VSC9670

Scalable Architecture Framing Engine for T1

Data Book Revision 4.0

VITESSE
SEMICONDUCTOR CORPORATION

VSC9670

Scalable Architecture Framing Engine for T1

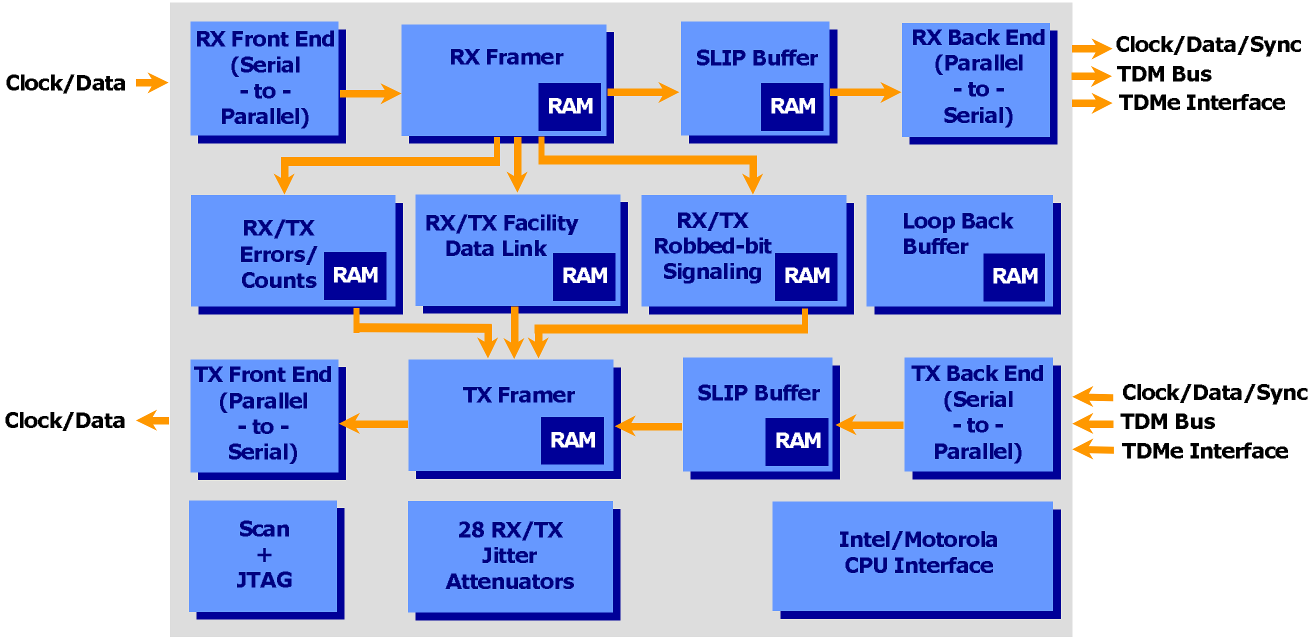
Overview

The Scalable Architecture Framing Engine for T1, the VSC9670, provides 28 channels of T1 framing, allowing for full channelized T3 processing. Typical applications include: Internet Service Provider (ISP) or central office Frame Relay/ATM switches, high density T1 connections for multiplexers, switches, routers, and digital private branch exchanges (PBX), Network Service Provider (NSP) infrastructure remote access concentrators and TDM/Sonet access concentrators or Add-Drop Multiplexers. The scalable framer uses a shared state machine approach, with each channel state stored in RAM. In the RX direction, each one of the 28 T1 bit streams is converted into an internal 8-bit wide data path, and a single shared RX state machine performs framing and assists in OAM and statistics function. Similarly in the TX direction, the internal data path is widened to 8 bits, and a shared TX state machine is used. A third shared state machine is provided for errors and statistics. RX data is input from (or TX data is output to) the line interface side to or from the VSC9670 on a clock/data interface. On the system side, RX data can be output to (or TX data can be input from) a clock/data/sync interface, or RX data can be connected with a low pin-count interface to an HDLC, PPP, ATM delineator or a TDM backplane.

Features

- A variety of loopbacks (to DS1 or DS0 resolution)
- Direct connection to system-side TDM buses at 2.048 MHz, 4.096 MHz or 8.192 MHz, and provides a data-optimized, low pin count, muxed interface
- Simple (Intel / Moto / generic muxed) 8-bit processor interface
- Performance monitoring, facility data link (FDL), robbed bit signaling, and slip buffers
- Optimized system-side interface for packet or cell data, and connects to fast TDM backplanes
- Jitter attenuation in both receive and transmit directions
- DS1 or DS0 compliant line/local loopbacks and per link diagnostic
- Interfaces seamlessly to the Vitesse Semiconductor VSC9680 single chip solution for Packet, Cell, H.100, and T3 interfaces
- Available in a 456 pin PBGA+ package, and uses 3.3V CMOS technology, with 5V tolerant I/O pads.

VSC9670 Block Diagram



Revision Information

Revision	Vitesse Document Number	History
3.0	G56049-0	VSC9670 Databook, Advanced Product Information
4.0	G56049-0	Full release; Revise package drawing, Figure 5.1

Copyright © 2001 Vitesse Semiconductor Corporation

Print date: July 2001

Vitesse Semiconductor Corporation retains the right to make change to its products or specifications to improve performance, reliability or manufacturability. While the information furnished herein is held to be accurate and reliable, no responsibility will be assumed by Vitesse for its use. Furthermore, the information contained herein does not convey to the purchaser of microelectronic devices any license under the patent right of any manufacturer.

Vitesse Semiconductor Corporation products are not intended for use in life support products where failure of the Vitesse Semiconductor Corporation product could reasonably be expected to result in death or personal injury. Anyone using a Vitesse Semiconductor Corporation product in such an application without the express written consent of an officer of Vitesse Semiconductor Corporation does so at their own risk, and agrees to fully indemnify Vitesse Semiconductor Corporation for any damages that may result from such use or sale.

Vitesse is a registered trademark and H-GaAs is a trademark of Vitesse Semiconductor Corporation. All other product or service names used in this publication are for identification purposes only, and may be trademarks or registered trademarks of their respective companies. All other trademarks or registered trademarks mentioned herein are the property of their respective holders.

Specifications herein are subject to change without notice and Vitesse Semiconductor Corporation is not responsible for any damages caused by such changes.

PRINTED IN THE U.S.A.

Table of Contents

Section 1 Overview of the VSC9670 1

Section 2 Functional Description 30

List of Figures

List of Tables

Section 1 Overview of the VSC9670 1

Section 2 Functional Description 30

Section 3 Register Map 78

1.0 Overview of the VSC9670

The Scalable Architecture Framing Engine for T1, the VSC9670, provides 28 channels of T1 framing, allowing for full channelized T3 processing. The highly integrated VSC9670 is designed for high channelization, high density points in the access and switching infrastructure, such as access concentrators, routers and switches. Typical applications include:

- Internet Service Provider (ISP) or central office Frame Relay / ATM switches
- High density T1 connections for multiplexers, switches, routers, and digital private branch exchanges (PBX)
- Network Service Provider (NSP) infrastructure remote access concentrators
- TDM / Sonet access concentrators or Add-Drop Multiplexers

The scalable framer uses a shared state machine approach, with each channel state stored in RAM. In the RX direction, each one of the 28 T1 bit streams is converted into an internal 8-bit wide data path, and a single shared RX state machine performs framing and assists in OAM and statistics function. Similarly in the TX direction, the internal data path is widened to 8 bits, and a shared TX state machine is used. A third shared state machine is provided for errors and statistics. There are additional shared machines and associated RAMs for FDL (Facility Data Link) handling, robbed bit signaling, and so on. Shared RAM is also used for slip buffering.

The shared state machine and shared RAM approach reduces gate count and provides a clear scalable path to higher densities. State storage is in on-chip SRAM, instead of flops.

RX data are input from (or TX data are output to) the line interface side to or from the VSC9670 on a clock/data interface. On the system side, RX data can be output to (or TX data can be input from) a clock/data/sync interface, or RX data can be connected with low pin-count to an HDLC, PPP, ATM delineator or a TDM backplane. The optimized system-side interface for packet or cell data applications connects directly to fast 2.048 MHz, 4.096 MHz or 8.192 MHz TDM backplanes.

The VSC9670 includes these features:

- A variety of loopbacks (to DS1 or DS0 resolution)
- Direct connection to system-side TDM buses at 2.048 MHz, 4.096 MHz or 8.192 MHz, and provides a data-optimized, low pin count, muxed interface
- Simple (Intel / Moto / generic muxed) 8-bit processor interface

The VSC9670 also provides:

- Performance monitoring, facility data link (FDL), robbed bit signaling, and slip buffers
- Optimized system-side interface for packet or cell data, and connects to fast TDM backplanes
- Jitter attenuation in both receive and transmit directions
- DS1 or DS0 compliant line/local loopbacks and per link diagnostic

The VSC9670 includes a simple 8-bit Intel/Motorola CPU interface for control and sensing, and for FDL data I/O. This device interfaces seamlessly to the Vitesse Semiconductor VSC9680 single chip for Packet, Cell, H.100, and T3 interfaces.

The VSC9670 is available in a 456 pin EPBGA, and uses 3.3V CMOS technology, with 5V tolerant I/O pads.

1.1 Block Diagram

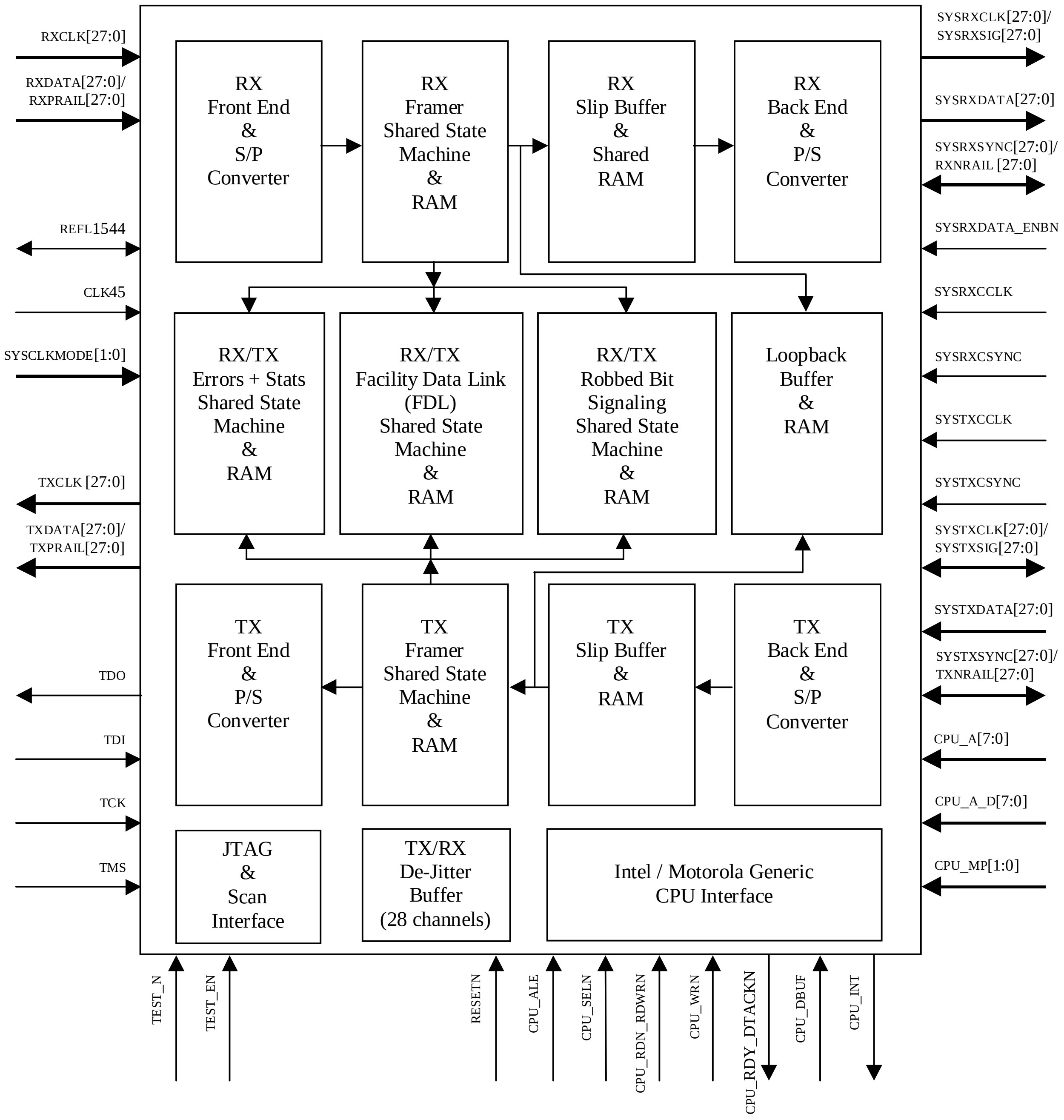


Figure 1.1 VSC9670 Block Diagram

1.2 Features

Two separate register bank interfaces provide fast context switching support for User and ISR modes of operation.

Each receiver section in each one of the 28 DS1s performs these functions:

- Accepts a 2-pin clock and data inputs from LIU and provides clock jitter attenuation
- Frames data from SF and ESF (including CRC-6 checking); supports Japanese J1 format
- Provides LOS, OOF, Red, SF Yellow, ESF Yellow and AIS alarms
- Provides the following performance monitoring statistics with 1-second sized counters:
    - ESF CRC-6 error counts
    - Framing errors
    - LOF or COFA (change of framing alignment)

- Detects in-band codes such as loop-up, loop-down, and idle
- Independently detects pseudo-random patterns, configurable for common BERT and repetitive pattern detection, and selectable on a per-DS0 basis basis; supports Fractional and Full T1 BERT
- Terminates the ESF Facility Data Link (Bit Oriented Codes and HDLC delineation, with CRC checking) into a 128-byte FIFO, with a flexible CPU interface.
- Extracts and buffers robbed bit signaling (with de-bounce) to the CPU, and provides optional interrupts on state change
- Provides 256 entry Rx signaling bit FIFO to quickly transfer signaling bit data to the CPU
- Provides optional per-DS0 data force using a CPU-written byte (uses slip buffer)
- Each of the 28 internal Rx line-side clocks can be muxed to an external pin
- Provides a dedicated slip buffer for decoupling incoming timing from the backplane, or outputs clock / data / sync
- In slip buffer mode, line-side Pdata, Ndata, and clock interfaces are available along with a system side signaling bus interface

Each transmitter section in each one of the 28 DS1s performs these functions:

- Outputs clock and data to the LIU or M13
- Frames data to SF, ESF (includes CRC-6 generation); supports Japanese J1 format
- Supports transmission of Yellow and AIS alarm signals
- Supports transmission of in-band codes for loop-up, loop-down and idle code
- Provides various force functions at DS0 resolution, including 1's density control
- Generates pseudo-random patterns independently, configurable for common BERT and repetitive patterns, and selectable on a per-DS0 basis; supports Full and Fractional T1 BERT
- Inserts ESF Facility Data Link (Bit Oriented Codes and HDLC packet with CRC generation) with a flexible 128 byte FIFO interface to the CPU
- Provides for CPU insertion of robbed bit signaling on a per-DS0 selectable basis
- Provides for CPU write of per-DS0 force data, using slip buffer
- Provides a dedicated slip buffer for decoupling backplane timing from line timing

1.3 References

- AT&T Publication TR-62411-Accunet T1.5 Service Description and Interface Specification December, 1990
- AT&T Publication TR-54106-Requirements for Interfacing Digital Terminal Equipment to Services Employing the Extended Super Frame, September, 1989
- AT&T Publication TR-43801-Channel Bank Requirements and Objectives, September, 1982
- ANSI T1.403-1995, Network-to-Customer Installation - DS1 Metallic Interface.
- AT&T Publication TR-303 Integrated Digital Loop Carrier System Generic Requirements 1995

