
TCAS2_BASE + 0004h	TCAS2 Configuration #2	0052h
TCAS2_BASE + 0005h	TCAS2 Configuration #3	0053h
TCAS2_BASE + 0006h	TCAS2 Configuration #4	0054h
TCAS2_BASE + 0007h	TCAS2 Configuration #5	00B5h
TCAS2_BASE + 0008h	TCAS2 Configuration #6	00B6h
TCAS2_BASE + 0009h	TCAS2 Configuration #7	0057h
TCAS2_BASE + 000Ah	TCAS2 Configuration #8	0098h
TCAS2_BASE + 000Bh	TCAS2 Configuration #9	0059h
TCAS2_BASE + 000Ch	TCAS2 Configuration #10	007Ah
TCAS2_BASE + 000Dh	TCAS2 Configuration #11	005Bh

Table 106 TCAS3 Register Setup

ADDRESS	DESCRIPTION	VALUE
TCAS3_BASE + 0002h	TCAS3 Configuration #0	0030h
TCAS3_BASE + 0003h	TCAS3 Configuration #1	0051h
TCAS3_BASE + 0004h	TCAS3 Configuration #2	0032h
TCAS3_BASE + 0005h	TCAS3 Configuration #3	00B3h
TCAS3_BASE + 0006h	TCAS3 Configuration #4	0030h
TCAS3_BASE + 0007h	TCAS3 Configuration #5	0055h
TCAS3_BASE + 0008h	TCAS3 Configuration #6	0032h
TCAS3_BASE + 0009h	TCAS3 Configuration #7	0097h
TCAS3_BASE + 000Ah	TCAS3 Configuration #8	0030h
TCAS3_BASE + 000Bh	TCAS3 Configuration #9	0059h
TCAS3_BASE + 000Ch	TCAS3 Configuration #10	0032h
TCAS3_BASE + 000Dh	TCAS3 Configuration #11	007Bh

6.4.10 DS-3 Setup

The DS-3 channels are setup as follows:

Table 107 DS-3 Channel Arrangement

CHANNEL	TYPE	DS3_BASE	Timeslot Location
24	DS-3 HDLC	0B30h	9,1
25	DS-3 Direct Mapped ATM	0B90h	10,1
29	DS-3 PLCP ATM	0D10h	10,2
30	DS-3 PLCP ATM	0D70h	11,2
32	DS-3 Direct Mapped ATM	0E90h	9,3
34	DS-3 HDLC	0EF0h	11,3
39	DS-3 PLCP ATM	10D0h	16,1
45	DS-3 Direct Mapped ATM	13D0h	14,3
47	DS-3 HDLC	13D0h	16,3

In order to set up the channels in the above manner, the following registers must be written.

Table 108 DS-3 Register Setup

ADDRESS	DESCRIPTION	VALUE
DS3_BASE + 08h	DS3 FRMR Configuration	0089h
DS3_BASE + 0Ch	DS3 TRAN Configuration	0001h

For the PLCP ATM Channels, the following registers are written.

Table 109 DS3 PLCP Register Setup

ADDRESS	DESCRIPTION	VALUE
DS3_BASE	SPLR Configuration	0004h
DS3_BASE + 04Ch	SPLT Configuration	0004h

6.4.11 SONET/SDH Inband Error Processor (Mixed Rate)

The SIRP1 is setup for 4 x STS-3c connections.

Table 110 - Mixed Rate SIRP1 Setup

ADDRESS	DESCRIPTION	VALUE
SIRP1_BASE + 00h	SIRP Timeslot #0 Configuration	0007h
SIRP1_BASE + 01h	SIRP Timeslot #1 Configuration	0007h
SIRP1_BASE + 02h	SIRP Timeslot #2 Configuration	0007h
SIRP1_BASE + 03h	SIRP Timeslot #3 Configuration	0007h

The SIRP2 is set up for one STS-12c connection

Table 111 - Mixed Rate SIRP2 Setup

ADDRESS	DESCRIPTION	VALUE
SIRP2_BASE + 00h	SIRP Timeslot #0 Configuration	0007h

The SIRP3 is set up for 12 x STS-1/DS-3 Connections.

Table 112 - Mixed Rate SIRP3 Setup

ADDRESS	DESCRIPTION	VALUE
SIRP3_BASE + 00h	SIRP Timeslot #0 Configuration	00071h
SIRP3_BASE + 01h	SIRP Timeslot #1 Configuration	0007h
SIRP3_BASE + 02h	SIRP Timeslot #2 Configuration	0007h
SIRP3_BASE + 03h	SIRP Timeslot #3 Configuration	0007h
SIRP3_BASE + 04h	SIRP Timeslot #4 Configuration	0007h
SIRP3_BASE + 05h	SIRP Timeslot #5 Configuration	0007h
SIRP3_BASE + 06h	SIRP Timeslot #6 Configuration	0007h
SIRP3_BASE + 07h	SIRP Timeslot #7 Configuration	0007h
SIRP3_BASE + 08h	SIRP Timeslot #8 Configuration	0007h
SIRP3_BASE + 09h	SIRP Timeslot #9 Configuration	0007h
SIRP3_BASE + 0Ah	SIRP Timeslot #A Configuration	0007h
SIRP3_BASE + 0Bh	SIRP Timeslot #B Configuration	0007h

The SIRP4 is set up for 2 x STS-3c and 6 x STS-1/DS-3 connections.

Table 113 - Mixed Rate SIRP4 Setup

ADDRESS	DESCRIPTION	VALUE
SIRP1_BASE + 00h	SIRP Timeslot #0 Configuration	0007h
SIRP1_BASE + 01h	SIRP Timeslot #1 Configuration	0007h
SIRP1_BASE + 02h	SIRP Timeslot #2 Configuration	0007h
SIRP1_BASE + 03h	SIRP Timeslot #3 Configuration	0007h
SIRP1_BASE + 05h	SIRP Timeslot #6 Configuration	0007h
SIRP1_BASE + 07h	SIRP Timeslot #7 Configuration	0007h
SIRP1_BASE + 09h	SIRP Timeslot #9 Configuration	0007h
SIRP1_BASE + 0Bh	SIRP Timeslot #11 Configuration	0007h

6.4.12 Enable Blocks (Mixed Rate)

Once all of the registers are set up as required, the blocks can be enabled.

Table 114 Enabling RCAS/TCAS Blocks

ADDRESS	DESCRIPTION	VALUE
RCAS0_BASE	RCAS12 Channel Disable	0FF0h
RCAS1_BASE	RCAS12 Channel Disable	0FFEh
RCAS2_BASE	RCAS12 Channel Disable	0000h
RCAS3_BASE	RCAS12 Channel Disable	0F00h
TCAS0_BASE	TCAS12 Channel Configuration	0FF0h
TCAS1_BASE	TCAS12 Channel Configuration	0FFEh
TCAS2_BASE	TCAS12 Channel Configuration	0000h
TCAS3_BASE	TCAS12 Channel Configuration	0F00h

Table 115 Enabling RXSDQ/TXSDQ Blocks

ADDRESS	DESCRIPTION	VALUE
0050h	RXSDQ FIFO Reset	0000h
0060h	TXSDQ FIFO Reset	0000h

Table 116 Enabling RXPHY/TXPHY Blocks

ADDRESS	DESCRIPTION	UL3 VALUE	PL3 VALUE
0040h	RXPHY Configuration	0000h	0000h
0048h	TXPHY Configuration	0000h	0040h

7 TCL EXAMPLE CODE

The following code fragments are tcl scripts used with the S/UNI MACH48 Reference Design. They demonstrate the configurations outlined in this document. This code was used on the CHESS reference design that incorporates a SPECTRA-2488 or SPECTRA-4x155, a TBS, a TSE and the S/UNI MACH48.

7.1 General Tcl Procedures

The following Tcl procedures are examples of indirect reads and writes used in configuration examples, sections 7.2-7.4.

```
#################################################################
#PROC WAITONBUSY: This procedure will wait for a specified bit
#to go to a value of one before returning.
#################################################################
proc waitOnBusy {iSlot xAddress xBusyBitMask} {

set busyBit 1
while {$busyBit} {
  set busyBit [read $iSlot $xAddress]
  set busyBit [expr {$busyBit & $xBusyBitMask}]
}
}

#################################################################
#PROC SDQSETUP: This procedure will write values to the SDQ
#indirect registers.
#################################################################
proc sdqSetup {iSlot iSdqOffset iPhyID iFifoNum iBlockPtr iFifoBS iPosSelect
iEnable iDT iBT} {

#puts "Wait for the busy bit"
waitOnBusy $iSlot [expr {$iSdqOffset + 0x0008}] 0x8000

set tempSdqData [expr { 0x0000 +
                    ($iEnable << 15) +
                    ($iPosSelect << 14) +
                    ($iFifoNum << 8) +
                    ($iFifoBS << 6) +
                    ($iBlockPtr) } ]
#puts [format "Writing %x to %x + 0x0009" $tempSdqData $iSdqOffset]
write $iSlot [expr {$iSdqOffset + 0x0009}] $tempSdqData
```

```
set tempSdqData [expr { 0x0000 + ($iDT << 8) + ($iBT) } ]
#puts [format "Writing %x to %x + 0x000A" $tempSdqData $iSdqOffset]
write $iSlot [expr {$iSdqOffset + 0x000A}] $tempSdqData

set tempSdqData [expr { 0x0000 + ($iPhyID) } ]
#puts [format "Writing %x to %x + 0x0008" $tempSdqData $iSdqOffset]
write $iSlot [expr {$iSdqOffset + 0x0008}] $tempSdqData

}

################################################################
#PROC RTDPSETUP: This procedure will write values to the RTDP
#indirect registers.
################################################################
proc rtdpSetup {iSlot iPhyID xReg1 xReg2 xReg3 xReg4 xReg5} {

global RTDP0_BASE RTDP1_BASE RTDP2_BASE RTDP3_BASE
if {$iPhyID < 12} {
  set rtdpOffset $RTDP0_BASE
} elseif {$iPhyID < 24} {
  set rtdpOffset $RTDP1_BASE
} elseif {$iPhyID < 36} {
  set rtdpOffset $RTDP2_BASE
} else {
  set rtdpOffset $RTDP3_BASE
}

#Wait for the busy bit
waitOnBusy $iSlot $rtdpOffset 0x8000

write $iSlot [expr {$rtdpOffset + 0x0001}] $xReg1
write $iSlot [expr {$rtdpOffset + 0x0002}] $xReg2
write $iSlot [expr {$rtdpOffset + 0x0003}] $xReg3
write $iSlot [expr {$rtdpOffset + 0x0004}] $xReg4
write $iSlot [expr {$rtdpOffset + 0x0005}] $xReg5
write $iSlot $rtdpOffset [expr {int([expr {fmod($iPhyID,12)}])}]
}

################################################################
#PROC TTDPSETUP: This procedure will write values to the TTDP
#indirect registers.
################################################################
proc ttdpSetup {iSlot iPhyID xReg1 xReg2 xReg3} {

global TTDP0_BASE TTDP1_BASE TTDP2_BASE TTDP3_BASE
if {$iPhyID < 12} {
  set ttdpOffset $TTDP0_BASE
```

```
} elseif {$iPhyID < 24} {
  set ttdpOffset $TTDP1_BASE
} elseif {$iPhyID < 36} {
  set ttdpOffset $TTDP2_BASE
} else {
  set ttdpOffset $TTDP3_BASE
}

#Wait for the busy bit
waitOnBusy $iSlot $ttdpOffset 0x8000

write $iSlot [expr {$ttdpOffset + 0x0001}] $xReg1
write $iSlot [expr {$ttdpOffset + 0x0002}] $xReg2
write $iSlot [expr {$ttdpOffset + 0x0003}] $xReg3
write $iSlot $ttdpOffset [expr {int([expr {fmod($iPhyID,12)}])}]
}

######################################################################
#PROC SETRXPHYCALENTRY: This procedure will write an indirect calendar
#entry.
######################################################################
proc setRxphyCalEntry {iSlot iCalPos iChan} {

global RXPHY_BASE

# Wait for the busy bit
waitOnBusy $iSlot [expr {$RXPHY_BASE + 0x0005}] 0x8000
write $iSlot [expr {$RXPHY_BASE + 0x0005}] [expr {$iChan + ($iCalPos << 8)}]
}
######################################################################
#PROC RXPHYCALREPORT: This procedure will output the current data
#written to the calendar.
######################################################################
proc RxphyCalReport {iSlot} {

global RXPHY_BASE

set numEntries [expr {[read $iSlot [expr {$RXPHY_BASE + 0x0004}]] & 0x007F}]
puts [format "Calendar Size = %d" $numEntries]
for {set i 0} {$i < ($numEntries+1)} {incr i 1} {
 # Wait for the busy bit
 waitOnBusy $iSlot [expr {$RXPHY_BASE + 0x0005}] 0x8000
 write $iSlot [expr {$RXPHY_BASE + 0x0005}] [expr {($i << 8) + 0x0080}]
 # Wait for the busy bit
 waitOnBusy $iSlot [expr {$RXPHY_BASE + 0x0005}] 0x8000
 set rxphyTempData [read $iSlot [expr {$RXPHY_BASE + 0x0005}]]
 set calAddr [expr {($rxphyTempData & 0x7F00) >> 8}]
```

```
 set calData [expr {($rxphyTempData & 0x003F)}]
 puts [format "Rxphy calendar addr %d contains channel %d" $calAddr $calData ]
 }
}
#########################################################################
#PROC MACHINIT: This Procedure will write the required default values
#to the MACH48
#########################################################################
proc MACHinit {iSlot} {

 global WT8TE0_BASE WT8TE1_BASE WT8TE2_BASE WT8TE3_BASE
  global PT8TE0_BASE PT8TE1_BASE PT8TE2_BASE PT8TE3_BASE
  global WR8TD0_BASE WR8TD1_BASE WR8TD2_BASE WR8TD3_BASE
  global PR8TD0_BASE PR8TD1_BASE PR8TD2_BASE PR8TD3_BASE

puts "Initializing MACH"
    # Set the T8TE's to run in HPT mode
set mode 0x5555
set mode2 0x0055
    write $iSlot [expr {$WT8TE0_BASE + 0x0002}] $mode
    write $iSlot [expr {$WT8TE0_BASE + 0x0003}] $mode2
    write $iSlot [expr {$WT8TE1_BASE + 0x0002}] $mode
    write $iSlot [expr {$WT8TE1_BASE + 0x0003}] $mode2
    write $iSlot [expr {$WT8TE2_BASE + 0x0002}] $mode
    write $iSlot [expr {$WT8TE2_BASE + 0x0003}] $mode2
    write $iSlot [expr {$WT8TE3_BASE + 0x0002}] $mode
    write $iSlot [expr {$WT8TE3_BASE + 0x0003}] $mode2
    write $iSlot [expr {$PT8TE0_BASE + 0x0002}] $mode
    write $iSlot [expr {$PT8TE0_BASE + 0x0003}] $mode2
    write $iSlot [expr {$PT8TE1_BASE + 0x0002}] $mode
    write $iSlot [expr {$PT8TE1_BASE + 0x0003}] $mode2
    write $iSlot [expr {$PT8TE2_BASE + 0x0002}] $mode
    write $iSlot [expr {$PT8TE2_BASE + 0x0003}] $mode2
    write $iSlot [expr {$PT8TE3_BASE + 0x0002}] $mode
    write $iSlot [expr {$PT8TE3_BASE + 0x0003}] $mode2

    #Analog patches
    write $iSlot [expr {$WR8TD0_BASE + 0x3}] 0xCC34
    write $iSlot [expr {$WR8TD1_BASE + 0x3}] 0xCC34
    write $iSlot [expr {$WR8TD2_BASE + 0x3}] 0xCC34
    write $iSlot [expr {$WR8TD3_BASE + 0x3}] 0xCC34
    write $iSlot [expr {$PR8TD0_BASE + 0x3}] 0xCC34
    write $iSlot [expr {$PR8TD1_BASE + 0x3}] 0xCC34
    write $iSlot [expr {$PR8TD2_BASE + 0x3}] 0xCC34
    write $iSlot [expr {$PR8TD3_BASE + 0x3}] 0xCC34

}
```