1.4 Application Examples

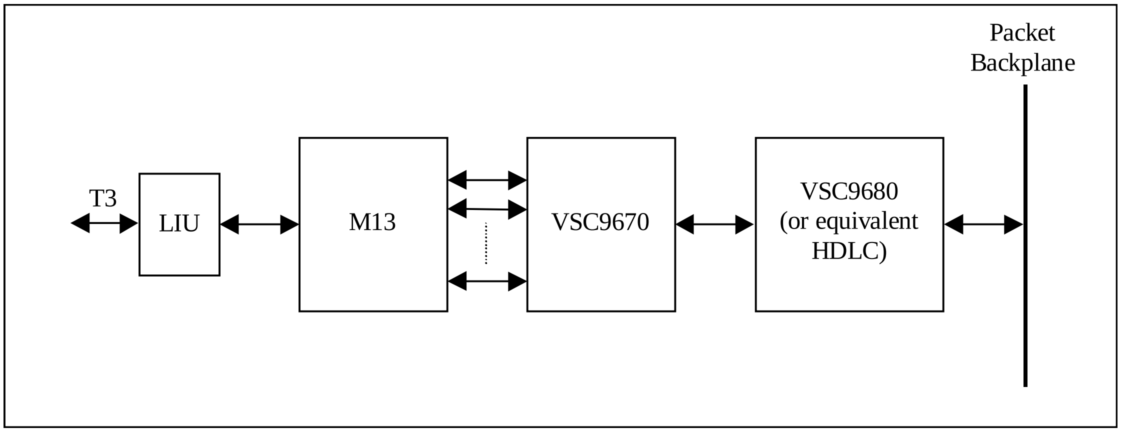


Figure 1.2 Terminating Single Channelized T3 – Frame Relay Switch

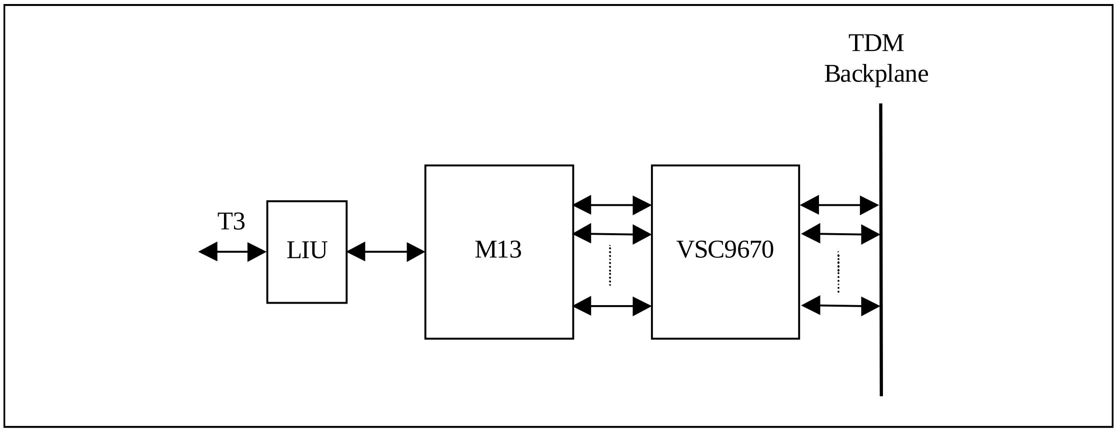


Figure 1.3 Terminating Single Channelized T3 – TDM Backplane Architecture

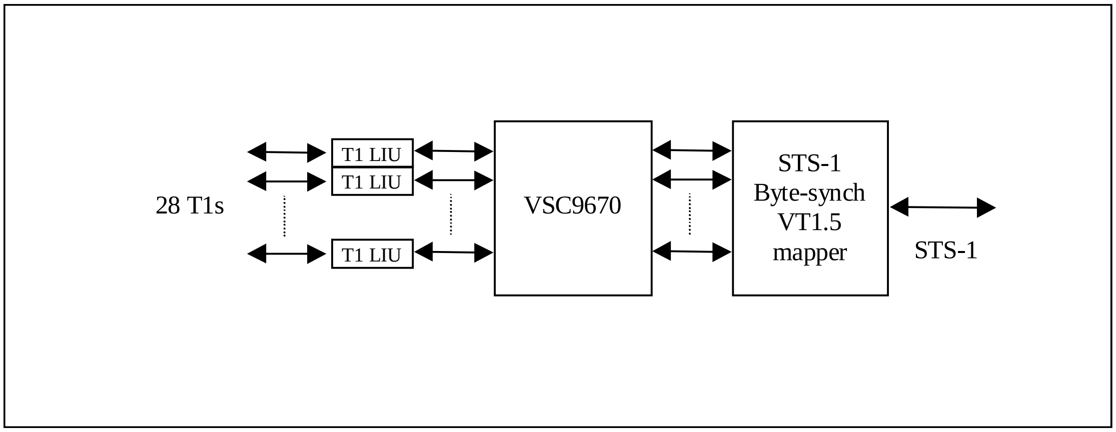


Figure 1.4 Back-Hauling Multiple T1s – Sonet Application

1.5 Pin Diagrams

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
AF	Vdd_IO	systx_data22	systx_clk22	systxsync19	VSS	systxsync17	systx_clk16	systxsync14	VSS	systx_data12	systx_data11	systx_data10	VSS	VSS	systxsync8	systx_clk7	systx_clk6	VSS	systx_data4	systxsync2	systxsync0	VSS	sysclkmode0	cpu_ale	sysrx_data27	Vdd_Core	AF
AE	systx_clk23	VSS	systxsync22	systx_data20	systx_clk18	Vdd_IO	systx_clk17	systx_data15	systx_clk14	Vdd_IO	systx_clk12	systx_clk11	systx_data9	systxsync9	systx_clk8	systx_data6	Vdd_Core	systx_clk5	systx_clk3	systx_clk1	Vdd_Core	sysrx_data4	sysrx_data3	sysrx_data_enbn	VSS	sysrxsync27	AE
AD	systx_data23	systxsync23	Vdd_IO	systx_clk21	systx_data19	systxsync18	VSS	systx_data16	systxsync15	systxsync13	VSS	systxsync11	systx_clk10	systx_clk9	systxsync7	VSS	systxsync5	systxsync3	systxsync1	VSS	systx_csync	sysrx_data0	cpu_wrn	Vdd_IO	sysrx_clk27	sysrx_data26	AD
AC	systxsync26	systx_clk25	systx_data24	VSS	systxsync21	systx_clk20	systx_data18	Vdd_IO	systxsync16	systx_data14	systx_clk13	Vdd_IO	systxsync10	systx_data8	Vdd_Core	systx_data5	systxsync4	systx_clk2	Vdd_Core	systx_cclk	sysrx_data1	cpu_rdn_rdwrn	VSS	sysrx_data25	sysrxsync25	sysrx_clk24	AC
AB	VSS	systx_data27	systx_clk26	systxsync24	Vdd_IO	systx_data21	systxsync20	systx_clk19	systx_data17	systx_clk15	systx_data13	systxsync12	Vcc_Bias	systx_data7	systxsync6	Vdd_IO	systx_clk4	systx_clk0	sysclkmode1	sysrx_data2	cpu_seln	Vdd_Core	sysrx_clk26	sysrxsync24	sysrx_data22	VSS	AB
AA	systx_data2	Vdd_IO	systxsync27	systx_data25	systx_clk24																	sysrxsync26	sysrx_data24	sysrx_clk23	Vdd_IO	sysrxsync22	AA
Y	tx_data0	systx_data1	VSS	systx_clk27	systxsync25																	sysrx_clk25	sysrxsync23	VSS	sysrx_clk22	sysrx_clk21	Y
W	tx_clk3	tx_clk1	systx_data0	Vdd_IO	systx_data26																	sysrx_data23	Vdd_IO	sysrx_data21	sysrx_data20	sysrxsync19	W
V	VSS	tx_data3	tx_data1	tx_clk0	systx_data3																	Vcc_Bias	sysrxsync21	sysrxsync20	sysrx_clk19	VSS	V
U	tx_data5	Vdd_IO	tx_data4	tx_data2	tx_clk2																	sysrx_clk20	sysrx_data19	sysrxsync18	Vdd_IO	sysrx_data17	U
T	tx_clk7	tx_data6	VSS	tx_clk5	tx_clk4						VSS	VSS	VSS	VSS	VSS	VSS						sysrx_data18	sysrx_clk18	VSS	sysrx_clk17	sysrx_data16	T
R	tx_clk9	tx_data8	tx_data7	Vdd_IO	tx_clk6						VSS	VSS	VSS	VSS	VSS	VSS						sysrxsync17	Vdd_IO	sysrxsync16	sysrx_data15	sysrxsync15	R
P	VSS	tx_data10	tx_clk10	tx_data9	tx_clk8						VSS	VSS	VSS	VSS	VSS	VSS						sysrx_clk16	sysrx_clk15	sysrx_data14	sysrxsync14	VSS	P
N	VSS	tx_clk11	tx_data11	tx_clk12	tx_data13						VSS	VSS	VSS	VSS	VSS	VSS						sysrxsync12	sysrxsync13	sysrx_data13	sysrx_clk14	VSS	N
M	tx_data12	tx_clk13	tx_clk14	Vdd_IO	tx_data15						VSS	VSS	VSS	VSS	VSS	VSS						sysrx_clk11	Vdd_IO	sysrx_clk12	sysrx_data12	sysrx_clk13	M
L	tx_data14	tx_clk15	VSS	tx_data16	Vcc_Bias						VSS	VSS	VSS	VSS	VSS	VSS						sysrx_data9	sysrxsync10	VSS	sysrxsync11	sysrx_data11	L
K	tx_clk16	Vdd_IO	tx_clk17	tx_data18	tx_clk19																	sysrxsync8	sysrx_data8	sysrx_clk10	Vdd_IO	sysrx_data10	K
J	VSS	tx_data17	tx_data19	tx_clk21	tx_clk23																	Vcc_Bias	sysrx_clk7	sysrx_clk8	sysrxsync9	VSS	J
H	tx_clk18	tx_clk20	tx_data21	Vdd_IO	tx_clk25																	sysrxsync4	Vdd_IO	sysrx_data6	sysrx_data7	sysrx_clk9	H
G	tx_data20	tx_clk22	VSS	tx_data24	tx_clk27																	sysrxsync2	sysrx_clk5	VSS	sysrxsync6	sysrxsync7	G
F	tx_data22	Vdd_IO	tx_clk24	tx_data26	rx_clk1																	sysrxsync0	sysrx_clk3	sysrxsync5	Vdd_IO	sysrx_clk6	F
E	VSS	tx_data23	tx_clk26	rx_data0	Vdd_Core	rx_data4	rx_data6	rx_data8	rx_data10	rx_data14	VSS	rx_clk18	Vcc_Bias	rx_data24	rx_data26	TDI	TDO	cpu_a1	cpu_a5	cpu_a_d0	cpu_a_d4	Vdd_IO	sysrx_clk1	sysrxsync3	sysrx_data5	VSS	E
D	tx_data25	tx_data27	rx_clk0	VSS	rx_clk5	rx_clk7	rx_clk9	Vdd_Core	rx_data12	rx_clk15	rx_clk17	Vdd_Core	clk45	rx_clk23	Vdd_IO	rx_data27	TRST_N	cpu_dbuf	Vdd_IO	cpu_a4	cpu_rdy_dtackn	cpu_a_d3	VSS	sysrxsync1	sysrx_clk2	sysrx_clk4	D
C	rx_data1	rx_clk2	Vdd_IO	rx_data5	rx_data7	rx_data9	VSS	rx_clk12	rx_clk14	rx_data16	VSS	rx_data19	rx_clk21	rx_data22	rx_clk25	VSS	TEST_EN	resetn	cpu_mp0	VSS	cpu_a3	cpu_a7	cpu_a_d2	Vdd_IO	sysrx_csync	sysrx_clk0	C
B	rx_data2	VSS	rx_data3	rx_clk6	rx_clk10	Vdd_Core	rx_data11	rx_data13	rx_clk16	Vdd_Core	rx_data18	rx_clk20	rx_data21	rx_clk22	rx_clk24	rx_clk26	Vdd_IO	TCK	ref_1544	cpu_mp1	Vdd_IO	cpu_a2	cpu_a_d1	cpu_a_d6	VSS	sysrx_cclk	B
A	Vdd_Core	rx_clk3	rx_clk4	rx_clk8	VSS	rx_clk11	rx_clk13	rx_data15	VSS	rx_data17	rx_clk19	rx_data20	VSS	VSS	rx_data23	rx_data25	rx_clk27	VSS	TMS	cpu_int	cpu_a0	VSS	cpu_a6	cpu_a_d5	cpu_a_d7	Vdd_IO	A
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	

VSC9670
456-PIN EPBGA
(bottom view)

135 Power/ground Pins
173 System Side Interface Pins
112 Line Side Interface Pins
36 General, Microprocessor &
JTAG/SCAN Interface Pins

Figure 1.5 Pin Diagram - Power/Ground and All I/O Signals

VSC9670
456-PIN EPBGA
(bottom view)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
AF																							sysclk mode0	cpu_ale		
AE																										
AD																							cpu_wrn			
AC																						cpu_rdn_rdwrn				
AB																			sysclk mode1		cpu_seln					
AA																										
Y																										
W																										
V																										
U																										
T																										
R																										
P																										
N																										
M																										
L																										
K																										
J																										
H																										
G																										
F																										
E																TDI	TDO	cpu_a1	cpu_a5	cpu_a_d0	cpu_a_d4					
D													clk45				TRST_N	cpu_dbuf		cpu_a4	cpu_rdy_dtackn	cpu_a_d3				
C																	TEST_EN	resetn	cpu_mp0		cpu_a3	cpu_a7	cpu_a_d2			
B																		TCK	ref_1544	cpu_mp1		cpu_a2	cpu_a_d1	cpu_a_d6		
A																			TMS	cpu_int	cpu_a0		cpu_a6	cpu_a_d5	cpu_a_d7	

5 General Signal Pins
25 Microprocessor Interface Pins
6 JTAG/SCAN Interface Pins

Figure 1.6 Pin Diagram - General, Microprocessor & JTAG/SCAN

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
AF																											AF
AE																											AE
AD																											AD
AC																											AC
AB																											AB
AA																											AA
Y	tx_data0																										Y
W	tx_clk3	tx_clk1																									W
V		tx_data3	tx_data1	tx_clk0																							V
U	tx_data5		tx_data4	tx_data2	tx_clk2																						U
T	tx_clk7	tx_data6		tx_clk5	tx_clk4																						T
R	tx_clk9	tx_data8	tx_data7		tx_clk6																						R
P		tx_data10	tx_clk10	tx_data9	tx_clk8																						P
N		tx_clk11	tx_data11	tx_clk12	tx_data13																						N
M	tx_data12	tx_clk13	tx_clk14		tx_data15																						M
L	tx_data14	tx_clk15		tx_data16																							L
K	tx_clk16		tx_clk17	tx_data18	tx_clk19																						K
J		tx_data17	tx_data19	tx_clk21	tx_clk23																						J
H	tx_clk18	tx_clk20	tx_data21		tx_clk25																						H
G	tx_data20	tx_clk22		tx_data24	tx_clk27																						G
F	tx_data22		tx_clk24	tx_data26	rx_clk1																						F
E		tx_data23	tx_clk26	rx_data0		rx_data4	rx_data6	rx_data8	rx_data10	rx_data14		rx_clk18		rx_data24	rx_data26												E
D	tx_data25	tx_data27	rx_clk0		rx_clk5	rx_clk7	rx_clk9		rx_data12	rx_clk15	rx_clk17			rx_clk23		rx_data27											D
C	rx_data1	rx_clk2		rx_data5	rx_data7	rx_data9		rx_clk12	rx_clk14	rx_data16		rx_data19	rx_clk21	rx_data22	rx_clk25												C
B	rx_data2		rx_data3	rx_clk6	rx_clk10		rx_data11	rx_data13	rx_clk16		rx_data18	rx_clk20	rx_data21	rx_clk22	rx_clk24	rx_clk26											B
A		rx_clk3	rx_clk4	rx_clk8		rx_clk11	rx_clk13	rx_data15		rx_data17	rx_clk19	rx_data20			rx_data23	rx_data25	rx_clk27										A
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	

VSC9670
456-PIN EPBGA
(bottom view)

56 Line Side Receive Interface Pins
56 Line Side Transmit Interface Pins

Figure 1.7 Pin Diagram - Line Side Interface

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
AF		systx_data22	systx_clk22	systxsync19		systxsync17	systx_clk16	systxsync14		systx_data12	systx_data11	systx_data10			systxsync8	systx_clk7	systx_clk6		systx_data4	systxsync2	systxsync0				sysrx_data27		**AF**
AE	systx_clk23		systxsync22	systx_data20	systx_clk18		systx_clk17	systx_data15	systx_clk14		systx_clk12	systx_clk11	systx_data9	systxsync9	systx_clk8	systx_data6		systx_clk5	systx_clk3	systx_clk1		sysrx_data4	sysrx_data3	sysrx_data_enbn		sysrxsync27	**AE**
AD	systx_data23	systxsync23		systx_clk21	systx_data19	systxsync18		systx_data16	systxsync15	systxsync13		systxsync11	systx_clk10	systx_clk9	systxsync7		systxsync5	systxsync3	systxsync1		systx_csync	sysrx_data0			sysrx_clk27	sysrx_data26	**AD**
AC	systxsync26	systx_clk25	systx_data24		systxsync21	systx_clk20	systx_data18		systxsync16	systx_data14	systx_clk13		systxsync10	systx_data8		systx_data5	systxsync4	systx_clk2		systx_cclk	sysrx_data1			sysrx_data25	sysrxsync25	sysrx_clk24	**AC**
AB		systx_data27	systx_clk26	systxsync24		systx_data21	systxsync20	systx_clk19	systx_data17	systx_clk15	systx_data13	systxsync12		systx_data7	systxsync6		systx_clk4	systx_clk0		sysrx_data2			sysrx_clk26	sysrxsync24	sysrx_data22		**AB**
AA	systx_data2		systxsync27	systx_data25	systx_clk24																	sysrxsync26	sysrx_data24	sysrx_clk23		sysrxsync22	**AA**
Y		systx_data1		systx_clk27	systxsync25																	sysrx_clk25	sysrxsync23		sysrx_clk22	sysrx_clk21	**Y**
W			systx_data0		systx_data26																	sysrx_data23		sysrx_data21	sysrx_data20	sysrxsync19	**W**
V					systx_data3																		sysrxsync21	sysrxsync20	sysrx_clk19		**V**
U																						sysrx_clk20	sysrx_data19	sysrxsync18		sysrx_data17	**U**
T																						sysrx_data18	sysrx_clk18		sysrx_clk17	sysrx_data16	**T**
R																						sysrxsync17		sysrxsync16	sysrx_data15	sysrxsync15	**R**
P																						sysrx_clk16	sysrx_clk15	sysrx_data14	sysrxsync14		**P**
N																						sysrxsync12	sysrxsync13	sysrx_data13	sysrx_clk14		**N**
M																						sysrx_clk11		sysrx_clk12	sysrx_data12	sysrx_clk13	**M**
L																						sysrx_data9	sysrxsync10		sysrxsync11	sysrx_data11	**L**
K																						sysrxsync8	sysrx_data8	sysrx_clk10		sysrx_data10	**K**
J																							sysrx_clk7	sysrx_clk8	sysrxsync9		**J**
H																						sysrxsync4		sysrx_data6	sysrx_data7	sysrx_clk9	**H**
G																						sysrxsync2	sysrx_clk5		sysrxsync6	sysrxsync7	**G**
F																						sysrxsync0	sysrx_clk3	sysrxsync5		sysrx_clk6	**F**
E																							sysrx_clk1	sysrxsync3	sysrx_data5		**E**
D																								sysrxsync1	sysrx_clk2	sysrx_clk4	**D**
C																									sysrx_csync	sysrx_clk0	**C**
B																										sysrx_cclk	**B**
A																											**A**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	