7.2 Example 1: 1 x STS-48c ATM PL3

```
############################################################
#SETUP MACH
############################################################
#Take MACH out of Reset
write mach48 0x1 0x3dc

#SET TO HPT MODE, DEFAULT FIX
MACHinit mach48

#Set TSI to STS-48c mode
#IT IS DEFAULT

############################################################
#SETUP THE TCFP
#ONLY THE FIRST BLOCK NEEDS TO BE SET UP

puts "setting up TCFP"
write mach48 0xb0 0x0c3

############################################################
#SETUP THE TCAS

puts "Setting up TCAS"
write mach48 0x1f2 0x0010
write mach48 0x1f3 0x0010
write mach48 0x1f4 0x0010
write mach48 0x1f5 0x0010
write mach48 0x1f6 0x0010
write mach48 0x1f7 0x0010
write mach48 0x1f8 0x0010
write mach48 0x1f9 0x0010
write mach48 0x1fa 0x0010
write mach48 0x1fb 0x0010
write mach48 0x1fc 0x0010
write mach48 0x1fd 0x0010

write mach48 0x202 0x0010
write mach48 0x203 0x0010
write mach48 0x204 0x0010
write mach48 0x205 0x0010
write mach48 0x206 0x0010
write mach48 0x207 0x0010
write mach48 0x208 0x0010
```

```
write mach48 0x209 0x0010
write mach48 0x20a 0x0010
write mach48 0x20b 0x0010
write mach48 0x20c 0x0010
write mach48 0x20d 0x0010

write mach48 0x212 0x0010
write mach48 0x213 0x0010
write mach48 0x214 0x0010
write mach48 0x215 0x0010
write mach48 0x216 0x0010
write mach48 0x217 0x0010
write mach48 0x218 0x0010
write mach48 0x219 0x0010
write mach48 0x21a 0x0010
write mach48 0x21b 0x0010
write mach48 0x21c 0x0010
write mach48 0x21d 0x0010

write mach48 0x222 0x0010
write mach48 0x223 0x0010
write mach48 0x224 0x0010
write mach48 0x225 0x0010
write mach48 0x226 0x0010
write mach48 0x227 0x0010
write mach48 0x228 0x0010
write mach48 0x229 0x0010
write mach48 0x22a 0x0010
write mach48 0x22b 0x0010
write mach48 0x22c 0x0010
write mach48 0x22d 0x0010

#############################################################
#SETUP THE RXPHY

#Set the Calendar Length- ONLY ONE PHY - STS-48c
write mach48 0x44 0

setRxphyCalEntry mach48 0 0

#############################################################
#SETUP THE SDQ's
puts "setting up SDQ's"
#           iSlot  iSdqOffset  iPhyID iFifoNum iBlockPtr iFifoBS iPosSelect
iEnable  iDT iBT
sdqSetup mach48  0x50   0    0   0   3   0   1   3   3
sdqSetup mach48  0x60   0    0   0   3   0   1   3   3
```

```
############################################################
#SETUP THE RCFP - FOR ATM TRAFFIC, POS = 0x041b
#ONLY THE FIRST BLOCK NEEDS TO BE SET UP
puts "setting up RCFP"
write mach48 0x70 0x011b

############################################################
#SETUP THE RCAS
puts "Setting up RCAS"
write mach48 0x1b2 0x0010
write mach48 0x1b3 0x0010
write mach48 0x1b4 0x0010
write mach48 0x1b5 0x0010
write mach48 0x1b6 0x0010
write mach48 0x1b7 0x0010
write mach48 0x1b8 0x0010
write mach48 0x1b9 0x0010
write mach48 0x1ba 0x0010
write mach48 0x1bb 0x0010
write mach48 0x1bc 0x0010
write mach48 0x1bd 0x0010
write mach48 0x1c2 0x0010
write mach48 0x1c3 0x0010
write mach48 0x1c4 0x0010
write mach48 0x1c5 0x0010
write mach48 0x1c6 0x0010
write mach48 0x1c7 0x0010
write mach48 0x1c8 0x0010
write mach48 0x1c9 0x0010
write mach48 0x1ca 0x0010
write mach48 0x1cb 0x0010
write mach48 0x1cc 0x0010
write mach48 0x1cd 0x0010
write mach48 0x1d2 0x0010
write mach48 0x1d3 0x0010
write mach48 0x1d4 0x0010
write mach48 0x1d5 0x0010
write mach48 0x1d6 0x0010
write mach48 0x1d7 0x0010
write mach48 0x1d8 0x0010
write mach48 0x1d9 0x0010
write mach48 0x1da 0x0010
write mach48 0x1db 0x0010
write mach48 0x1dc 0x0010
write mach48 0x1dd 0x0010
write mach48 0x1e2 0x0010
write mach48 0x1e3 0x0010
```

```
write mach48 0x1e4 0x0010
write mach48 0x1e5 0x0010
write mach48 0x1e6 0x0010
write mach48 0x1e7 0x0010
write mach48 0x1e8 0x0010
write mach48 0x1e9 0x0010
write mach48 0x1ea 0x0010
write mach48 0x1eb 0x0010
write mach48 0x1ec 0x0010
write mach48 0x1ed 0x0010

#############################################################
#SETUP THE SIRP
write mach48 0x14c0 0x0007
#############################################################
#ENABLE the BLOCKS
puts "Enabling Blocks"
#Enable SDQs
write mach48 0x60 0
write mach48 0x50 0

#Enable TXPHY, RXPHY
write mach48 0x48 0x40
write mach48 0x40 0

puts "Enable RCAS"
write mach48 0x1b0 0
write mach48 0x1c0 0
write mach48 0x1d0 0
write mach48 0x1e0 0
#############################################################
TSEinit
mapTSE3 tse1
portEnable4
portEnable5
#############################################################
#SETUP THE DELAYS FOR THE SYSTEM
after 1000
write mach48 0x11       127
write mach48 0x13       30
write tse1   0x40       60
write tbs5   0x5        127

#Center the FIFOs for the LVDS Links
center

puts "Enabling TCAS"
```

```
after 1000
write mach48 0x1f0 0
write mach48 0x200 0
write mach48 0x210 0
write mach48 0x220 0
```

7.3 Example 2: 16 x STS-3c ATM PL3

```
#Sets HPT mode and enables R8TDs
MACHinit mach48
#Take the MACH out of reset. Set RHPP_EN,SEREN,DLOOP,POSL3,RWSEL_EN
write mach48 0x1 0x3d0

#Set TSI into BYPASS mode
write mach48 0x12 0x8055

#Setup TTDP on Chan 0-3,12-15,24-27,36-39
puts "setting up TTDP"
ttdpSetup mach48   0  0x0185 0x016A 0x0003
ttdpSetup mach48   1  0x0185 0x016A 0x0003
ttdpSetup mach48   2  0x0185 0x016A 0x0003
ttdpSetup mach48   3  0x0185 0x016A 0x0003
ttdpSetup mach48  12  0x0185 0x016A 0x0003
ttdpSetup mach48  13  0x0185 0x016A 0x0003
ttdpSetup mach48  14  0x0185 0x016A 0x0003
ttdpSetup mach48  15  0x0185 0x016A 0x0003
ttdpSetup mach48  24  0x0185 0x016A 0x0003
ttdpSetup mach48  25  0x0185 0x016A 0x0003
ttdpSetup mach48  26  0x0185 0x016A 0x0003
ttdpSetup mach48  27  0x0185 0x016A 0x0003
ttdpSetup mach48  36  0x0185 0x016A 0x0003
ttdpSetup mach48  37  0x0185 0x016A 0x0003
ttdpSetup mach48  38  0x0185 0x016A 0x0003
ttdpSetup mach48  39  0x0185 0x016A 0x0003


#Setup TCAS_0 for 4xSTS-3c channels
puts "setting up TCAS"
write mach48 0x1f2 0x0030
write mach48 0x1f3 0x0031
write mach48 0x1f4 0x0032
write mach48 0x1f5 0x0033
write mach48 0x1f6 0x0030
write mach48 0x1f7 0x0031
write mach48 0x1f8 0x0032
write mach48 0x1f9 0x0033
```

```
write mach48 0x1fa 0x0030
write mach48 0x1fb 0x0031
write mach48 0x1fc 0x0032
write mach48 0x1fd 0x0033

write mach48 0x202 0x0030
write mach48 0x203 0x0031
write mach48 0x204 0x0032
write mach48 0x205 0x0033
write mach48 0x206 0x0030
write mach48 0x207 0x0031
write mach48 0x208 0x0032
write mach48 0x209 0x0033
write mach48 0x20a 0x0030
write mach48 0x20b 0x0031
write mach48 0x20c 0x0032
write mach48 0x20d 0x0033

write mach48 0x212 0x0030
write mach48 0x213 0x0031
write mach48 0x214 0x0032
write mach48 0x215 0x0033
write mach48 0x216 0x0030
write mach48 0x217 0x0031
write mach48 0x218 0x0032
write mach48 0x219 0x0033
write mach48 0x21a 0x0030
write mach48 0x21b 0x0031
write mach48 0x21c 0x0032
write mach48 0x21d 0x0033

write mach48 0x222 0x0030
write mach48 0x223 0x0031
write mach48 0x224 0x0032
write mach48 0x225 0x0033
write mach48 0x226 0x0030
write mach48 0x227 0x0031
write mach48 0x228 0x0032
write mach48 0x229 0x0033
write mach48 0x22a 0x0030
write mach48 0x22b 0x0031
write mach48 0x22c 0x0032
write mach48 0x22d 0x0033


#################################################################
puts "Setting up RXPHY Calendar "
```

```
# Set the calendar length to 48
write mach48 0x0044 0x002F

setRxphyCalEntry mach48  0   0
setRxphyCalEntry mach48  1  12
setRxphyCalEntry mach48  2  24
setRxphyCalEntry mach48  3  36
setRxphyCalEntry mach48  4   1
setRxphyCalEntry mach48  5  13
setRxphyCalEntry mach48  6  25
setRxphyCalEntry mach48  7  37
setRxphyCalEntry mach48  8   2
setRxphyCalEntry mach48  9  14
setRxphyCalEntry mach48 10  26
setRxphyCalEntry mach48 11  38
setRxphyCalEntry mach48 12   3
setRxphyCalEntry mach48 13  15
setRxphyCalEntry mach48 14  27
setRxphyCalEntry mach48 15  39
setRxphyCalEntry mach48 16   0
setRxphyCalEntry mach48 17  12
setRxphyCalEntry mach48 18  24
setRxphyCalEntry mach48 19  36
setRxphyCalEntry mach48 20   1
setRxphyCalEntry mach48 21  13
setRxphyCalEntry mach48 22  25
setRxphyCalEntry mach48 23  37
setRxphyCalEntry mach48 24   2
setRxphyCalEntry mach48 25  14
setRxphyCalEntry mach48 26  26
setRxphyCalEntry mach48 27  38
setRxphyCalEntry mach48 28   3
setRxphyCalEntry mach48 29  15
setRxphyCalEntry mach48 30  27
setRxphyCalEntry mach48 31  39
setRxphyCalEntry mach48 32   0
setRxphyCalEntry mach48 33  12
setRxphyCalEntry mach48 34  24
setRxphyCalEntry mach48 35  36
setRxphyCalEntry mach48 36   1
setRxphyCalEntry mach48 37  13
setRxphyCalEntry mach48 38  25
setRxphyCalEntry mach48 39  37
setRxphyCalEntry mach48 40   2
setRxphyCalEntry mach48 41  14
setRxphyCalEntry mach48 42  26
```

```
setRxphyCalEntry mach48 43  38
setRxphyCalEntry mach48 44   3
setRxphyCalEntry mach48 45  15
setRxphyCalEntry mach48 46  27
setRxphyCalEntry mach48 47  39

#####################################################################
#####################################################################
puts "Setting up the SDQ"
#        iSlot  iSdqOffset  iPhyID iFifoNum iBlockPtr iFifoBS iPosSelect
iEnable iDT iBT
#
sdqSetup mach48 0x50   0   0    0   2   0   1   3   3
sdqSetup mach48 0x50   1   1    6   2   0   1   3   3
sdqSetup mach48 0x50   2   2   12   2   0   1   3   3
sdqSetup mach48 0x50   3   3   18   2   0   1   3   3
sdqSetup mach48 0x50  12   0    0   2   0   1   3   3
sdqSetup mach48 0x50  13   1    6   2   0   1   3   3
sdqSetup mach48 0x50  14   2   12   2   0   1   3   3
sdqSetup mach48 0x50  15   3   18   2   0   1   3   3
sdqSetup mach48 0x50  24   0    0   2   0   1   3   3
sdqSetup mach48 0x50  25   1    6   2   0   1   3   3
sdqSetup mach48 0x50  26   2   12   2   0   1   3   3
sdqSetup mach48 0x50  27   3   18   2   0   1   3   3
sdqSetup mach48 0x50  36   0    0   2   0   1   3   3
sdqSetup mach48 0x50  37   1    6   2   0   1   3   3
sdqSetup mach48 0x50  38   2   12   2   0   1   3   3