VSC9670
456-PIN EPBGA
(bottom view)

86 System Side Transmit Interface Pins
87 System Side Receive Interface Pins

Figure 1.8 Pin Diagram - System Side Interface

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
AF	Vdd_IO				Vss				Vss				Vss	Vss				Vss				Vss				Vdd_Core	AF
AE		Vss				Vdd_IO				Vdd_IO							Vdd_Core				Vdd_Core				Vss		AE
AD			Vdd_IO				Vss				Vss					Vss				Vss				Vdd_IO			AD
AC				Vss				Vdd_IO				Vdd_IO			Vdd_Core				Vdd_Core				Vss				AC
AB	Vss				Vdd_IO								Vcc_Bias			Vdd_IO						Vdd_Core				Vss	AB
AA		Vdd_IO																							Vdd_IO		AA
Y			Vss																					Vss			Y
W				Vdd_IO																			Vdd_IO				W
V	Vss																					Vcc_Bias				Vss	V
U		Vdd_IO																							Vdd_IO		U
T			Vss								Vss	Vss	Vss	Vss	Vss	Vss								Vss			T
R				Vdd_IO							Vss	Vss	Vss	Vss	Vss	Vss							Vdd_IO				R
P	Vss										Vss	Vss	Vss	Vss	Vss	Vss										Vss	P
N	Vss										Vss	Vss	Vss	Vss	Vss	Vss										Vss	N
M				Vdd_IO							Vss	Vss	Vss	Vss	Vss	Vss							Vdd_IO				M
L			Vss		Vcc_Bias						Vss	Vss	Vss	Vss	Vss	Vss								Vss			L
K		Vdd_IO																							Vdd_IO		K
J	Vss																					Vcc_Bias				Vss	J
H				Vdd_IO																			Vdd_IO				H
G			Vss																					Vss			G
F		Vdd_IO																							Vdd_IO		F
E	Vss				Vdd_Core						Vss		Vcc_Bias									Vdd_IO				Vss	E
D				Vss				Vdd_Core				Vdd_Core			Vdd_IO				Vdd_IO				Vss				D
C			Vdd_IO				Vss				Vss					Vss				Vss				Vdd_IO			C
B		Vss				Vdd_Core				Vdd_Core							Vdd_IO				Vdd_IO				Vss		B
A	Vdd_Core				Vss				Vss				Vss	Vss				Vss				Vss				Vdd_IO	A
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	

VSC9670
456-PIN EPBGA
(bottom view)

5 +5 V Bias Power Pins (Vcc_Bias)
12 Core Power Pins (Vdd_Core)
33 I/O Power Pins (Vdd_IO)
85 I/O & CORE Ground Pins (Vss)

Figure 1.9 Pin Diagram - Power & Ground

1.6 Pin Descriptions

Table 1.1 Pin List - General

Pin Name	Pin Count	Pin No.	I/O Type	Description
CLK45	1	D13	I	Crystal Clock Input. This input clock provides timing for VSC9670 internal CLK45. It is nominally a 44.736 MHz +/- 32 ppm.
REF1544	1	B19	I/O	TX Reference Input Clock. This input clock is used to generate the TXCLK for the Line Interface. It is nominally a 1.544 MHz +/- 25 ppm clock. Line Receive Output Clock. This is generated from one of the RXCLK[27:0] signals, based on bits set in the RX2REF_SEL register.
RESETN	1	C18	I	Chip Reset. This active low input provides an asynchronous reset. When asserted low, this signal forces VSC9670 to initialize all internal registers and counters to their reset values. RESETN must be asserted low for at least 100 ns.
SYSCLKMODE[1] SYSCLKMODE[0]	2	AB19 AF23	I	System Clock Mode Selection. Two-bit encoded operation mode: SYSCLKMODE[1:0] Operation Mode 00 Data Termination 01 Low Latency 10 Pin Efficient 11 Slip Buffer (see Section 1.7 Pin Mode Selection for more detail)

Table 1.2 Pin List - Line Side Interface

Pin Name	Pin Count	Pin No.	I/O Type	Description
RXCLK[27]	28	A17	I	Line Receive Clock (1.544 MHz). RXCLK is the clock signal recovered from the signal received at Line Interface Unit.
RXCLK[26]		B16		
RXCLK[25]		C15		
RXCLK[24]		B15		
RXCLK[23]		D14		
RXCLK[22]		B14		
RXCLK[21]		C13		
RXCLK[20]		B12		
RXCLK[19]		A11		
RXCLK[18]		E12		
RXCLK[17]		D11		
RXCLK[16]		B9		
RXCLK[15]		D10		
RXCLK[14]		C9		
RXCLK[13]		A7		
RXCLK[12]		C8		
RXCLK[11]		A6		
RXCLK[10]		B5		
RXCLK[9]		D7		
RXCLK[8]		A4		
RXCLK[7]		D6		
RXCLK[6]		B4		
RXCLK[5]		D5		
RXCLK[4]		A3		
RXCLK[3]		A2		
RXCLK[2]		C2		
RXCLK[1]		F5		
RXCLK[0]		D3		

Table 1.3 Pin List - Line Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
RXDATA[27]/RXPRAIL[27]	28	D16	I	Line Receive Data. Receive Digital DS-1 Data (RXDATA[27:0]), when the VSC9670 is configured to receive single-rail data by setting the use_pn_rail bit 1 to zero in the CONTROL_FE (0x40 - 0x5B) registers. These inputs are sampled by RXCLK[27:0]. Receive Positive Line Pulse (RXPRAIL[27:0]). These inputs are available when the VSC9670 is configured to receive dual-rail formatted data by setting the use_pn_rail bit 1 to one in the CONTROL_FE (0x40 - 0x5B) registers. The RXPRAIL interface can only be used in Slip Buffer mode or Pin-Efficient Muxed Data mode.
RXDATA[26]/RXPRAIL[26]		E15		
RXDATA[25]/RXPRAIL[25]		A16		
RXDATA[24]/RXPRAIL[24]		E14		
RXDATA[23]/RXPRAIL[23]		A15		
RXDATA[22]/RXPRAIL[22]		C14		
RXDATA[21]/RXPRAIL[21]		B13		
RXDATA[20]/RXPRAIL[20]		A12		
RXDATA[19]/RXPRAIL[19]		C12		
RXDATA[18]/RXPRAIL[18]		B11		
RXDATA[17]/RXPRAIL[17]		A10		
RXDATA[16]/RXPRAIL[16]		C10		
RXDATA[15]/RXPRAIL[15]		A8		
RXDATA[14]/RXPRAIL[14]		E10		
RXDATA[13]/RXPRAIL[13]		B8		
RXDATA[12]/RXPRAIL[12]		D9		
RXDATA[11]/RXPRAIL[11]		B7		
RXDATA[10]/RXPRAIL[10]		E9		
RXDATA[9]/RXPRAIL[9]		C6		
RXDATA[8]/RXPRAIL[8]		E8		
RXDATA[7]/RXPRAIL[7]		C5		
RXDATA[6]/RXPRAIL[6]		E7		
RXDATA[5]/RXPRAIL[5]		C4		
RXDATA[4]/RXPRAIL[4]		E6		
RXDATA[3]/RXPRAIL[3]		B3		
RXDATA[2]/RXPRAIL[2]		B1		
RXDATA[1]/RXPRAIL[1]		C1		
RXDATA[0]/RXPRAIL[0]		E4		

Table 1.2 Pin List - Line Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
TXCLK[27]	28	G5	O	Line Transmit Clock (1.544 MHz). TXCLK is an output clock signal to Line Interface Unit.
TXCLK[26]		E3		
TXCLK[25]		H5		
TXCLK[24]		F3		
TXCLK[23]		J5		
TXCLK[22]		G2		
TXCLK[21]		J4		
TXCLK[20]		H2		
TXCLK[19]		K5		
TXCLK[18]		H1		
TXCLK[17]		K3		
TXCLK[16]		K1		
TXCLK[15]		L2		
TXCLK[14]		M3		
TXCLK[13]		M2		
TXCLK[12]		N4		
TXCLK[11]		N2		
TXCLK[10]		P3		
TXCLK[9]		R1		
TXCLK[8]		P5		
TXCLK[7]		T1		
TXCLK[6]		R5		
TXCLK[5]		T4		
TXCLK[4]		T5		
TXCLK[3]		W1		
TXCLK[2]		U5		
TXCLK[1]		W2		
TXCLK[0]		V4		

Table 1.2 Pin List - Line Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
TXDATA[27]/TXPRAIL[27]	28	D2	O	Line Transmit Data. Transmit DS-1 Data (TXDATA[27:0]). These output signals are available when the VSC9670 is configured to transmit single-rail data by setting the use_pn_rail bit 1 to zero in the CONTROL_FE (0x40 - 0x5B) registers. These outputs are updated by TXCLK[27:0]. Transmit Positive Line Pulse (TXPRAIL[27:0]). These signals are available on the output when the VSC9670 is configured to transmit dual-rail formatted data by setting the use_pn_rail bit 1 to one in the CONTROL_FE (0x40 - 0x5B) registers. The TXPRAIL interface can only be used in Slip Buffer mode or Pin-Efficient Muxed Data mode.
TXDATA[26]/TXPRAIL[26]		F4		
TXDATA[25]/TXPRAIL[25]		D1		
TXDATA[24]/TXPRAIL[24]		G4		
TXDATA[23]/TXPRAIL[23]		E2		
TXDATA[22]/TXPRAIL[22]		F1		
TXDATA[21]/TXPRAIL[21]		H3		
TXDATA[20]/TXPRAIL[20]		G1		
TXDATA[19]/TXPRAIL[19]		J3		
TXDATA[18]/TXPRAIL[18]		K4		
TXDATA[17]/TXPRAIL[17]		J2		
TXDATA[16]/TXPRAIL[16]		L4		
TXDATA[15]/TXPRAIL[15]		M5		
TXDATA[14]/TXPRAIL[14]		L1		
TXDATA[13]/TXPRAIL[13]		N5		
TXDATA[12]/TXPRAIL[12]		M1		
TXDATA[11]/TXPRAIL[11]		N3		
TXDATA[10]/TXPRAIL[10]		P2		
TXDATA[9]/TXPRAIL[9]		P4		
TXDATA[8]/TXPRAIL[8]		R2		
TXDATA[7]/TXPRAIL[7]		R3		
TXDATA[6]/TXPRAIL[6]		T2		
TXDATA[5]/TXPRAIL[5]		U1		
TXDATA[4]/TXPRAIL[4]		U3		
TXDATA[3]/TXPRAIL[3]		V2		
TXDATA[2]/TXPRAIL[2]		U4		
TXDATA[1]/TXPRAIL[1]		V3		
TXDATA[0]/TXPRAIL[0]		Y1		

Table 1.3 Pin List - System Side Interface

Pin Name	Pin Count	Pin No.	I/O Type	Description
SYSRXCLK[27]/SYSRXSIG[27]	28	AD25	O, T	System Receive Clock. In Data Termination mode or Low Latency mode, these pins SYSRXCLK[27:0] are used for System Receive Clock Outputs. System Receive Signaling. In Slip Buffer mode, these pins are used for System Receive Signaling Output Bus, when the rx_sigbus_ena bit 4 set to one in the RXMISC (0x10) register. Otherwise, these pins are tri-stated. In Pin-Efficient Muxed Data mode, these pins are tri-stated.
SYSRXCLK[26]/SYSRXSIG[26]		AB23		
SYSRXCLK[25]/SYSRXSIG[25]		Y22		
SYSRXCLK[24]/SYSRXSIG[24]		AC26		
SYSRXCLK[23]/SYSRXSIG[23]		AA24		
SYSRXCLK[22]/SYSRXSIG[22]		Y25		
SYSRXCLK[21]/SYSRXSIG[21]		Y26		
SYSRXCLK[20]/SYSRXSIG[20]		U22		
SYSRXCLK[19]/SYSRXSIG[19]		V25		
SYSRXCLK[18]/SYSRXSIG[18]		T23		
SYSRXCLK[17]/SYSRXSIG[17]		T25		
SYSRXCLK[16]/SYSRXSIG[16]		P22		
SYSRXCLK[15]/SYSRXSIG[15]		P23		
SYSRXCLK[14]/SYSRXSIG[14]		N25		
SYSRXCLK[13]/SYSRXSIG[13]		M26		
SYSRXCLK[12]/SYSRXSIG[12]		M24		
SYSRXCLK[11]/SYSRXSIG[11]		M22		
SYSRXCLK[10]/SYSRXSIG[10]		K24		
SYSRXCLK[9]/SYSRXSIG[9]		H26		
SYSRXCLK[8]/SYSRXSIG[8]		J24		
SYSRXCLK[7]/SYSRXSIG[7]		J23		
SYSRXCLK[6]/SYSRXSIG[6]		F26		
SYSRXCLK[5]/SYSRXSIG[5]		G23		
SYSRXCLK[4]/SYSRXSIG[4]		D26		
SYSRXCLK[3]/SYSRXSIG[3]		F23		
SYSRXCLK[2]/SYSRXSIG[2]		D25		
SYSRXCLK[1]/SYSRXSIG[1]		E23		
SYSRXCLK[0]/SYSRXSIG[0]		C26		
SYSRXDATA_ENBN	1	AE24	I	System Receive Data Enable. This input is an active low signal to enable System Receive Data Output. When de-asserted, this pin enables the tri-state outputs of all SYSRXDATA[27:0] or MUXCLK45/MUXRX [DATA, SYNC, SYNC45, ENA].

Table 1.3 Pin List - System Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
SYSRXDATA[27]	28	AF25	O, T	System Receive Data. In Data Termination mode, Low Latency mode or Slip Buffer mode, these pins SYSRXDATA[27:0] are used for System Receive Data Outputs. In Pin-Efficient Muxed Data mode, there are only five fast muxed pins (MUXRXDATA, MUXRXSYNC, MUXRXSYNC45, MUXCLK45 and MUXRXENA) to use for System Receive Data Interface Outputs (for direct connect to delineator or Vitesse Semiconductor VSC9680 device). These outputs are tri-stated when the SYSRXDATA_ENBN input is high, otherwise they are continuously actively driven.
SYSRXDATA[26]		AD26		
SYSRXDATA[25]		AC24		
SYSRXDATA[24]		AA23		
SYSRXDATA[23]		W22		
SYSRXDATA[22]		AB25		
SYSRXDATA[21]		W24		
SYSRXDATA[20]		W25		
SYSRXDATA[19]		U23		
SYSRXDATA[18]		T22		
SYSRXDATA[17]		U26		
SYSRXDATA[16]		T26		
SYSRXDATA[15]		R25		
SYSRXDATA[14]		P24		
SYSRXDATA[13]		N24		
SYSRXDATA[12]		M25		
SYSRXDATA[11]		L26		
SYSRXDATA[10]		K26		
SYSRXDATA[9]		L22		
SYSRXDATA[8]		K23		
SYSRXDATA[7]		H25		
SYSRXDATA[6]		H24		
SYSRXDATA[5]		E25		
SYSRXDATA[4]/MUXCLK45		AE22		
SYSRXDATA[3]/ MUXRXDATA		AE23		
SYSRXDATA[2]/ MUXRXSYNC		AB20		
SYSRXDATA[1]/MUXRXSYNC45		AC21		
SYSRXDATA[0]/ MUXRXENA		AD22		

Table 1.3 Pin List - System Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
SYSRXSYNC[27]/RXNRAIL[27]/ RXBPV[27]	28	AE26	I/O	System Receive Synchronization. In Data Termination mode or Low Latency mode, these pins SYSRXSYNC[27:0] are used for System Receive Frame Synchronization Outputs. In these two modes the use_pn_rail bit 1 has to set zero in the CONTROL_FE (0x40 - 0x5B) registers. Receive Negative Line Pulse. These input signals RXNRAIL[27:0] are available when the VSC9670 is configured to receive dual-rail formatted data by setting the use_pn_rail bit 1 to one in the CONTROL_FE (0x40 - 0x5B) registers. The RXNRAIL interface can only be used in Slip Buffer mode or Pin-Efficient Muxed Data mode. Receive Bipolar Violation. These input signals RXBPV[27:0] are available to receive line bipolar violation indication when the VSC9670 is configured to receive single-rail data by setting the use_pn_rail bit 1 to zero in the CONTROL_FE (0x40 - 0x5B) registers. The RXBPV interface can only be used in Slip Buffer mode or Pin-Efficient Muxed Data mode.
SYSRXSYNC[26]/RXNRAIL[26]/ RXBPV[26]		AA22		
SYSRXSYNC[25]/RXNRAIL[25]/ RXBPV[25]		AC25		
SYSRXSYNC[24]/RXNRAIL[24]/ RXBPV[24]		AB24		
SYSRXSYNC[23]/RXNRAIL[23]/ RXBPV[23]		Y23		
SYSRXSYNC[22]/RXNRAIL[22]/ RXBPV[22]		AA26		
SYSRXSYNC[21]/RXNRAIL[21]/ RXBPV[21]		V23		
SYSRXSYNC[20]/RXNRAIL[20]/ RXBPV[20]		V24		
SYSRXSYNC[19]/RXNRAIL[19]/ RXBPV[19]		W26		
SYSRXSYNC[18]/RXNRAIL[18]/ RXBPV[18]		U24		
SYSRXSYNC[17]/RXNRAIL[17]/ RXBPV[17]		R22		
SYSRXSYNC[16]/RXNRAIL[16]/ RXBPV[16]		R24		
SYSRXSYNC[15]/RXNRAIL[15]/ RXBPV[15]		R26		
SYSRXSYNC[14]/RXNRAIL[14]/ RXBPV[14]		P25		
SYSRXSYNC[13]/RXNRAIL[13]/ RXBPV[13]		N23		
SYSRXSYNC[12]/RXNRAIL[12]/ RXBPV[12]		N22		
SYSRXSYNC[11]/RXNRAIL[11]/ RXBPV[11]		L25		
SYSRXSYNC[10]/RXNRAIL[10]/ RXBPV[10]		L23		

Table 1.3 Pin List - System Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
SYSRXSYNC[9]/RXNRAIL[9]/ RXBPV[9]		J25		
SYSRXSYNC[8]/RXNRAIL[8]/ RXBPV[8]		K22		
SYSRXSYNC[7]/RXNRAIL[7]/ RXBPV[7]		G26		
SYSRXSYNC[6]/RXNRAIL[6]/ RXBPV[6]		G25		
SYSRXSYNC[5]/RXNRAIL[5]/ RXBPV[5]		F24		
SYSRXSYNC[4]/RXNRAIL[4]/ RXBPV[4]		H22		
SYSRXSYNC[3]/RXNRAIL[3]/ RXBPV[3]		E24		
SYSRXSYNC[2]/RXNRAIL[2]/ RXBPV[2]		G22		
SYSRXSYNC[1]/RXNRAIL[1]/ RXBPV[1]		D24		
SYSRXSYNC[0]/RXNRAIL[0]/ RXBPV[0]		F22		
SYSRXCCLK	1	B26	I	System Receive Common Clock. This is a shared common input clock for all 28 DS-1 receive channels for Slip Buffer mode only. This input should be tied to ground in all other modes.
SYSRXCSYNC	1	C25	I	System Receive Common Synchronization. This is a shared common frame Synchronization input for all 28 DS-1 receive channels for Slip Buffer mode only. This input should be tied to ground in all other modes.

Table 1.3 Pin List - System Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
SYSTXCLK[27]/SYSTXSIG[27]	28	Y4	I/O, T	System Transmit Clock. In Data Termination mode, these pins SYSTXCLK[27:0] are used for System Transmit Clock Outputs. In Low Latency mode, these pins SYSTXCLK[27:0] are used for System Transmit Clock Inputs. System Transmit Signaling. In Slip Buffer mode, these pins SYSTXSIG[27:0] are used for System Transmit Signaling input Bus when the tx_sigbus_ena bit 4 is set to one in the TXMISC (0x18) register. Otherwise, these pins are tri-stated. In Pin-Efficient Muxed Data mode, these pins are tri-stated.
SYSTXCLK[26]/SYSTXSIG[26]		AB3		
SYSTXCLK[25]/SYSTXSIG[25]		AC2		
SYSTXCLK[24]/SYSTXSIG[24]		AA5		
SYSTXCLK[23]/SYSTXSIG[23]		AE1		
SYSTXCLK[22]/SYSTXSIG[22]		AF3		
SYSTXCLK[21]/SYSTXSIG[21]		AD4		
SYSTXCLK[20]/SYSTXSIG[20]		AC6		
SYSTXCLK[19]/SYSTXSIG[19]		AB8		
SYSTXCLK[18]/SYSTXSIG[18]		AE5		
SYSTXCLK[17]/SYSTXSIG[17]		AE7		
SYSTXCLK[16]/SYSTXSIG[16]		AF7		
SYSTXCLK[15]/SYSTXSIG[15]		AB10		
SYSTXCLK[14]/SYSTXSIG[14]		AE9		
SYSTXCLK[13]/SYSTXSIG[13]		AC11		
SYSTXCLK[12]/SYSTXSIG[12]		AE11		
SYSTXCLK[11]/SYSTXSIG[11]		AE12		
SYSTXCLK[10]/SYSTXSIG[10]		AD13		
SYSTXCLK[9]/SYSTXSIG[9]		AD14		
SYSTXCLK[8]/SYSTXSIG[8]		AE15		
SYSTXCLK[7]/SYSTXSIG[7]		AF16		
SYSTXCLK[6]/SYSTXSIG[6]		AF17		
SYSTXCLK[5]/SYSTXSIG[5]		AE18		
SYSTXCLK[4]/SYSTXSIG[4]		AB17		
SYSTXCLK[3]/SYSTXSIG[3]		AE19		
SYSTXCLK[2]/SYSTXSIG[2]		AC18		
SYSTXCLK[1]/SYSTXSIG[1]		AE20		
SYSTXCLK[0]/SYSTXSIG[0]		AB18		

Table 1.3 Pin List - System Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
SYSTXDATA[27]	28	AB2	I/O	System Transmit Data. In Data Termination mode, Low Latency mode or Slip Buffer mode, these pins SYSTXDATA[27:0] are used for System Transmit Data Inputs. In Pin-Efficient Muxed Data mode, there are only four fast muxed pins (MUXTXSYNC, MUXTXSYNC45, MUXTXENA as output and MUXTXDATA as input) to use for System Transmit Data Interface (for direct connect to delineator or Vitesse Semiconductor VSC9680 device). These four signals are updated and sampled by internal CLK45 clock.
SYSTXDATA[26]		W5		
SYSTXDATA[25]		AA4		
SYSTXDATA[24]		AC3		
SYSTXDATA[23]		AD1		
SYSTXDATA[22]		AF2		
SYSTXDATA[21]		AB6		
SYSTXDATA[20]		AE4		
SYSTXDATA[19]		AD5		
SYSTXDATA[18]		AC7		
SYSTXDATA[17]		AB9		
SYSTXDATA[16]		AD8		
SYSTXDATA[15]		AE8		
SYSTXDATA[14]		AC10		
SYSTXDATA[13]		AB11		
SYSTXDATA[12]		AF10		
SYSTXDATA[11]		AF11		
SYSTXDATA[10]		AF12		
SYSTXDATA[9]		AE13		
SYSTXDATA[8]		AC14		
SYSTXDATA[7]		AB14		
SYSTXDATA[6]		AE16		
SYSTXDATA[5]		AC16		
SYSTXDATA[4]		AF19		
SYSTXDATA[3]/ MUXTXDATA		V5		
SYSTXDATA[2]/ MUXTXSYNC		AA1		
SYSTXDATA[1]/MUXTXSYNC45		Y2		
SYSTXDATA[0]/ MUXTXENA		W3		

Table 1.3 Pin List - System Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
SYSTXSYNC[27]/TXNRAIL[27]/ TXBPV[27]	28	AA3	I/O	System Transmit Synchronization. In Data Termination mode, these pins SYSTXSYNC[27:0] are used for System Transmit Frame Synchronization Outputs. In Low Latency mode, these pins SYSTXSYNC[27:0] are used for System Transmit Frame Synchronization Inputs. In these two modes the use_pn_rail bit 1 has to set zero in the CONTROL_FE (0x40 - 0x5B) registers. Transmit Negative Line Pulse. These output signals TXNRAIL[27:0] are available when the VSC9670 is configured to transmit dual-rail formatted data by setting the use_pn_rail bit 1 to one in the CONTROL_FE (0x40 - 0x5B) registers. The TXNRAIL interface can only be used in Slip Buffer mode or Pin-Efficient Muxed Data mode. Transmit Bipolar Violation. These output signals TXBPV[27:0] are available to transmit line bipolar violation indication when the VSC9670 is configured to transmit single-rail data by setting the use_pn_rail bit 1 to zero in the CONTROL_FE (0x40 - 0x5B) registers. The TXBPV interface can only be used in Slip Buffer mode or Pin-Efficient Muxed Data mode.
SYSTXSYNC[26]/TXNRAIL[26]/ TXBPV[26]		AC1		
SYSTXSYNC[25]/TXNRAIL[25]/ TXBPV[25]		Y5		
SYSTXSYNC[24]/TXNRAIL[24]/ TXBPV[24]		AB4		
SYSTXSYNC[23]/TXNRAIL[23]/ TXBPV[23]		AD2		
SYSTXSYNC[22]/TXNRAIL[22]/ TXBPV[22]		AE3		
SYSTXSYNC[21]/TXNRAIL[21]/ TXBPV[21]		AC5		
SYSTXSYNC[20]/TXNRAIL[20]/ TXBPV[20]		AB7		
SYSTXSYNC[19]/TXNRAIL[19]/ TXBPV[19]		AF4		
SYSTXSYNC[18]/TXNRAIL[18]/ TXBPV[18]		AD6		
SYSTXSYNC[17]/TXNRAIL[17]/ TXBPV[17]		AF6		
SYSTXSYNC[16]/TXNRAIL[16]/ TXBPV[16]		AC9		
SYSTXSYNC[15]/TXNRAIL[15]/ TXBPV[15]		AD9		
SYSTXSYNC[14]/TXNRAIL[14]/ TXBPV[14]		AF8		
SYSTXSYNC[13]/TXNRAIL[13]/ TXBPV[13]		AD10		
SYSTXSYNC[12]/TXNRAIL[12]/ TXBPV[12]		AB12		
SYSTXSYNC[11]/TXNRAIL[11]/ TXBPV[11]		AD12		
SYSTXSYNC[10]/TXNRAIL[10]/ TXBPV[10]		AC13		

Table 1.3 Pin List - System Side Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
SYSTXSYNC[9]/TXNRAIL[9]/ TXBPV[9] SYSTXSYNC[8]/TXNRAIL[8]/ TXBPV[8] SYSTXSYNC[7]/TXNRAIL[7]/ TXBPV[7] SYSTXSYNC[6]/TXNRAIL[6]/ TXBPV[6] SYSTXSYNC[5]/TXNRAIL[5]/ TXBPV[5] SYSTXSYNC[4]/TXNRAIL[4]/ TXBPV[4] SYSTXSYNC[3]/TXNRAIL[3]/ TXBPV[3] SYSTXSYNC[2]/TXNRAIL[2]/ TXBPV[2] SYSTXSYNC[1]/TXNRAIL[1]/ TXBPV[1] SYSTXSYNC[0]/TXNRAIL[0]/ TXBPV[0]		AE14 AF15 AD15 AB15 AD17 AC17 AD18 AF20 AD19 AF21		
SYSTXCCLK	1	AC20	I	System Transmit Common Clock. This is a shared common input clock for all 28 DS-1 transmit channels for Slip Buffer mode only. This input should be tied to ground in all other modes.
SYSTXCSYNC	1	AD21	I	System Transmit Common Synchronization. This is a shared common frame Synchronization input for all 28 DS-1 transmit channels for Slip Buffer mode only. This input should be tied to ground in all other modes.

Table 1.4 Pin List - Microprocessor Interface

Pin Name	Pin Count	Pin No.	I/O Type	Description
CPU_A[7]	8	C22	I	CPU Address Bus (Non-multiplexed Mode). These inputs select one of the VSC9670 internal registers during read or write access.
CPU_A[6]		A23		
CPU_A[5]		E19		
CPU_A[4]		D20		
CPU_A[3]		C21		
CPU_A[2]		B22		
CPU_A[1]		E18		
CPU_A[0]		A21		
CPU_A_D[7]	8	A25	I/O, T	CPU Address/Data Bus (Multiplexed Mode). CPU Data Bus (Non-multiplexed Mode). These bi-directional, tri-stated data buses are used to access VSC9670 internal registers.
CPU_A_D[6]		B24		
CPU_A_D[5]		A24		
CPU_A_D[4]		E21		
CPU_A_D[3]		D22		
CPU_A_D[2]		C23		
CPU_A_D[1]		B23		
CPU_A_D[0]		E20		
CPU_ALE	1	AF24	I	CPU Address Latch Enable. In Multiplexed Mode, this input is active high, and CPU address bus is latched on falling edge of this signal. In Non-multiplexed Mode, this pin shall be tied to high.
CPU_SELN	1	AB21	I	CPU Select. This active low input signal selects the VSC9670 for a CPU read or write operation.
CPU_WRN	1	AD23	I	CPU Write Enable. In Intel bus mode, this active low input operates with CPU_SELN to configure the data bus lines CPU_A_D[7:0] as input (Write cycle). In special Motorola multiplexed mode, the falling edge of this signal acts as a data strobe (Write cycle).

Table 1.4 Pin List - Microprocessor Interface (continued)

Pin Name	Pin Count	Pin No.	I/O Type	Description
CPU_RDN_RDWRN	1	AC22	I	CPU Read or Write Enable. In Intel bus mode, this active low input operates with CPU_SELN to configure the data bus lines CPU_A_D[7:0] as output (Read cycle). In Motorola bus mode, this input operates with CPU_SELN to configure the data bus lines CPU_A_D[7:0] as output (Read cycle) when CPU_RDN_RDWRN = 1, or to configure the data bus lines CPU_A_D[7:0] as input (write cycle) when CPU_RDN_RDWRN = 0. In special Motorola multiplexed mode, this signal is sampled during the falling edge of CPU_WRN. If CPU_RDN_RDWRN is low then the VSC9670 is being written to otherwise the VSC9670 is being read from.
CPU_RDY_DTACKN	1	D21	O	CPU Ready or Data Acknowledge. In Intel Mode this signal is not used. In Motorola Mode this signal indicates valid data on data bus (during CPU read) and indicates data has been accepted (during CPU write).
CPU_DBUF	1	D18	I	CPU Double Buffered error counts. Input to VSC9670 from external 1-second timer (typically). To control switchover of double buffered error counts. A single high pulse of CLK45 enables this function. When asserted this signal will cause the internal COUNT_RAM to switch over from the active copy to the shadow copy (the active bank becomes the shadow and the shadow bank becomes the active). The CPU then has 1 second to read the entries from the bank which has just become the shadow bank.
CPU_INT	1	A20	O	CPU Interrupt Request. This output is the interrupt indication signal. Depending on the programmable polarity selection, this output is driven high or low while one or more of the internal interrupt sources is active and unmasked.
CPU_MP[1] CPU_MP[0]	2	B20 C19	I	CPU Microprocessor Interface Select. CPU_MP[1:0] Operation Mode 00 Intel 01 Motorola 10 Multiplexed 11 Special Motorola Multiplexed

Table 1.5 Pin List - JTAG Interface and SCAN Interface

Pin Name	Pin Count	Pin No.	I/O Type	Description
TDI	1	E16	I	Scan Data In.
TCK	1	B18	I	Scan Clock.
TMS	1	A19	I	Scan Mode Select.
TRST_N	1	D17	I	Scan Reset (Active Low).
TEST_EN	1	C17	I	Scan Test Enable. This pin is used during SCAN testing. This signal when asserted will cause some internal muxes to select CLK45 as the register clock instead of the normal operating CLK1544.
TDO	1	E17	O	Scan Data Out.