sdqSetup mach48 0x50  39   3   18   2   0   1   3   3


#####################################################################

sdqSetup mach48 0x60   0   0    0   2   0   1   3   3
sdqSetup mach48 0x60   1   1    6   2   0   1   3   3
sdqSetup mach48 0x60   2   2   12   2   0   1   3   3
sdqSetup mach48 0x60   3   3   18   2   0   1   3   3
sdqSetup mach48 0x60  12   0    0   2   0   1   3   3
sdqSetup mach48 0x60  13   1    6   2   0   1   3   3
sdqSetup mach48 0x60  14   2   12   2   0   1   3   3
sdqSetup mach48 0x60  15   3   18   2   0   1   3   3
sdqSetup mach48 0x60  24   0    0   2   0   1   3   3
sdqSetup mach48 0x60  25   1    6   2   0   1   3   3
sdqSetup mach48 0x60  26   2   12   2   0   1   3   3
sdqSetup mach48 0x60  27   3   18   2   0   1   3   3
sdqSetup mach48 0x60  36   0    0   2   0   1   3   3
sdqSetup mach48 0x60  37   1    6   2   0   1   3   3
sdqSetup mach48 0x60  38   2   12   2   0   1   3   3
```

```
sdqSetup mach48 0x60  39   3   18   2   0   1   3   3
#################################################################

#Setup RTDP on Chan 0-3
puts "Setting up RTDP"
rtdpSetup mach48  0  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48  1  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48  2  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48  3  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 12  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 13  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 14  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 15  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 24  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 25  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 26  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 27  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 36  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 37  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 38  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 39  0x011b 0x0402 0x0300 0x0168 0x01FF


#Receive Timeslot Registers (0x0002->0x0007)
puts "Setting up timeslot registers"
write mach48 0x0002 0x5555
write mach48 0x0003 0x5555
write mach48 0x0004 0x5555
write mach48 0x0005 0x5555
write mach48 0x0006 0x5555
write mach48 0x0007 0x5555
#Transmit Timeslot Registers (0x0008->0x000D)
write mach48 0x0008 0x5555
write mach48 0x0009 0x5555
write mach48 0x000a 0x5555
write mach48 0x000b 0x5555
write mach48 0x000c 0x5555
write mach48 0x000d 0x5555

#Setup RCAS_0 for 4xSTS-3c channels
puts "Setting up RCAS"
write mach48 0x1b2 0x0030
write mach48 0x1b3 0x0031
write mach48 0x1b4 0x0032
write mach48 0x1b5 0x0033
write mach48 0x1b6 0x0030
write mach48 0x1b7 0x0031
```

```
write mach48 0x1b8 0x0032
write mach48 0x1b9 0x0033
write mach48 0x1ba 0x0030
write mach48 0x1bb 0x0031
write mach48 0x1bc 0x0032
write mach48 0x1bd 0x0033

write mach48 0x1c2 0x0030
write mach48 0x1c3 0x0031
write mach48 0x1c4 0x0032
write mach48 0x1c5 0x0033
write mach48 0x1c6 0x0030
write mach48 0x1c7 0x0031
write mach48 0x1c8 0x0032
write mach48 0x1c9 0x0033
write mach48 0x1ca 0x0030
write mach48 0x1cb 0x0031
write mach48 0x1cc 0x0032
write mach48 0x1cd 0x0033

write mach48 0x1d2 0x0030
write mach48 0x1d3 0x0031
write mach48 0x1d4 0x0032
write mach48 0x1d5 0x0033
write mach48 0x1d6 0x0030
write mach48 0x1d7 0x0031
write mach48 0x1d8 0x0032
write mach48 0x1d9 0x0033
write mach48 0x1da 0x0030
write mach48 0x1db 0x0031
write mach48 0x1dc 0x0032
write mach48 0x1dd 0x0033

write mach48 0x1e2 0x0030
write mach48 0x1e3 0x0031
write mach48 0x1e4 0x0032
write mach48 0x1e5 0x0033
write mach48 0x1e6 0x0030
write mach48 0x1e7 0x0031
write mach48 0x1e8 0x0032
write mach48 0x1e9 0x0033
write mach48 0x1ea 0x0030
write mach48 0x1eb 0x0031
write mach48 0x1ec 0x0032
write mach48 0x1ed 0x0033

puts "Setting up the SIRP"
```

```
write mach48 0x14c0 0x0007
write mach48 0x14c1 0x0007
write mach48 0x14c2 0x0007
write mach48 0x14c3 0x0007

write mach48 0x14d0 0x0007
write mach48 0x14d1 0x0007
write mach48 0x14d2 0x0007
write mach48 0x14d3 0x0007

write mach48 0x14e0 0x0007
write mach48 0x14e1 0x0007
write mach48 0x14e2 0x0007
write mach48 0x14e3 0x0007

write mach48 0x14f0 0x0007
write mach48 0x14f1 0x0007
write mach48 0x14f2 0x0007
write mach48 0x14f3 0x0007


#Set up RHPP for STS-3c Traffic
write mach48 0x1602 0xf
write mach48 0x1682 0xf
write mach48 0x1702 0xf
write mach48 0x1782 0xf

#Enable BLOCKS:TCAS,RCAS
puts "Enable TX,RX SDQs"
write mach48 0x60 0
write mach48 0x50 0

puts "Enable TX,RX Phys"
write mach48 0x48 0x0040
write mach48 0x40 0

puts "Enabling RCAS"
write mach48 0x1b0 0xff0
write mach48 0x1c0 0xff0
write mach48 0x1d0 0xff0
write mach48 0x1e0 0xff0

TSEinit
mapTSE3 tse1
portEnable4
portEnable5
```

```
#Setup Delays for the system
puts "setting up system delays"
write mach48 0x11 127
write mach48 0x13 30
write tse1 0x40 60
write tbs5 0x5 127

center

puts "Enabling TCAS"
write mach48 0x1f0 0xff0
write mach48 0x200 0xff0
write mach48 0x210 0xff0
write mach48 0x220 0xff0
```