Table 1.6 Pin List - Power and Ground

Pin Name	Pin Count	Pin No.	I/O Type	Description
VDD_CORE	12	A1, B6, B10, D8, D12, E5, AB22, AC15, AC19, AE17, AE21, AF26	Power	The CORE power supply pins VDD_CORE should be connected to a well decoupled +3.3V supply in common with VDD_IO.
VDD_IO	33	C3, F2, H4, K2, M4, R4, U2, W4, AA2, AB5, AF1, AC8, AC12, AD3, AE6, AE10, R23, U25, W23, AA25, AD24, A26, E22, F25, H23, K25, M23, B17, B21, C24, D15, D19, AB16	Power	The I/O Pads power supply pins VDD_IO should be connected to a well decoupled +3.3V supply in common with VDD_CORE.
VCC_BIAS	5	V22, J22, E13, L5, AB13	Power	+5 V Bias input power pins. These VCC_BIAS tolerant input pins should be connected to a +5V power supply if the device is interfacing to 5V powered logic. If 5V tolerant is not required, these pins should be connected to the +3.3V power supply.
VSS	85	A5, A9, A13, A14, A18, A22, B2, B25, C7, C11, C16, C20, D4, D23, E1, E26, G3, G24, J1, J26, L3, L11, L12, L13, L14, L15, L16, L24, M11, M12, M13, M14, M15, M16, N1, N11, N12, N13, N14, N15, N16, N26, P1, P11, P12, P13, P14, P15, P16, P26, R11, R12, R13, R14, R15, R16, T3, T11, T12, T13, T14, T15, T16, T24, V1, V26, Y3, Y24, AB1, AB26, AC4, AC23, AD7, AD11, AD16, AD20, AE2, AE25, AF5, AF9, AF13, AF14, AF18, AF22, E11	Power	The I/O and CORE Pads ground pins VSS should be connected to GND.

1.7 Pin Mode Selection

The external SYSCLKMODE[1:0] pins help to select various clocking flows, system side interface options, and LIU interface options.

Table 1.7 RX System Side Interface Related Pin Use

SYSCLKMODE[1:0]	Operation Mode	System Side RX Pins		
		SYSRXDATA/ MUXRX(5-pin)	SYSRXCLK/ SYSRXSIG	SYSRXSYNC/ RXNRAIL/ RXBPV
00	Data Termination	OUT (SYSRXDATA)	OUT (SYSRXCLK)	OUT (SYSRXSYNC)
01	Low Latency	OUT (SYSRXDATA)	OUT (SYSRXCLK)	OUT (SYSRXSYNC)
10	Pin Efficient Interface	OUT (MUXRX 5-pin) Interface	TRI (not used)	IN (RXNRAIL, if use_pn_rail=1) or IN (RXBPV, if use_pn_rail=0) (both for LIU)
11	Slip Buffer	OUT (SYSRXDATA)	TRI (rx_sigbus_ena=0) or OUT (SYSRXSIG for RX signaling)	IN (RXNRAIL, if use_pn_rail=1) or IN (RXBPV, if use_pn_rail=0) (both for LIU)

Table 1.8 RX Line Side Interface Related Pin Use

SYSCLKMODE[1:0]	Operation Mode	Line Side RX Pins	
		RXDATA/RXPRAIL	RXCLK
00	Data Termination	IN (RXDATA)	IN (RXCLK)
01	Low Latency	IN (RXDATA)	IN (RXCLK)
10	Pin Efficient Interface	IN (RXDATA, if use_pn_rail=0) or IN (RXPRAIL, if use_pn_rail=1)	IN (RXCLK)
11	Slip Buffer	IN (RXDATA, if use_pn_rail=0) or IN (RXPRAIL, if use_pn_rail=1)	IN (RXCLK)

The frc_dtm_mode bit in the TXMISC register is used to force the TX side into Data Termination Mode while the RX side remains in SlipBuffer Mode (SYSCLKMODE[1:0] = 11).

Table 1.9 TX System Side Interface Related Pin Use

frc_dtm_ mode	SYSCLK MODE[1:0]	Operation Mode	System Side TX Pins		
			SYSTXDATA/ MUXTX(4-pin)	SYSTXCLK/ SYSTXSIG	SYSTXSYNC/ TXNRAIL/TXBPV
1	xx	Data Termination	IN (SYSTXDATA)	OUT (SYSTXCLK)	OUT (SYSTXSYNC)
0	00	Data Termination	IN (SYSTXDATA)	OUT (SYSTXCLK)	OUT (SYSTXSYNC)
0	01	Low Latency	IN (SYSTXDATA)	IN (SYSTXCLK)	IN (SYSTXSYNC)
0	10	Pin Efficient Interface	IN/OUT (MUXTX 4-pin) Interface	TRI (not used)	OUT (TXNRAIL, if use_pn_rail=1) or OUT (TXBPV, if use_pn_rail=0) (both for LIU)
0	11	Slip Buffer	IN (SYSTXDATA)	TRI (tx_sigbus_ena=0) or IN (SYSTXSIG for TX signaling)	OUT (TXNRAIL, if use_pn_rail=1) or OUT (TXBPV, if use_pn_rail=0) (both for LIU)

Table 1.10 TX Line Side Interface Related Pin Use

frc_dtm_ mode	SYSCLK MODE[1:0]	Operation Mode	Line Side TX Pins	
			TXDATA/TXPRAIL	TXCLK
1	xx	Data Termination	OUT (TXDATA)	OUT (TXCLK)
0	00	Data Termination	OUT (TXDATA)	OUT (TXCLK)
0	01	Low Latency	OUT (TXDATA)	OUT (TXCLK)
0	10	Pin Efficient Interface	OUT (TXDATA, if use_pn_rail=0) or OUT (TXPRAIL, if use_pn_rail=1)	OUT (TXCLK)
0	11	Slip Buffer	OUT (TXDATA, if use_pn_rail=0) or OUT (TXPRAIL, if use_pn_rail=1)	OUT (TXCLK)

In Data Termination and Low Latency modes, each DS1 requires 3 signals on the system side (clock, data and sync), and no pins can be reused. On the LIU line side, only 2 signals are available per DS1, and hence the LIU must support both clock and data connections.

In Pin Efficient and Slip Buffer modes, some saved system side pins can be reused for the line side interface. The LIU line side can therefore be used in a clock/data/BPV (TXBPV/RXBPV) type of connection or clock/P-rail (TXPRAIL/RXPRAIL) /N-rail (TXNRAIL/RXNRAIL) type of connection.

In Slip Buffer mode, the robbed bit signaling bus SYSTXSIG and SYSRXSIG can also be active, and the SYSTXCCLK and SYSRXCCLK pins are used as common system-side clocks for all DS1s.

2.0 Functional Description

The highly integrated VSC9670 uses a time-sliced state machine to provide up to 28 channels of DS1 framing and other processing.

A front end stage is used to widen each DS1 data path from a single bit to an 8-bit path. The RX and TX state machines work on this byte data path. Other state machines perform error logging and counts, FDL processing, robbed bit signaling, etc. Internal SRAMs are used for slip buffering, if enabled. A back end stage re-converts the internal 8-bit data path to a 1-bit wide data path in order to interface to a standard TDM backplane. The back end can connect directly to a high speed TDM bus, as well as to a muxed low pin-count interface.

2.1 Front End Logic

Each front end channel in the RX direction has the following logic: a serial-to-parallel converter (shift register) that creates 8-bit wide data, a second register stage for double buffering, and request/grant logic. When a given channel’s shift register is full, the byte is transferred into the second stage register and a request flag is asserted. The RX FSM bandwidth is sized so that it can always service a request from port #j (sink one byte’s worth of data arriving on DS1 #j) in enough time to prevent a double-buffer over-run, even if all other 27 DS1 requests are pending at the same time.

Similar logic is used in the TX direction. A parallel to serial converter (shift register) is loaded from a second stage buffer register as needed. Request/grant logic allows the TX front stage to ask the TX state machine for an octet of data to fill the double buffer register stage. The TX machine processing capacity is such that it will always respond to the request (source one byte of data to DS1 port #j) in time to prevent double buffer under-run.

2.2 RX State Machine

The RX state machine is driven by requests from the TDM front end. It processes requests from the 28 incoming DS1 data streams in an arbitrary order.

When servicing a request from a particular channel, the RX state machine processes the data octet from that channel. It may establish framing if the channel is in OOF, or create updates for other state machines. In some cases, the RX state machine may overwrite all or part of the incoming byte - for example, when forcing an AIS pattern downstream.

After the octet is processed, it is handed off to the back end logic along with sync position information. The state machine information that was used for this channel is then written back to the RX state RAM, and the RX state machine is ready to service the next request from the front end block.

During OOF, the RX direction internal counters are free-running, maintaining the previous frame alignment. The framing algorithm is run if framing is currently lost, and the counters are updated after framing is recovered. Essentially the RX state machine adds sync-position information to a incoming data byte from the front end section, and passes data/sync to the back end. Errors are logged simultaneously, FDL and signaling bits are stripped off, and miscellaneous functions such as loop-up/loop-down code detection and pseudo-random pattern detection are performed.

As required by TR62411, the RX framing algorithm acquires a frame within a maximum average re-frame time of 50 ms in both ESF and SF modes. High performance is achieved even in the presence of line errors because the algorithm examines 772 bits (ESF) or 193 bits (SF) at a time. The shared state machine implementation does not affect frame acquisition performance. This means that all 28 channels can simultaneously acquire frames within the same or less time as a single channel acquisition.

The algorithm also has strong protections against mimic patterns: in SF mode, declaration of frame sync acquisition is inhibited if mimics are found, and in ESF mode, CRC6 can be used to filter out mimics.

Pattern detection in the RX state machine comes in two flavors. There is a simple state machine that is always active and which detects the simpler patterns, such as AIS (all 1's, the in-band loopback code) or the SF yellow alarm code (bit2 always '0'). There is also a pseudo-random number (PN) pattern detector, which can be used together with the PN pattern generator in the TX state machine to perform in-service or out-of-service line quality tests. Note that the two types of pattern detectors are completely independent and simultaneously active.

2.2.1 RX Repetitive Pattern Detection

In the RX direction, the following simple repetitive patterns will be detected:

If LOS, detect

1's density	Detect when adequate 1's density returns after LOS in bipolar rail mode (12.5% pulse density for 175 ± 75 clocks)

Else detect

Unframed all 1s	AIS (must be in OOF to flag AIS detection)
00001…	Loop-up: inband loopback activate (CSU)
00011…	Loop-up: inband loopback activate (NI)
001 …	Loop-down: inband loopback deactivate (CSU)
00111 …	Loop-down: inband loopback deactivate (NI)
Bit2 = 0	SF Yellow alarm
00010111	Idle code (octet aligned in each DS0)
10100	ANSI T1.403

The above patterns are detected over 16 intervals of 3 ms each (except for LOS). The pattern is said to be detected if a threshold number of these 16 intervals shows the desired pattern. The pattern detector behavior is widely configurable using per-chip registers MIS_3MS_0, MIS_3MS_1, MIS_WIN16_0, MIS_WIN16_1, but most users should follow the programming suggested in the register map. After pattern detection by the RX state machine, alarm integration and reporting is provided in the performance monitoring block.

2.2.2 RX Pseudo-Random Number (BERT) Pattern Detector

The pseudo-random number (PN) detector consists of a 32-bit shift register with an XOR feedback tap. The free running XOR feedback register can be used as a generator or as a detector of bit patterns corresponding to polynomials of the form 2^n -1.

For use as a detector, “size” number of incoming bits are loaded into the shift register as a seed value, and the free-running register output is compared to the subsequent input bit patterns.

If a certain number of matches are found, the pattern is said to be in sync, and any further single bit mismatches are logged into an error count register. If a certain threshold rate of mismatches is exceeded, synchronization is lost. The BERT unit then attempts to re-sync using the pattern detection algorithm.

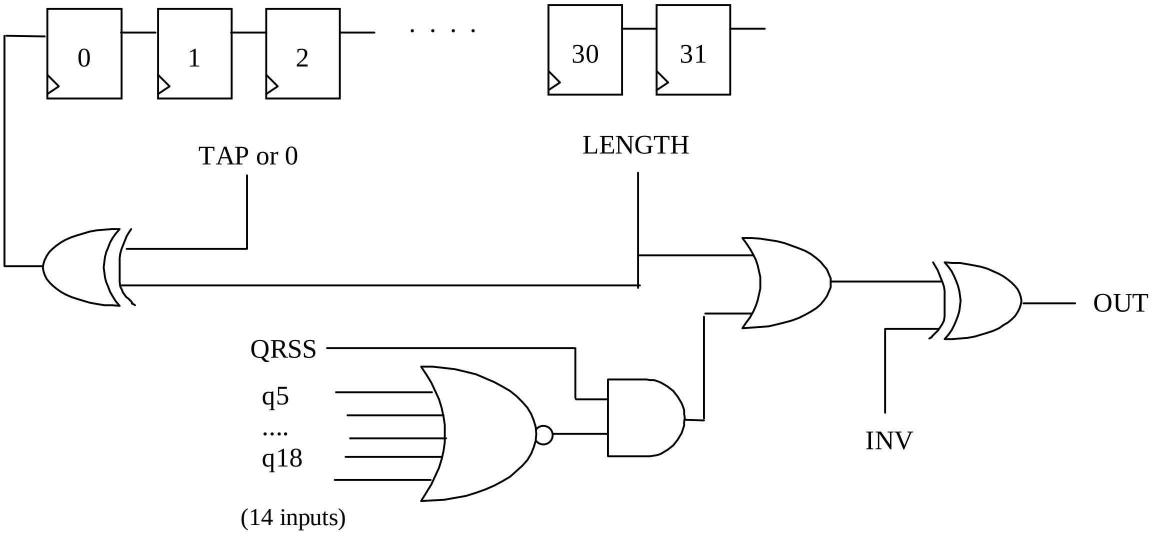


Figure 2.1 RX BERT Pattern Detector

As can be seen from the register map in Section 3.3 TX State Machine, Registers and RAM, one longword location in the RX state RAM is used for storing the PN pattern. The PN detector can be configured in terms of tap position and size, in order to detect commonly used BERT patterns such as 2^{11} -1, 2^{15} -1, 2^{23} -1, etc.

The INV setting inverts the output of the shift register. QRSS pattern detection is also provided as shown (QRSS is a 2^{20} polynomial with protection against more than 14 consecutive zeros).

The PN detector can also be configured with DS0 resolution to determine which DS0s or combination of DS0s to check for PN pattern matching. This provides the flexibility to run the PN pattern detection on some unused DS0s in a fractional T1 application for continuous in-service line performance monitoring.

The PN detector can also be configured for DS0-resolution detection of repetitive patterns of arbitrary length. In this case, the “TAP” input to the XOR gate is made ‘0’.

The PN or repetitive pattern detection algorithm tolerates incoming bit errors and can lock on a pattern with an error rate worse than BER = 1 x 10^{-3}. Once the detector is synchronized, a count of bit errors is maintained with a “main” and a “shadow” copy. An internal 1-second event, a CPU write, or an external pulse can be used to switch between the count copies. (See bits 1/0 in the ERRMISC register in Section 3.4.1 Per Chip Registers)

2.3 TX State Machine

The TX state machine is driven by requests from the TDM front end. It processes requests in an arbitrary order.

While processing the front end request for an outgoing data byte for a particular channel, the TX state machine supplies the byte, and as a side effect may create requests or updates for the errors, FDL, or signaling state machines. In some cases, the TX state machine may overwrite all or part of the incoming byte (such as when forcing a loop-back code upstream).

In order to supply the byte requested by the front end, a new byte is requested from the back end logic. Sync position information is also provided to the back end. The state information that was used for this channel is then written back to the TX state RAM, and the TX state machine is then ready to service the next request from the front end block.

Essentially, the TX state machine adds sync position information to a data byte being requested by the front end section and passes sync to the back end. Errors may be optionally inserted simultaneously, FDL and signaling bits are added, and miscellaneous functions such as loop-up/loop-down code generation and pseudo-random pattern generation are performed.

In addition to repetitive and BERT patterns, the TX machine can insert per-DS1 forcing patterns (such as upstream AIS) and per-DS0 patterns (such as IDLE, DMW, etc.). The Register Map section provides more details.

2.3.1 TX Pattern Insertion (Repetitive and Pseudo-Random BERT)

The pseudo-random number (PN) generator is similar to the PN detector in the RX machine. A 32-bit shift register with a feedback tap is available and can be configured in terms of size and tap position. Commonly used BERT patterns such as 2^{11} - 1, 2^{15} - 1, 2^{23} - 1, etc. can be generated, as well as QRSS pattern generation.

The free running output of the PN generator can be selected for insertion with per-DS0 resolution on one or more DS0s. This provides the flexibility to run the PN pattern generation on some unused DS0s in a fractional T1 application, for continuous in-service line performance monitoring.

The PN generator can also be configured for per-DS0 resolution generation of repetitive patterns up to 32 bits in length. In this case, the “TAP” input to the XOR gate is made ‘0’.

In all these cases, the CPU can insert a single error in the outgoing pattern.

2.4 Slip Buffer and Back End Logic

Separate slip buffers are provided in the RX and TX directions. Two frames of data are buffered in each direction for each one of the 28 DS1 channels. The slip buffers are optional and can be omitted from the data flow when a common backplane clock interface is not required.

The back end logic is the converse of the front end logic: in the RX direction, it performs the parallel-to-serial data conversion, and in the TX direction, it performs the serial-to-parallel data conversion.

If the slip buffer mode is enabled, the RX slip buffer is inserted between the RX state machine and the RX back end. The write pointer into the RX slip buffer is controlled by the incoming DS1 timing and the read pointer is controlled by the backplane timing. If the slip buffer read and write pointers cross, a frame-aligned slip is performed and an RX frame slip indicator is provided to the CPU.

Similarly if a slip buffer mode is enabled, the TX slip buffer goes between the TX state machine and the TX back end. The read and write pointers are driven by the TX FSM timing and backplane timing respectively. Frame aligned slip is automatically performed if a timing mismatch causes a pointer crossing, and the CPU is notified.

The slip buffers provide a convenient location for implementation of per-DS0 data forcing. In a DS1’s RX slip buffer, the CPU can write a byte value into the appropriate DS0 location in both frames and set an enable bit. The slip buffer write process will avoid modifying that byte location, and the CPU-written value will be sent out constantly to the system side on the appropriate DS0 timeslot.

Similarly in the TX slip buffer, per-DS0 forcing can be used to overwrite system side data in specific DS0 locations before sending the data out to the line side. The byte that is used to replace the DS0 data must be written by the CPU into the appropriate DS0 location in both frame positions of the slip buffer.

2.5 Clocking Schemes

The flexible, scalable framer VSC9670 can support various clocking schemes. A single DS1 slice is illustrated in Figure 2.2, Figure 2.3, and Figure 2.4. It is important to understand that in the core of the VSC9670 architecture,

- For the RX side, data and timing flow from the line side to the system side
- For the TX side, timing flows from the line side to the system side, and data flows from the system side to the line side.

Other timing philosophies can be accommodated as shown in one of the following three modes.

2.5.1 Data Termination Mode

In the data termination mode in the RX direction, data, timing and sync flow from the M13 or DS1 Line Interface Unit into the framer block and into the HDLC delineator block. The front end performs data packing, collecting 8 bits at a time and handing them off to the RX FSM. After processing, the RX FSM hands off 8 bits at a time to the back end stage which performs the parallel data byte to serial bit stream conversion.

The back end shift register absorbs the delay variations attributable to the data path width conversion and state machine arbitration. The back end shift register is clocked by a de-jittered version of rxclk if de-jittering is enabled. Normally, de-jittering will not be used in a data application.

In the TX direction, timing flows from the VSC9670 front end to the HDLC block, and data flows from the HDLC block to the M13 or DS1 LIU. A 1.544 MHz clock must be presented to the front end shift register’s txclk input, and the same clock must be fed to the M13 or LIU. This reference can be connected to rxclk for loop timing, or to a local high quality oscillator.

The front end can be thought of as performing a packing of “timing”, converting 8 TXCLK’s into a request to the TX FSM. The TX FSM in turn pulls a byte of data from the back end shift register, performs various processing, and hands off the byte to the front end. The back end performs the parallel-to-serial conversion of “timing”; it removes the jitter caused by both single-bit to 8-bit data path conversion and by state machine arbitration and processing.

In the data termination mode, SYSRXCLK and SYSTXCLK can be gapped at the frame position for a direct, glue-less interface to PMC-Sierra HDLC controllers. For both the RX and TX direction sync signals, which are VSC9670 outputs, it is possible to generate an 8 KHz sync (or multiframe sync) which is pulsed once every 1.5 ms in SF mode or once every 3 ms in ESF mode. The back-end shift register is set for a size of 32 bits, which is adequate to absorb the shared state machine jitter.

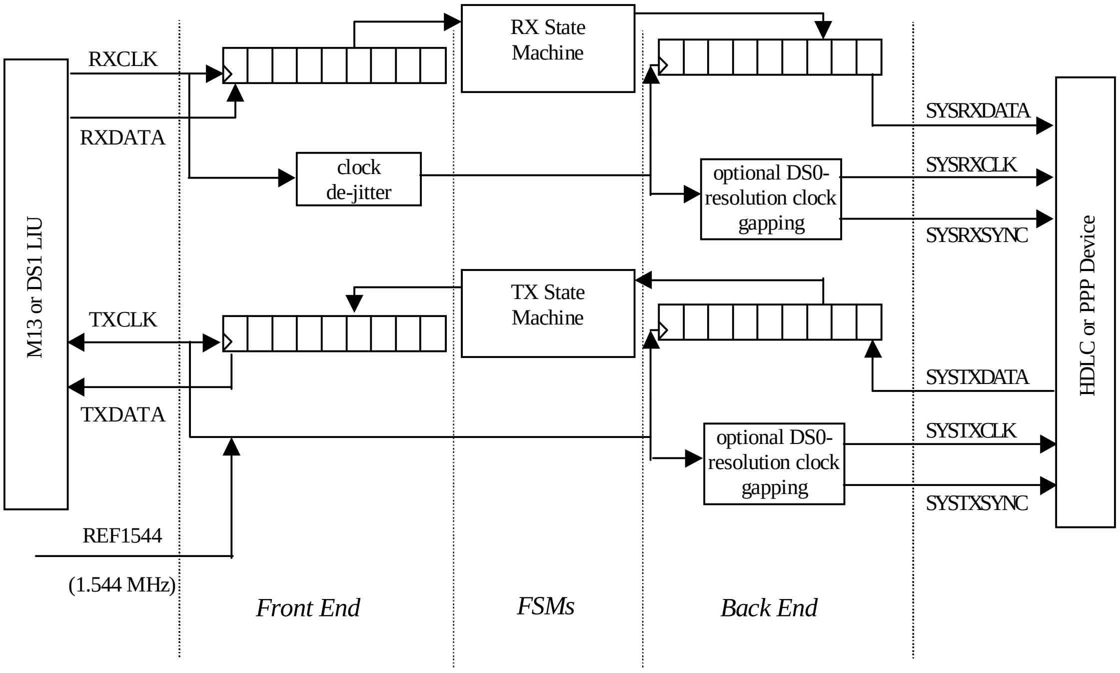


Figure 2.2 Data Termination Mode Signals Flow (per DS1)

2.5.2 TDM Backplane with Slip-Buffer Mode

In this TDM Backplane mode, the RX slip buffer is enabled and allows decoupling of the received rxclk timing from the backplane SYSRXCCLK (the system receive common clock) timing. As before, data, sync and timing flow from the front end through the RX FSM and into the RX slip buffer.

The back end shift register only exists for the purpose of converting RX slip buffer read data (8-bit wide data) into a single bit data stream. The back end shift register read pointer is driven by SYSRXCCLK, and when the shift register is less than half-full, a byte is loaded from the RX slip buffer. Hence the read pointer into the slip buffer runs at a rate directly determined by the backplane SYSRXCCLK. The slip buffer write pointer runs at a rate determined by RXCLK.

In the TX direction, the TX slip buffer is enabled, and hence backplane TX clock, data and sync are all inputs into the framer block. The VSC9670 “back end” exists only to convert signals from the bit stream oriented backplane pins to the octet oriented slip buffer.

The back end loads a TX data byte into the slip buffer each time the back end shift register is less than half-empty. Therefore, the slip buffer write pointer moves at a rate directly traceable to backplane systxcclk (the system transmit common clock). The TX slip buffer read pointer runs at the txclk rate, and hence clock decoupling occurs between txclk and systxcclk rates.

The clock TXCLK which drives the VSC9670 front end can be the same as the (smoothed) RXCLK if loop timing is being used, or it can be a local 1.544 MHz oscillator.

The back end shift register is set for a size of 32 bits, which is adequate to absorb the shared slip state machine jitter

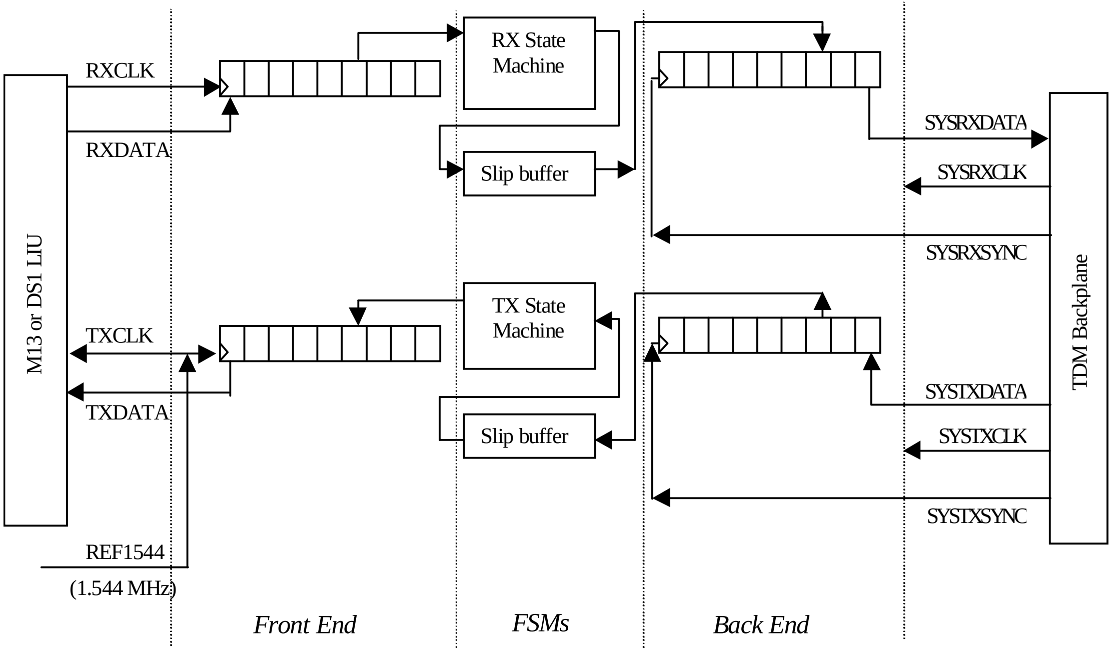


Figure 2.3 TDM Backplane with Slip-Buffer Mode Signals Flow

2.5.3 Low Latency Mode

The low latency mode is similar to the data termination mode except that systxclk is a chip input. This mode is useful if the TDM nature of the DS1 streams needs to be preserved without adding a slip buffer latency.

In the low latency TDM mode in the RX direction, the front end section receives data and timing from the line, and this same timing source is connected to the back end shift register (with optional clock de-jitter). In the TX direction, the front end section is driven by the same clock as the back end (systxcclk-27, which is an input to the VSC9670).

As before, the back end logic absorbs the delay variation attributed to data path width conversion and state machine arbitration. Similar to DTM, 8 KHz rate frame sync or SF or ESF rate sync can be generated or accepted.

The back end shift register can be set for a size of 32 or 72 bits depending upon how much incoming jitter is expected and how much remains after attenuation by the optional De-Jitter Buffer.

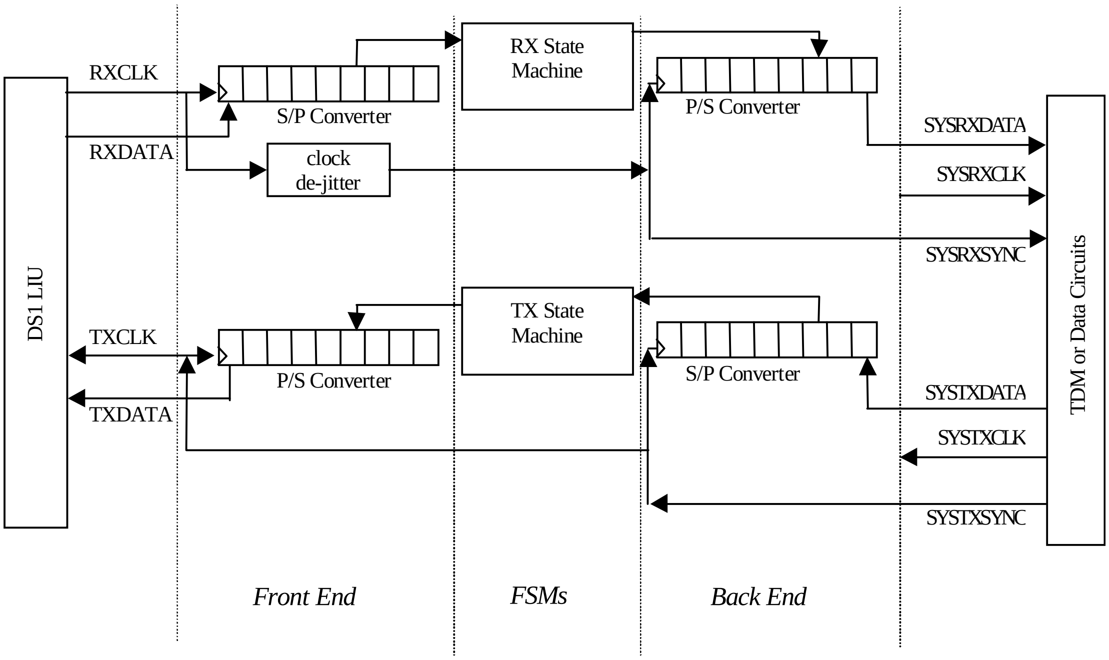


Figure 2.4 Low Latency Mode Signals Flow (per DS1)

2.6 Special System Side Interfaces

Apart from the standard clock/data/sync type of system side interface, two types of special interfaces are also provided:

- direct attachment to a high speed TDM backplane such as H.100
- for data termination applications, a high speed, pin-efficient, muxed data bus

Note that this section explains two different ways of attaching the VSC9670 to the rest of the system, but does not introduce any fundamentally new or different clocking modes. Specifically the 2.048 MHz, 4.096 MHz and 8.192 MHz high speed backplane interfaces all share the slip buffer mode of clocking flow. Similarly the muxed data bus is an example of data termination mode clocking flow because each individual DS1 preserves its distinct timing.

2.6.1 High Speed Backplane Attachment

In the backplane slip buffer mode, with both RX and TX slip buffers enabled, the VSC9670 can directly interface to a 1.544 MHz TDM backplane. The VSC9670 can also directly interface to a 2.048 MHz TDM backplane by mapping the 24 incoming DS0 timeslots to the 32 backplane DS0 timeslots by a repetitive structure of {1 unpopulated, 3 populated}.

ST28 can also directly connect to a 4.096 MHz or 8.192 MHz backplane. If connecting to a 4.096 MHz backplane, only half of the system-side RX or TX wires are active. In other words, SYSRXDATA0, SYSRXDATA2, SYSRXDATA4 … SYSRXDATA26 are active. The SYSRXDATA0 wire carries data from two DS1s, specifically DS1 #0 and DS1 #1; the SYSRXDATA2 wire carries data from DS1 #2 and DS1 #3, ... up to SYSRXDATA26, which carries data from DS1 #26 and DS1 #27.

The process is similar for TX data wires. Note that since the slip buffers are required to be used, all the individual DS1 RX timings have been reduced to one common backplane RX timing (SYSRXCSYNC, SYSRXCCLK), and similarly for the TX timings (SYSTXCSYNC, SYSTXCCLK).

Similarly, to connect the VSC9670 to a 8.192 MHz backplane, only one-quarter of the system-side RX or TX data wires are active. Again note that the slip buffers are required; hence all RX data is presented on SYSRXDATA0, SYSRXDATA4, … up to SYSRXDATA24, with a common SYSRXCCLK and SYSRXCSYNC. All TX data is similarly received on SYSTXDATA0, SYSTXDATA4 … up to SYSTXDATA24, with a common SYSTXCCLK and SYSTXCSYNC.

The arrangement of individual DS0 timeslots in these high speed backplane cases is shown in Figure 2.5 below. In the 2.048 MHz backplane, the repeating structure is 32 DS0 timeslots, in which timeslots 0 ... 23 are arranged as shown. In the 4.096 MHz backplane, SYSRXDATA0 is shown carrying DS0s belonging to two DS1s - DS1 #0 and DS1 #1. The top line numbering shows the DS1 and the bottom line numbering shows the DS0 timeslots. Similarly in the 8.192 MHz picture, four DS1s are carried on one wire. The numbering again goes from 0 ... 3 for the four DS1s, and 0 ... 23 for the DS0s in each DS1.

2.6.2 Pin-Efficient Muxed Data Bus

In data termination applications the high speed muxed data bus allows direct, very pin-efficient connectivity to a companion HDLC or ATM cell delineation chip. An important benefit of this mode is that each embedded DS1 signal retains its own timing without having to be slip-buffer adapted to a common backplane rate.

The muxed data bus uses the common 44.736 MHz chip clock and consists of four wires: data, sync, sync45 and enable. The usage is as follows: with VSC9670 clock and data flow as shown in the Figure 2.6, note that the RX direction back end produces 28 independent streams of data / sync / clock.

These 28 streams are over-sampled using the 44.736 MHz clock. The high speed sampling circuit cycles in order from DS1 #0 to DS1 #27, and samples the data and sync values for each particular DS1 circuit. If no new values are available on a particular DS1 circuit, the sampler de-asserts the enable; if not, the sampler outputs the data and sync values and asserts enable.

Similarly in the TX direction, the 28 independent TX DS1s are “over-sampled” - that is, the 44.736 MHz sampling circuitry cycles through each DS1 back end logic and, if a new bit value is needed, the enable output is asserted and data values are input for that particular DS1. If a particular DS1 does not need a new data bit value, the enable signal is de-asserted and no new value is input into the VSC9670.

The operation of the high speed muxed data bus can be most easily understood from the waveform in Figure 2.6 which shows the RX muxed data bus signals, and internal signals in another device, say an HDLC or ATM delineator which is receiving VSC9670 framed data.