7.4 Example 3: 48 x STS-1 ATM PL3

```
#This script will set the MACH48 up in 48 x STS-1 mode.


#Sets HPT mode and enables R8TDs
MACHinit mach48
#Take the MACH out of reset. Set RHPP_EN,SEREN,POSL3,RWSEL_EN
write mach48 0x1 0x3d0

#Set TSI into BYPASS mode
write mach48 0x12 0x8055

#Setup TTDP
puts "setting up TTDP"
ttdpSetup mach48 0   0x0185 0x016A 0x0003
ttdpSetup mach48 1   0x0185 0x016A 0x0003
ttdpSetup mach48 2   0x0185 0x016A 0x0003
ttdpSetup mach48 3   0x0185 0x016A 0x0003
ttdpSetup mach48 4   0x0185 0x016A 0x0003
ttdpSetup mach48 5   0x0185 0x016A 0x0003
ttdpSetup mach48 6   0x0185 0x016A 0x0003
ttdpSetup mach48 7   0x0185 0x016A 0x0003
ttdpSetup mach48 8   0x0185 0x016A 0x0003
ttdpSetup mach48 9   0x0185 0x016A 0x0003
ttdpSetup mach48 10  0x0185 0x016A 0x0003
ttdpSetup mach48 11  0x0185 0x016A 0x0003

ttdpSetup mach48 12  0x0185 0x016A 0x0003
```

```
ttdpSetup mach48 13  0x0185 0x016A 0x0003
ttdpSetup mach48 14  0x0185 0x016A 0x0003
ttdpSetup mach48 15  0x0185 0x016A 0x0003
ttdpSetup mach48 16  0x0185 0x016A 0x0003
ttdpSetup mach48 17  0x0185 0x016A 0x0003
ttdpSetup mach48 18  0x0185 0x016A 0x0003
ttdpSetup mach48 19  0x0185 0x016A 0x0003
ttdpSetup mach48 20  0x0185 0x016A 0x0003
ttdpSetup mach48 21  0x0185 0x016A 0x0003
ttdpSetup mach48 22  0x0185 0x016A 0x0003
ttdpSetup mach48 23  0x0185 0x016A 0x0003

ttdpSetup mach48 24  0x0185 0x016A 0x0003
ttdpSetup mach48 25  0x0185 0x016A 0x0003
ttdpSetup mach48 26  0x0185 0x016A 0x0003
ttdpSetup mach48 27  0x0185 0x016A 0x0003
ttdpSetup mach48 28  0x0185 0x016A 0x0003
ttdpSetup mach48 29  0x0185 0x016A 0x0003
ttdpSetup mach48 30  0x0185 0x016A 0x0003
ttdpSetup mach48 31  0x0185 0x016A 0x0003
ttdpSetup mach48 32  0x0185 0x016A 0x0003
ttdpSetup mach48 33  0x0185 0x016A 0x0003
ttdpSetup mach48 34  0x0185 0x016A 0x0003
ttdpSetup mach48 35  0x0185 0x016A 0x0003

ttdpSetup mach48 36  0x0185 0x016A 0x0003
ttdpSetup mach48 37  0x0185 0x016A 0x0003
ttdpSetup mach48 38  0x0185 0x016A 0x0003
ttdpSetup mach48 39  0x0185 0x016A 0x0003
ttdpSetup mach48 40  0x0185 0x016A 0x0003
ttdpSetup mach48 41  0x0185 0x016A 0x0003
ttdpSetup mach48 42  0x0185 0x016A 0x0003
ttdpSetup mach48 43  0x0185 0x016A 0x0003
ttdpSetup mach48 44  0x0185 0x016A 0x0003
ttdpSetup mach48 45  0x0185 0x016A 0x0003
ttdpSetup mach48 46  0x0185 0x016A 0x0003
ttdpSetup mach48 47  0x0185 0x016A 0x0003


################################################################
#SETUP THE TRANSMIT CHANNEL ASSIGNER
#Setup TCAS for 48xSTS-1 channels
puts "setting up TCAS_0"
write mach48 0x1f2 0x0050
write mach48 0x1f3 0x0051
write mach48 0x1f4 0x0052
write mach48 0x1f5 0x0053
```

```
write mach48 0x1f6 0x0054
write mach48 0x1f7 0x0055
write mach48 0x1f8 0x0056
write mach48 0x1f9 0x0057
write mach48 0x1fa 0x0058
write mach48 0x1fb 0x0059
write mach48 0x1fc 0x005a
write mach48 0x1fd 0x005b

puts "setting up TCAS_1"
write mach48 0x202 0x0050
write mach48 0x203 0x0051
write mach48 0x204 0x0052
write mach48 0x205 0x0053
write mach48 0x206 0x0054
write mach48 0x207 0x0055
write mach48 0x208 0x0056
write mach48 0x209 0x0057
write mach48 0x20a 0x0058
write mach48 0x20b 0x0059
write mach48 0x20c 0x005a
write mach48 0x20d 0x005b

puts "setting up TCAS_2"
write mach48 0x212 0x0050
write mach48 0x213 0x0051
write mach48 0x214 0x0052
write mach48 0x215 0x0053
write mach48 0x216 0x0054
write mach48 0x217 0x0055
write mach48 0x218 0x0056
write mach48 0x219 0x0057
write mach48 0x21a 0x0058
write mach48 0x21b 0x0059
write mach48 0x21c 0x005a
write mach48 0x21d 0x005b

puts "setting up TCAS_3"
write mach48 0x222 0x0050
write mach48 0x223 0x0051
write mach48 0x224 0x0052
write mach48 0x225 0x0053
write mach48 0x226 0x0054
write mach48 0x227 0x0055
write mach48 0x228 0x0056
write mach48 0x229 0x0057
write mach48 0x22a 0x0058
```

```
write mach48 0x22b 0x0059
write mach48 0x22c 0x005a
write mach48 0x22d 0x005b

#####################################################################
puts "Setting up RXPHY Calendar "

# Set the calendar length to 48
write mach48 0x0044 47

setRxphyCalEntry mach48 0    0
setRxphyCalEntry mach48 1   12
setRxphyCalEntry mach48 2   24
setRxphyCalEntry mach48 3   36
setRxphyCalEntry mach48 4    1
setRxphyCalEntry mach48 5   13
setRxphyCalEntry mach48 6   25
setRxphyCalEntry mach48 7   37
setRxphyCalEntry mach48 8    2
setRxphyCalEntry mach48 9   14
setRxphyCalEntry mach48 10  26
setRxphyCalEntry mach48 11  38
setRxphyCalEntry mach48 12   3
setRxphyCalEntry mach48 13  15
setRxphyCalEntry mach48 14  27
setRxphyCalEntry mach48 15  39
setRxphyCalEntry mach48 16   4
setRxphyCalEntry mach48 17  16
setRxphyCalEntry mach48 18  28
setRxphyCalEntry mach48 19  40
setRxphyCalEntry mach48 20   5
setRxphyCalEntry mach48 21  17
setRxphyCalEntry mach48 22  29
setRxphyCalEntry mach48 23  41
setRxphyCalEntry mach48 24   6
setRxphyCalEntry mach48 25  18
setRxphyCalEntry mach48 26  30
setRxphyCalEntry mach48 27  42
setRxphyCalEntry mach48 28   7
setRxphyCalEntry mach48 29  19
setRxphyCalEntry mach48 30  31
setRxphyCalEntry mach48 31  43
setRxphyCalEntry mach48 32   8
setRxphyCalEntry mach48 33  20
setRxphyCalEntry mach48 34  32
setRxphyCalEntry mach48 35  44
setRxphyCalEntry mach48 36   9
```

```
setRxphyCalEntry mach48 37  21
setRxphyCalEntry mach48 38  33
setRxphyCalEntry mach48 39  45
setRxphyCalEntry mach48 40  10
setRxphyCalEntry mach48 41  22
setRxphyCalEntry mach48 42  34
setRxphyCalEntry mach48 43  46
setRxphyCalEntry mach48 44  11
setRxphyCalEntry mach48 45  23
setRxphyCalEntry mach48 46  35
setRxphyCalEntry mach48 47  47

#####################################################################

#####################################################################
puts "Setting up the SDQ"
#        iSlot  iSdqOffset  iPhyID iFifoNum iBlockPtr iFifoBS iPosSelect
iEnable iDT iBT
#RXSDQ
sdqSetup mach48 0x50   0    0    0   1   0   1   3   3
sdqSetup mach48 0x50   1    1    6   1   0   1   3   3
sdqSetup mach48 0x50   2    2   12   1   0   1   3   3
sdqSetup mach48 0x50   3    3   18   1   0   1   3   3
sdqSetup mach48 0x50   4    4    2   1   0   1   3   3
sdqSetup mach48 0x50   5    5    8   1   0   1   3   3
sdqSetup mach48 0x50   6    6   14   1   0   1   3   3
sdqSetup mach48 0x50   7    7   20   1   0   1   3   3
sdqSetup mach48 0x50   8    8    4   1   0   1   3   3
sdqSetup mach48 0x50   9    9   10   1   0   1   3   3
sdqSetup mach48 0x50   10  10   16   1   0   1   3   3
sdqSetup mach48 0x50   11  11   22   1   0   1   3   3
sdqSetup mach48 0x50   12  16    0   1   0   1   3   3
sdqSetup mach48 0x50   13  17    6   1   0   1   3   3
sdqSetup mach48 0x50   14  18   12   1   0   1   3   3
sdqSetup mach48 0x50   15  19   18   1   0   1   3   3
sdqSetup mach48 0x50   16  20    2   1   0   1   3   3
sdqSetup mach48 0x50   17  21    8   1   0   1   3   3
sdqSetup mach48 0x50   18  22   14   1   0   1   3   3
sdqSetup mach48 0x50   19  23   20   1   0   1   3   3
sdqSetup mach48 0x50   20  24    4   1   0   1   3   3
sdqSetup mach48 0x50   21  25   10   1   0   1   3   3
sdqSetup mach48 0x50   22  26   16   1   0   1   3   3
sdqSetup mach48 0x50   23  27   22   1   0   1   3   3
sdqSetup mach48 0x50   24  32    0   1   0   1   3   3
sdqSetup mach48 0x50   25  33    6   1   0   1   3   3
sdqSetup mach48 0x50   26  34   12   1   0   1   3   3
sdqSetup mach48 0x50   27   35   18   1   0   1   3   3
```

```
sdqSetup mach48 0x50   28   36    2   1   0   1   3   3
sdqSetup mach48 0x50   29   37    8   1   0   1   3   3
sdqSetup mach48 0x50   30   38   14   1   0   1   3   3
sdqSetup mach48 0x50   31   39   20   1   0   1   3   3
sdqSetup mach48 0x50   32   40    4   1   0   1   3   3
sdqSetup mach48 0x50   33   41   10   1   0   1   3   3
sdqSetup mach48 0x50   34   42   16   1   0   1   3   3
sdqSetup mach48 0x50   35   43   22   1   0   1   3   3
sdqSetup mach48 0x50   36   48    0   1   0   1   3   3
sdqSetup mach48 0x50   37   49    6   1   0   1   3   3
sdqSetup mach48 0x50   38   50   12   1   0   1   3   3
sdqSetup mach48 0x50   39   51   18   1   0   1   3   3
sdqSetup mach48 0x50   40   52    2   1   0   1   3   3
sdqSetup mach48 0x50   41   53    8   1   0   1   3   3
sdqSetup mach48 0x50   42   54   14   1   0   1   3   3
sdqSetup mach48 0x50   43   55   20   1   0   1   3   3
sdqSetup mach48 0x50   44   56    4   1   0   1   3   3
sdqSetup mach48 0x50   45   57   10   1   0   1   3   3
sdqSetup mach48 0x50   46   58   16   1   0   1   3   3
sdqSetup mach48 0x50   47   59   22   1   0   1   3   3

#####################################################################
#        iSlot  iSdqOffset  iPhyID iFifoNum iBlockPtr iFifoBS iPosSelect
#iEnable iDT iBT
#TXSDQ
sdqSetup mach48 0x60   0    0    0   1   0   1   3   3
sdqSetup mach48 0x60   1    1    6   1   0   1   3   3
sdqSetup mach48 0x60   2    2   12   1   0   1   3   3
sdqSetup mach48 0x60   3    3   18   1   0   1   3   3
sdqSetup mach48 0x60   4    4    2   1   0   1   3   3
sdqSetup mach48 0x60   5    5    8   1   0   1   3   3
sdqSetup mach48 0x60   6    6   14   1   0   1   3   3
sdqSetup mach48 0x60   7    7   20   1   0   1   3   3
sdqSetup mach48 0x60   8    8    4   1   0   1   3   3
sdqSetup mach48 0x60   9    9   10   1   0   1   3   3
sdqSetup mach48 0x60   10  10   16   1   0   1   3   3
sdqSetup mach48 0x60   11  11   22   1   0   1   3   3
sdqSetup mach48 0x60   12  16    0   1   0   1   3   3
sdqSetup mach48 0x60   13  17    6   1   0   1   3   3
sdqSetup mach48 0x60   14  18   12   1   0   1   3   3
sdqSetup mach48 0x60   15  19   18   1   0   1   3   3
sdqSetup mach48 0x60   16  20    2   1   0   1   3   3
sdqSetup mach48 0x60   17  21    8   1   0   1   3   3
sdqSetup mach48 0x60   18  22   14   1   0   1   3   3
sdqSetup mach48 0x60   19  23   20   1   0   1   3   3
sdqSetup mach48 0x60   20  24    4   1   0   1   3   3
sdqSetup mach48 0x60   21  25   10   1   0   1   3   3
```

```
sdqSetup mach48 0x60   22  26   16   1   0   1   3   3
sdqSetup mach48 0x60   23  27   22   1   0   1   3   3
sdqSetup mach48 0x60   24  32    0   1   0   1   3   3
sdqSetup mach48 0x60   25  33    6   1   0   1   3   3
sdqSetup mach48 0x60   26  34   12   1   0   1   3   3
sdqSetup mach48 0x60   27   35   18   1   0   1   3   3
sdqSetup mach48 0x60   28   36    2   1   0   1   3   3
sdqSetup mach48 0x60   29   37    8   1   0   1   3   3
sdqSetup mach48 0x60   30   38   14   1   0   1   3   3
sdqSetup mach48 0x60   31   39   20   1   0   1   3   3
sdqSetup mach48 0x60   32   40    4   1   0   1   3   3
sdqSetup mach48 0x60   33   41   10   1   0   1   3   3
sdqSetup mach48 0x60   34   42   16   1   0   1   3   3
sdqSetup mach48 0x60   35   43   22   1   0   1   3   3
sdqSetup mach48 0x60   36   48    0   1   0   1   3   3
sdqSetup mach48 0x60   37   49    6   1   0   1   3   3
sdqSetup mach48 0x60   38   50   12   1   0   1   3   3
sdqSetup mach48 0x60   39   51   18   1   0   1   3   3
sdqSetup mach48 0x60   40   52    2   1   0   1   3   3
sdqSetup mach48 0x60   41   53    8   1   0   1   3   3
sdqSetup mach48 0x60   42   54   14   1   0   1   3   3
sdqSetup mach48 0x60   43   55   20   1   0   1   3   3
sdqSetup mach48 0x60   44   56    4   1   0   1   3   3
sdqSetup mach48 0x60   45   57   10   1   0   1   3   3
sdqSetup mach48 0x60   46   58   16   1   0   1   3   3
sdqSetup mach48 0x60   47   59   22   1   0   1   3   3

#####################################################################
#Setup RTDP on Chan 0-11,12-23,24-35,36-47
puts "Setting up RTDP"
rtdpSetup mach48 0   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 1   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 2   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 3   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 4   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 5   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 6   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 7   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 8   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 9   0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 10  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 11  0x011b 0x0402 0x0300 0x0168 0x01FF

rtdpSetup mach48 12  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 13  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 14  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 15  0x011b 0x0402 0x0300 0x0168 0x01FF
```

```