Inside the delineator block there is an internal mod-28 counter, which counts from 0 ... 27 and aligns itself to MUXRXSYNC45 as shown, so that the mod-28 counter equals ‘0’ at MUXRXSYNC45 assertion.

The delineator block then has enough information to de-mux the 28 DS1 data streams which were muxed together by the VSC9670. Note that enable equal to ‘0’ means both data and sync in that clock tick are ignored. Since the muxed data bus clock (44.736 MHz) is faster than 28 * 1.544 MHz there will be some number of times in each DS1 context where enable is de-asserted. This effectively provides the timing elasticity required for the muxed data bus operation.

As shown in the waveform, the mod-28 counter is synchronized by MUXRXSYNC45 and de-muxes the 28 DS1 data streams. In the first text callout box, DS1 stream #5 has data =1 and sync bit set, which identifies the DS0 boundaries in this DS1 to the HDLC or ATM delineator chip.

In other text callout boxes examples are shown where the enable bit = 0, which means that the particular DS1 stream input data ran out of data (which happens every so often because of the over-sampling).

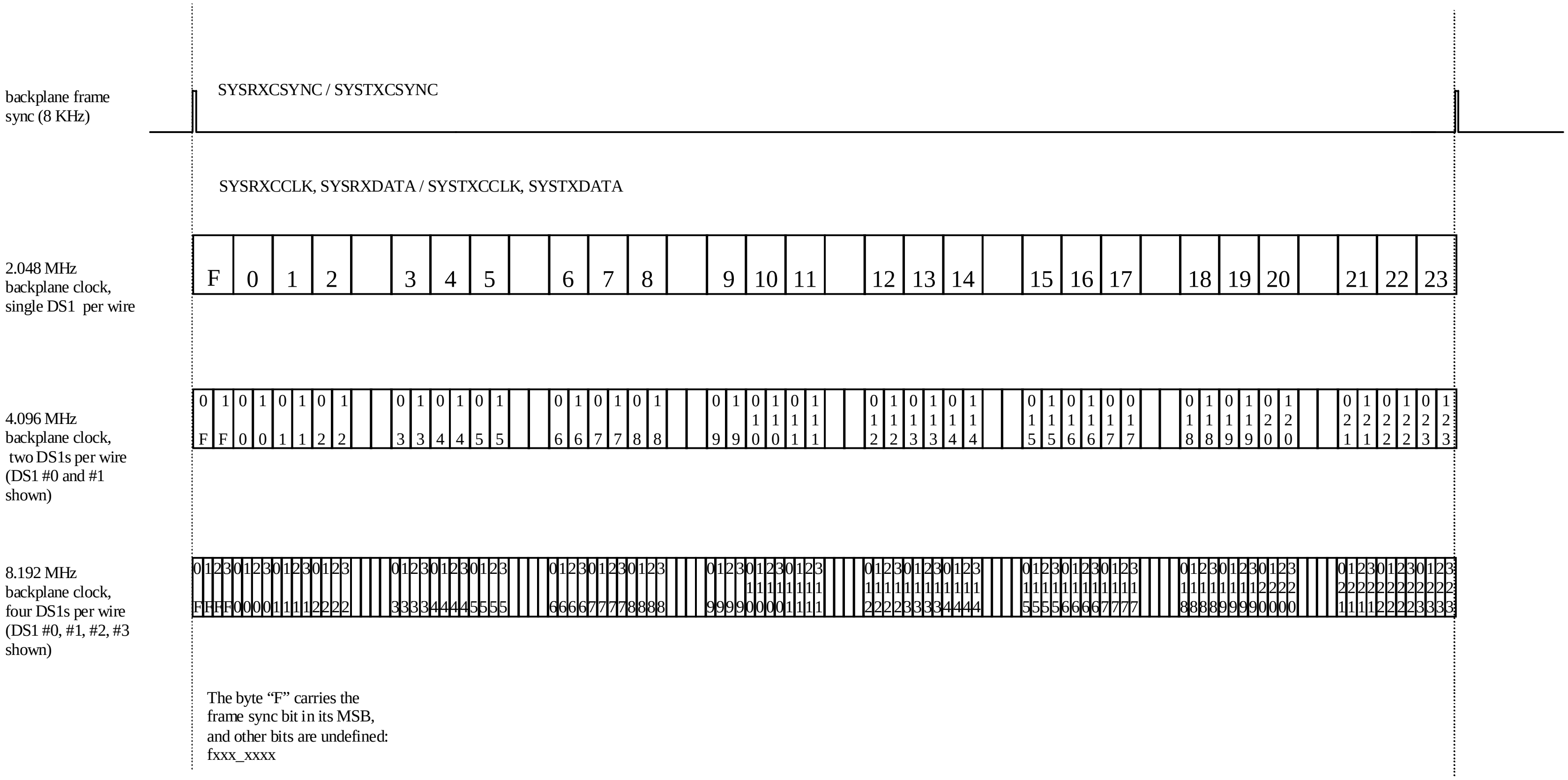


Figure 2.5 High Speed TDM Backplane with Slip Buffer Mode Waveform

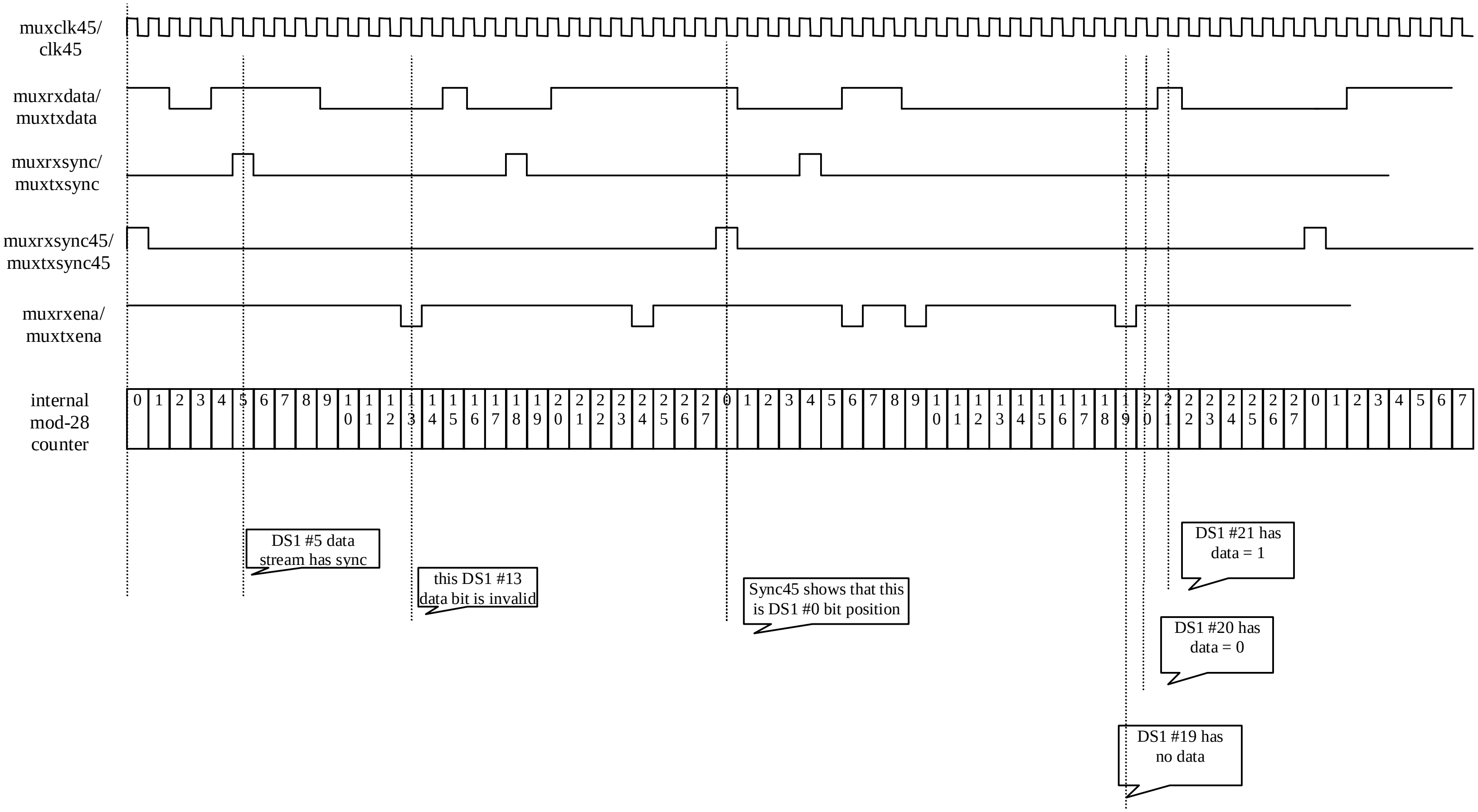


Figure 2.6   Pin-Efficient Muxed Data Waveform

2.7 De-jitter Buffer and Clock Divider

A fully digital clock de-jitter circuit is provided for each of the 28 DS1s. The principle of operation for each circuit is that the nominal 44.736 MHz (CLK45) chip input clock is used as a reference to a DPLL, which divides the reference by a variable. Both edges of CLK45 are used, so the granularity of jitter correction is 11 ns.

The per-channel 72-bit FIFO in the back end logic provides storage for decoupling the instantaneous incoming and de-jittered clock speeds. The jitter attenuator can be selected (on a per-DS1 basis) for insertion in the RX or TX directions, and also may be bypassed.

The VSC9670 de-jitter circuit is fully TR62411 compliant in terms of jitter tolerance and jitter attenuation. Specifically, no bit errors will result with an incoming jitter of 138 UI peak to peak, up to a frequency of 1 Hz, or 28 UI pp at 4.9Hz, and so on. Jitter attentuation follows the required 20 db per decade slope. (A full characterization report is available.)

A local clock divider is provided as a convenience to the user which follows the same principles as in the de-jitter circuitry. This divider eliminates the need for providing a high quality 1.544 MHz clock source to the VSC9670 for generating the TX line-side timing (see Figure 2.2). The internal divider creates a low jitter 1.544 MHz clock by adaptive division of the reference 44.736 MHz by 28 or 29, with an 11 ns granularity of edge position.

The internal divider can be selected as a per-DS1 clock TX direction timing reference instead of the REF1544 input pin.

2.8 Alarms and Performance Monitoring

The RX Alarms and Performance are monitored by a dedicated state machine. There are two RAMs associated with this machine: the flags RAM, which is 2 longwords per channel, and the counts RAM, which is 8 longwords per channel. The flags RAM contains status bits such as OOF and RED alarm. The counts RAM has double buffered versions of required performance monitoring statistics, such as CRC error counts.

The counters are sized to allow at least 1 second in between CPU reads, without overflowing under normal circumstances. Double buffer support is provided to further simplify CPU interfacing. There are two banks of counters: one bank is used as a working copy and the other as a frozen copy. When an internal 1-second counter expires (or when an external 1-second pin is pulsed, or upon CPU command), the VSC9670 switches counter banks between working and frozen, and optionally interrupts the CPU. The CPU then has up to 1 second to read the frozen counts before they are overwritten.

Another function that is performed in this state machine is the shared timing of long duration events, using a “walker” process. The “walker” process is a low priority read-modify-write access into the flags RAM, occurring once every 10 ms. For example, consider the process of detecting the loopup code 00001. The RX FSM uses a sliding window technique to detect the presence of the loopup code over the last 48 ms.

Once the code is detected by the RX FSM, the errors machine performs continuous detection over a period of many seconds, using a timestamp and the walker process. Once the required threshold is reached, the pattern detection is confirmed and an interrupt is optionally generated to the CPU.

The alarms and other conditions described in the following sections are detected using a combination of pattern detection in the RX state machine and longer duration timing and hardware integration in the Performance Monitoring machine.

2.8.1 LOS Alarm

TR62411 specifies that LOS should be declared after 175 ± 75 pulse positions occur without a pulse. Accordingly, VSC9670 will declare LOS after 128 consecutive bit periods with no incoming ‘1’s, and clear LOS upon finding a frame time (125 µs) with at least 16 instances of 8-bit periods, each containing at least a single ‘1’.

Note that LOS can be reliably detected by VSC9670 only when using a bipolar rail connection to the LIU. If the clk/data connection is used instead of the clk/P-rail/N-rail connection, LOS must be detected by the LIU.

A further degree of control is provided to deal with different types of LIUs. If the LIU uses a simple level slicer to recover the clock from the analog twisted pair, then the incoming RX clock of the VSC9670 will be lost when there is a line fault. If the LIU uses a PLL scheme for clock recovery, the incoming P-rail and N-rail data to the VSC9670 will be ‘0’, but the clock will be a free-running local signal.

In the first case, use pat_kill_los = 1, pat_kill_loc = 0; in the second case pat_kill_los = 0, pat_kill_loc = 1. (See Section 3.2.2 RX RAM Registers in the Register Map Section for definitions.)

2.8.2 Red Alarm

TR62411 requires a true hardware integrator to handle intermittent LOS or OOF conditions with different attack and decay slopes. The VSC9670 provides a programmable integrator whose characteristics can be set by three per-chip registers, which control the Red declaration time, Red clearing time, and the ratio of the attack and decay slope.

An example of programming which meets TR62411 requirements is: RED_SET_ADD = 8, RED_CLR_SUB = 2 and RED_CLR_SUB0 = 3. The VSC9670 then declares a Red alarm after 2.5 seconds of sustained LOS or OOF (if not caused by AIS), and clears the Red alarm 12 seconds after the last occurrence of either event. The integrator decays with a slope that is 1/4 the rise slope.

2.8.3 AIS (“Blue”) Alarm

ANSI T1.403-1995 defines AIS (Alarm Indication Signal) as the occurrence of all ‘1’s for a duration of 3 - 75 ms while in OOF state. In order to reliably declare the alarm on a line with occasional bit errors, the ‘1’s density requirement is defined as ≥ 99.9%.

TR62411 does not specify declaration or clearing times. The VSC9670 can be flexibly programmed for various environments.

For example, if AIS_SET_THR = 0 and AIS_CLR_THR = 0, the VSC9670’s AIS detection satisfies the ANSI requirement and will declare and clear AIS within about 50 ms. If AIS_SET_THR = 7 and AIS_CLR_THR = 7, the VSC9670 will declare or clear AIS in 1 second. In all cases, the VSC9670 will abandon AIS detection if framing is recovered.

2.8.4 Yellow Alarm (RAI)

In SF mode, ANSI T1.403 defines Yellow Alarm (Remote Alarm Indication) to be when bit2 of every DS0 octet is 0 for at least 1 second. TR62411 calls for a minimum detection time of 335 ms in the presence of a BER = 1 x 10^{-3}.

For example, if YEL_SET_THR = 7, the VSC9670 SF Yellow detection meets the ANSI requirement and declares Yellow alarm after 1 second. If YEL_SET_THR = 0, the VSC9670 meets the TR62411 requirement and declares SF Yellow within about 50 ms.

ESF Yellow Alarm detection is performed in the CPU by sensing BOC codes received by the FDL machine. See Section 2.9 FDL State Machine

2.8.5 In-Band Loop Codes

In-band loop codes are the following repetitive patterns:
00001, 00011, 001, 00111, 10100.

The VSC9670 can be programmed to sense these patterns either framed or unframed, and the detection time requirement is programmable for each pattern.

For example, ANSI T1.403 calls for 5 seconds of continuous repetition of a framed 00001 pattern to trigger a CSU loopback. By setting CSULU_SET_THR = 32, the VSC9670 can be programmed to detect this code, timed at 5 seconds per ANSI requirements, even in a BER = 1×10^{-3} errored environment.

2.8.6 Idle Signal

ANSI T1.403 specifies the IDLE code as being the DS0-aligned pattern of 00010111. The VSC9670 can be programmed to create an interrupt after a specific period of continuous IDLE pattern detection. For example, set IDLE_SET_THR = 14 for a continuous 2-second IDLE pattern detection. In ESF mode, the FDL requirements must also be satisfied; the CPU must perform the check from the RX FDL FIFO as explained in Section 2.9 FDL State Machine.

2.8.7 Performance Monitoring Counts and Indicators

Various counts are maintained, on a per-DS1 channel basis, in saturating counters that are sufficiently large to accumulate one second’s worth of data under normal conditions. A bank scheme is used so that the in-use bank and spare bank of counters can be switched, either by CPU control or based upon an internal 1-second timer. This allows the CPU to be interrupt driven and have a large latency in interrupt service without danger of the counts being overwritten.

The following counts and indicators are maintained:

- OOF, STKY_OOF:
  The OOF bit is a live indicator of RX Out of Frame status; the STKY_OOF bit gets set upon OOF and is cleared by a CPU write of 1 to the bit position.

- SEFE:
  The Severely Errored Framing Error is set if there is more than one frame bit error in a multiframe while the RX framer was in-frame (following the ANSI T1.403 definition). Depending upon the criteria for declaring OOF, this may or may not result in reacquisition of frame.

- OOF_EVENT[5:0]:
  During the 1-second period, this counts the number of OOF events.