rtdpSetup mach48 16  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 17  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 18  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 19  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 20  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 21  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 22  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 23  0x011b 0x0402 0x0300 0x0168 0x01FF

rtdpSetup mach48 24  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 25  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 26  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 27  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 28  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 29  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 30  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 31  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 32  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 33  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 34  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 35  0x011b 0x0402 0x0300 0x0168 0x01FF

rtdpSetup mach48 36  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 37  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 38  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 39  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 40  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 41  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 42  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 43  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 44  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 45  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 46  0x011b 0x0402 0x0300 0x0168 0x01FF
rtdpSetup mach48 47  0x011b 0x0402 0x0300 0x0168 0x01FF

#########################################################

#Receive Timeslot Registers (0x0002->0x0007)
puts "Setting up timeslot registers"
write mach48 0x0002 0xaaaa
write mach48 0x0003 0xaaaa
write mach48 0x0004 0xaaaa
write mach48 0x0005 0xaaaa
write mach48 0x0006 0xaaaa
write mach48 0x0007 0xaaaa
#Transmit Timeslot Registers (0x0008->0x000D)
write mach48 0x0008 0xaaaa
```

```
write mach48 0x0009 0xaaaa
write mach48 0x000a 0xaaaa
write mach48 0x000b 0xaaaa
write mach48 0x000c 0xaaaa
write mach48 0x000d 0xaaaa
################################################
#Setup RCAS for 48xSTS-1 channels
puts "Setting up RCAS_0"
write mach48 0x1b2 0x0050
write mach48 0x1b3 0x0051
write mach48 0x1b4 0x0052
write mach48 0x1b5 0x0053
write mach48 0x1b6 0x0054
write mach48 0x1b7 0x0055
write mach48 0x1b8 0x0056
write mach48 0x1b9 0x0057
write mach48 0x1ba 0x0058
write mach48 0x1bb 0x0059
write mach48 0x1bc 0x005a
write mach48 0x1bd 0x005b
puts "Setting up RCAS_1"
write mach48 0x1c2 0x0050
write mach48 0x1c3 0x0051
write mach48 0x1c4 0x0052
write mach48 0x1c5 0x0053
write mach48 0x1c6 0x0054
write mach48 0x1c7 0x0055
write mach48 0x1c8 0x0056
write mach48 0x1c9 0x0057
write mach48 0x1ca 0x0058
write mach48 0x1cb 0x0059
write mach48 0x1cc 0x005a
write mach48 0x1cd 0x005b
puts "Setting up RCAS_2"
write mach48 0x1d2 0x0050
write mach48 0x1d3 0x0051
write mach48 0x1d4 0x0052
write mach48 0x1d5 0x0053
write mach48 0x1d6 0x0054
write mach48 0x1d7 0x0055
write mach48 0x1d8 0x0056
write mach48 0x1d9 0x0057
write mach48 0x1da 0x0058
write mach48 0x1db 0x0059
write mach48 0x1dc 0x005a
write mach48 0x1dd 0x005b
puts "Setting up RCAS_3"
```

```
write mach48 0x1e2 0x0050
write mach48 0x1e3 0x0051
write mach48 0x1e4 0x0052
write mach48 0x1e5 0x0053
write mach48 0x1e6 0x0054
write mach48 0x1e7 0x0055
write mach48 0x1e8 0x0056
write mach48 0x1e9 0x0057
write mach48 0x1ea 0x0058
write mach48 0x1eb 0x0059
write mach48 0x1ec 0x005a
write mach48 0x1ed 0x005b
##################################################
puts "Setting up the SIRP"
write mach48 0x14c0 0x0007
write mach48 0x14c1 0x0007
write mach48 0x14c2 0x0007
write mach48 0x14c3 0x0007
write mach48 0x14c4 0x0007
write mach48 0x14c5 0x0007
write mach48 0x14c6 0x0007
write mach48 0x14c7 0x0007
write mach48 0x14c8 0x0007
write mach48 0x14c9 0x0007
write mach48 0x14ca 0x0007
write mach48 0x14cb 0x0007

write mach48 0x14d0 0x0007
write mach48 0x14d1 0x0007
write mach48 0x14d2 0x0007
write mach48 0x14d3 0x0007
write mach48 0x14d4 0x0007
write mach48 0x14d5 0x0007
write mach48 0x14d6 0x0007
write mach48 0x14d7 0x0007
write mach48 0x14d8 0x0007
write mach48 0x14d9 0x0007
write mach48 0x14da 0x0007
write mach48 0x14db 0x0007

write mach48 0x14e0 0x0007
write mach48 0x14e1 0x0007
write mach48 0x14e2 0x0007
write mach48 0x14e3 0x0007
write mach48 0x14e4 0x0007
write mach48 0x14e5 0x0007
write mach48 0x14e6 0x0007
```

```
write mach48 0x14e7 0x0007
write mach48 0x14e8 0x0007
write mach48 0x14e9 0x0007
write mach48 0x14ea 0x0007
write mach48 0x14eb 0x0007

write mach48 0x14f0 0x0007
write mach48 0x14f1 0x0007
write mach48 0x14f2 0x0007
write mach48 0x14f3 0x0007
write mach48 0x14f4 0x0007
write mach48 0x14f5 0x0007
write mach48 0x14f6 0x0007
write mach48 0x14f7 0x0007
write mach48 0x14f8 0x0007
write mach48 0x14f9 0x0007
write mach48 0x14fa 0x0007
write mach48 0x14fb 0x0007


##############################################
#Enable BLOCKS:SDQs,PHYs,RCAS

puts "Enable TX,RX SDQs"
write mach48 0x60 0
write mach48 0x50 0

puts "Enable TX,RX Phys"
write mach48 0x48 0x00c0
write mach48 0x40 0

puts "Enabling RCAS"
write mach48 0x1b0 0
write mach48 0x1c0 0
write mach48 0x1d0 0
write mach48 0x1e0 0

#####################################################
TSEinit
mapTSE3 tse1
portEnable4
portEnable5

#####################################################
#Setup Delays for the system
puts "setting up system delays"
after 1000
```

```
write mach48 0x11 127
write mach48 0x13 30
write tse1 0x40 60
write tbs5 0x5 127

center

#Turn off frame pulse in MACH and SP155
puts "Switching off SP155 and MACH FP"
write fpga 0x40 0

puts "Enabling TCAS"
after 1000
write mach48 0x1f0 0
write mach48 0x200 0
write mach48 0x210 0
write mach48 0x220  0

delay
```

NOTES

CONTACTING PMC-SIERRA, INC.

PMC-Sierra, Inc.
105-8555 Baxter Place Burnaby, BC
Canada V5A 4V7

Tel: (604) 415-6000

Fax: (604) 415-6200

Document Information: document@pmc-sierra.com
Corporate Information: info@pmc-sierra.com
Application Information: apps@pmc-sierra.com
(604) 415-4533
Web Site: http://www.pmc-sierra.com

PMC-2001532 (P2) ref PMC-1990823(P3) Issue date: Oct 2000