- COFA[3:0]:
  During the 1-second period, the Change of Frame Alignment counter shows the number of times that the off-line framer found a change in frame sync bit position after frame reestablishment.

- FERRCNT[12:0]:
  During the 1-second period, this counts the number of framing bit errors while the RX framer was in-frame.

- CRCERRCNT[11:0]:
  During the 1-second period, this counts the number of ESF multiframes that had at least one CRC6 error while in-frame. The expected maximum value is 333 or 334.

- CODERR[20:0]:
  During the 1-second period, this counts the number of line coding errors (bipolar violations, and also as an option excess 0 strings).

- PN_ERR[20:0]:
  During the 1-second period, this counts the number of PN pattern errors while the PN machine was in-sync.

2.9 FDL State Machine

The FDL state machine is updated by the RX state machine when a new FDL bit is obtained for a given channel, and is requested by the TX state machine to provide an FDL bit for a given channel.

The FDL state machine is responsible for the Bit Oriented Code (BOC) packet message recognition and insertion. A BOC is a 16-bit sequence consisting of 8 ones, a zero, 6 binary code bits and a trailing zero (the format 11111111 0bbbbbb0). If the FDL channel is carrying packet oriented messages, the state machine will perform the following:

- flag detection / bit de-stuffing / CRC-16 checking in the RX direction
- flag insertion / bit stuffing / CRC-16 calculation in the TX direction

There is also a “transparent mode” in which the machine will simply hand off RX direction FDL bits to the CPU, and insert TX direction FDL bits from the CPU, without any processing.

When the RX FDL machine is enabled from reset, it begins to look for a flag pattern (0111 1110 binary) or an abort pattern (1111 111 binary, that is, 7 ones). If a flag is received, the machine searches for a valid HDLC frame (some more data bytes, terminated by a closing flag 7E hex). If a sequence of 8 ones is received, the machine searches for valid bit oriented codes (1111 1111 0bbb bbb0 binary, where the 6 bit information field is shown as “b”).

2.9.1 Bit Oriented Codes

The VSC9670 provides sophisticated hardware support for BOCs (bit oriented codes) in the form of continuous-n-times-detection. That is, when several consecutive identical bit-oriented codes are detected, the CPU is notified in a compact {count, code} format.

The logic is as follows: the VSC9670 sends FDL received data to the CPU by means of the RX FDL FIFO. The FIFO data format is a 2-octet structure; the first octet is an identification word (explained below), which includes the number of times the BOC was repeated, and the second octet is the BOC itself. The first time a BOC is received, a CPU update is made after 10 repetitions to reduce latency. Subsequent updates are made after 31 detects.

This can be thought of as a way of compressing the incoming BOC stream to simplify CPU processing. The CPU reads the RX FIFO to pick up the new BOCs received since the last time the FIFO was read. A CPU interrupt can be set each time a valid BOC is received, or at a high water mark in the RX FIFO.

Hardware filtering is built into the BOC framing algorithm in order to absorb the effect of occasional line errors. The reception of three consecutive error-free BOC templates (1111_1111_0bbb_bbb0) establishes the BOC frame. Two bit errors anywhere in three consecutive 16-bit BOC frames will cause a loss of the BOC frame. The algorithm is geared to an environment where the transmitting end sends each BOC repeated at least 10 times. This hardware filtering of occasional bit errors reduces the CPU burden, and increases the likelihood of correctly reporting the intended BOC with the intended number of repeat counts.

Hardware support is also provided for TX direction continuous-n-times-transmission of BOCs. In the TX FDL FIFO, the CPU writes a 2-octet message consisting of an ID word, the number of times the code should be repeated (max 31), and the BOC code itself. The chip parses this 2-byte structure and outputs the desired code. When the TX FIFO is empty, the output FDL bits are selectable to be either all 1, or all flag (hex 7E). There is also a convenient way of inserting a background BOC code with occasional HDLC messages, as described in the register map.

The ID, counter and bit oriented code storage in the RX / TX FIFOs are explained pictorially later in this document.

Note: BOC codes written by the CPU into the TX FIFO are sent out LSB first. For octets read by the CPU from the RX FIFO, the LSB was the first bit received from the line.

2.9.2 Packet Oriented Messages

As described earlier, the RX FDL machine can delineate HDLC packets by searching for flag octets (hex 7E), and bit de-stuffing the data stream demarcated by two flag octets. The RX FDL machine also calculates and checks CRC-16, and checks for octet alignment errors. In addition, received CRC-16 is always presented to the CPU.

The TX FDL machine will generate HDLC packets by enclosing a CPU generated packet in flag octets (hex 7E), optionally calculating and adding a CRC-16, and bit stuffing the payload. The format of an HDLC message is:

FLAG(0X7E)	DATA	DATA	. . . .	FCS[15:8]	FCS[7:0]	FLAG(0X7E)

The length of the data field is essentially unlimited, though in practice it may be small.

Note: HDLC data bytes written by the CPU into the TX FIFO are sent out LSB first. For HDLC data bytes read by the CPU from the RX FIFO, the LSB is the first bit received from the line.

2.9.3 FDL FIFO Structure (“Sub-packet” Format)

For maximum flexibility, and for allowing interleaving of BOC and HDLC messages, a “sub-packet” format data structure is used in the RX and TX FDL FIFO’s.

A sub-packet is a block of data introduced by an ID byte. Sub-packets are a programming construct, intended to allow easy data storage and access into the VSC9670’s FIFOs. Sub-packets are composed of pieces of HDLC packet data (or BOC messages) and FDL control information. HDLC packets are formed from the information embedded in the sub-packets. This “in-line control” approach maximizes storage efficiency in the FIFO and eases CPU processing.

The sub-packet format is applicable to both bit oriented codes and packet oriented messages. In the BOC case, the sub-packet is a fixed length 2-octet data structure, consisting of the ID byte, a count of the number of times the BOC was received, and the BOC itself.

In the packet oriented message case, the HDLC packet is broken up into arbitrary sized fragments (1 to 31 bytes per fragment), and the sub-packet is defined as consisting of the ID byte and the fragment of data.

The per-channel RX and TX FIFOs are maintained by the use of read and write pointers. As an example, the RX FIFO is shown in Figure 2.7, with FIFO size = 22.

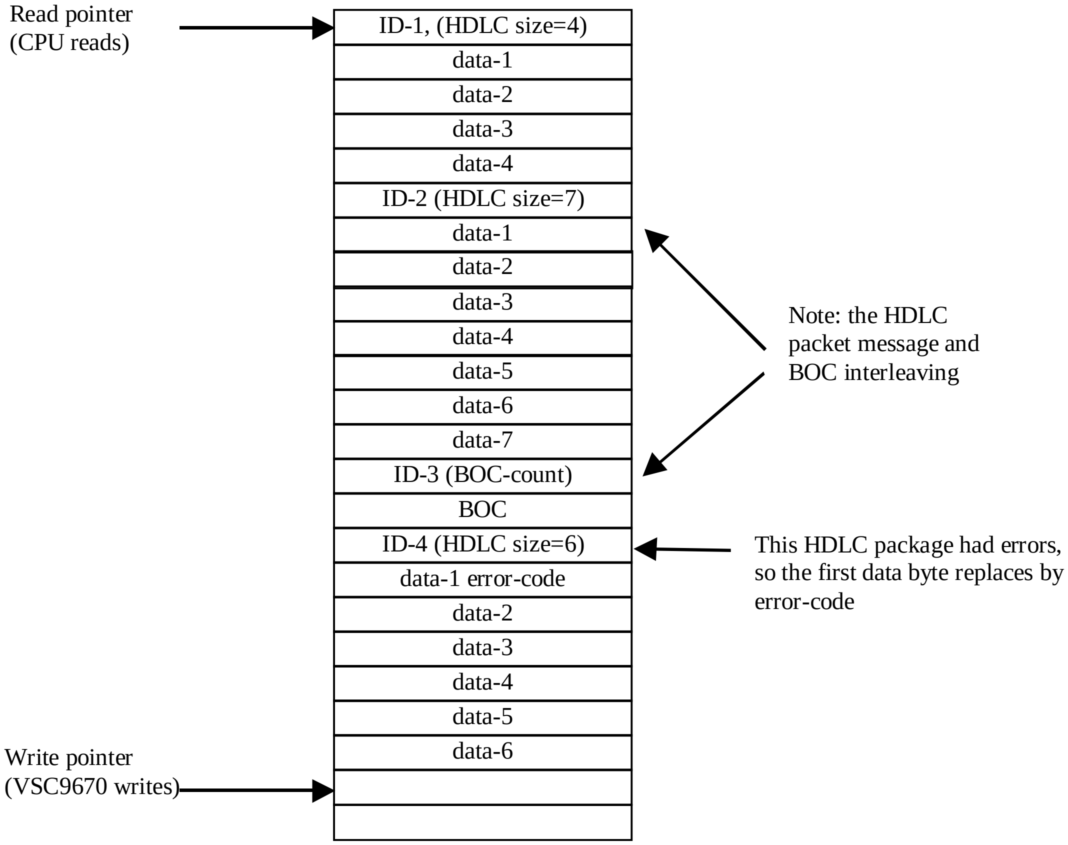


Figure 2.7 FDL RX FIFO Example

The RX FDL FIFO is shown in Figure 2.8 with the on-chip read and write pointer registers. The FIFO contains 4 sub-packets, each introduced by an ID byte.

Assume the CPU has been interrupted and is ready to service the RX FDL FIFO:

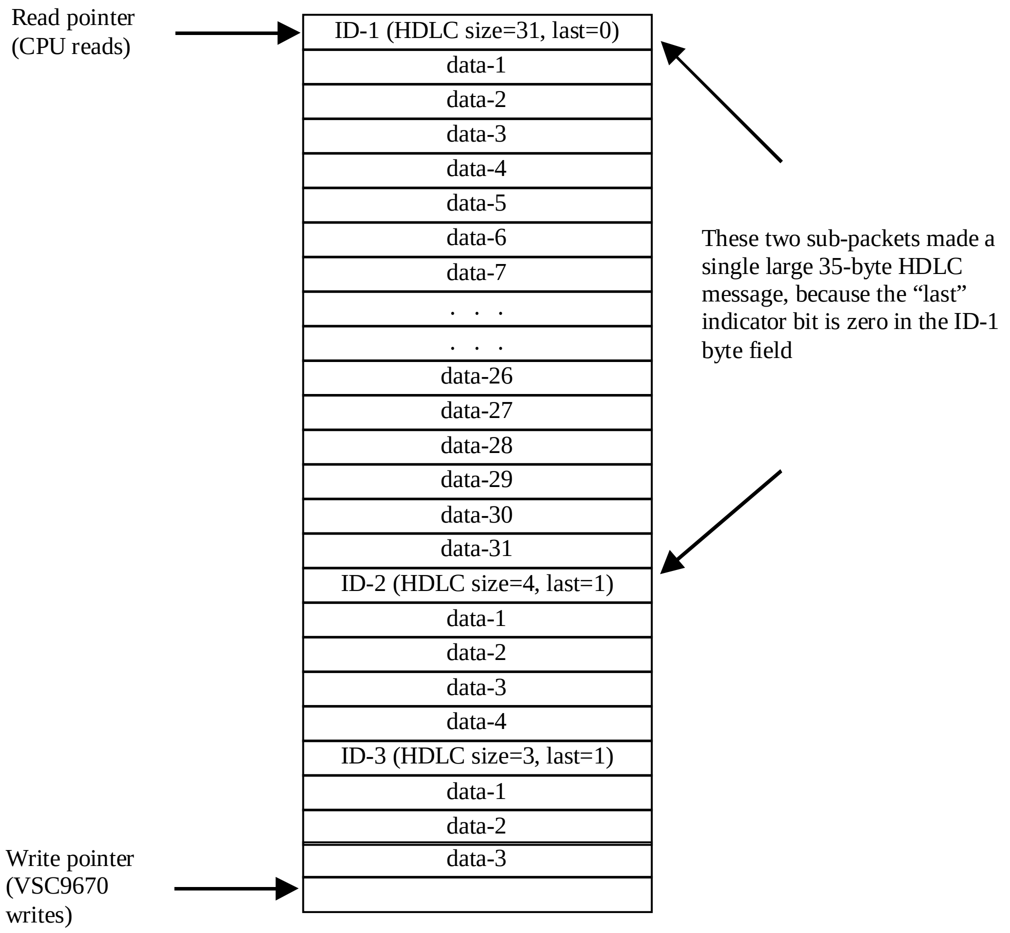
- The CPU first reads the FIFO size (in this case, 22), and knows that there are 22 new octets in the FIFO since the last read.
- The CPU will perform several reads to the RX FDL FIFO read location (see register map for details), and will be given bytes out of the RX FDL FIFO in succession.

The read and write pointers are maintained by the VSC9670.

The first octet the CPU reads will always be an ID byte. The ID byte contains a bit field that describes whether this sub-packet is a BOC type or HDLC type, and another bit field that specifies the number of bytes to follow in this sub-packet, allowing the CPU to know the location of the next ID byte. For example, after reading and parsing ID-1 (with size field = 4), the CPU knows that the fifth byte after this will be another ID byte.

In Figure 2.8, therefore, there are two HDLC messages (the first of size 4 bytes, the second of size 7 bytes), then a BOC, and then another HDLC message (starting with ID-4 byte). The last HDLC message had an error such as CRC mismatch, abort, octet misalignment, or packet-too-short. ID-4 will have a bit field set to indicate that this HDLC packet had errors, and that error information is available in the “error-code” byte. This error-code byte overwrites the first payload data byte as shown.

In order to permit HDLC messages to be longer than the limit allowed by the size bit field in the ID byte, a “last” indicator bit field is provided as part of the ID byte. If the “last” bit is set, this sub-packet is the last portion of the HDLC message. An HDLC message of arbitrary size can therefore be accommodated.



These two sub-packets made a single large 35-byte HDLC message, because the “last” indicator bit is zero in the ID-1 byte field

Figure 2.8 FDL RX FIFO Example: Sub-packet Concatenation

There are three sub-packets here: the first two comprise a single large (35-byte sized) HDLC message, and the third comprises a short (3-byte sized) HDLC message.

As can be seen, the size of the sub-packets is variable, and will be either the size of the HDLC frame data portion or a maximum of 31. It can be seen that the embedded control sub-packet concept is flexible, and comes with relatively low overhead. Bit oriented code sub-packets and HDLC message sub-packets can be simultaneously present in the FIFO.

In the TX direction, the FDL state machine can be programmed to output bit oriented codes or HDLC format messages. When the TX FDL FIFO is empty, the fill pattern can be selected to be all ‘1’s or the flag pattern (7E hex). When the CPU wants to transmit a BOC or HDLC message, it first checks to see how much space is available in the TX FDL FIFO. Then it creates and writes a sub-packet into the TX FDL FIFO.

After the CPU indicates to the TX FDL state machine that a complete sub-packet has been written into TX FDL FIFO, the FDL machine parses the sub-packet, sends out the data field and optionally interrupts the CPU when done. Even before the first sub-packet is finished, the CPU can load several other sub-packets into the FIFO, subject only to the constraint that each complete sub-packet must fit into FIFO memory.

Note that there is no overflow checking in the TX FIFO. The CPU must first ensure there is enough space before embarking on the write.

For example, to send out 5 repetitions of a specific BOC and an 11-byte HDLC message, the CPU writes TX FDL FIFO as shown in Figure 2.9.

The TX FDL state machine parses the ID-1 byte and sends out the bit oriented code 5 times, as requested. Then it parses ID-2 byte and sends out an HDLC packet consisting of 11 data bytes by prefacing them with the 7E flag, adding a CRC-16, bit stuffing as necessary, and adding one or more trailing 7E flag bytes

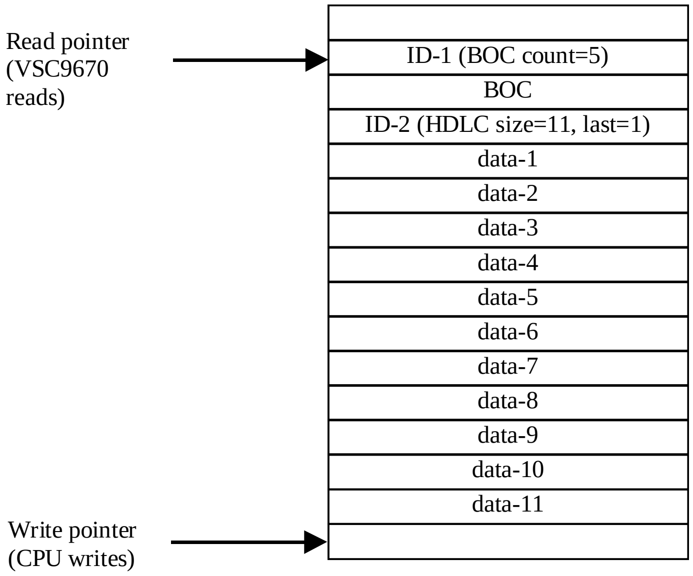


Figure 2.9 FDL TX FIFO Example

.In the case where the data portion of a HDLC message is longer than the length that can be specified in the ID byte, the message must be broken up into several sub-packets. The ID byte of the last sub-packet specifies that it is the end of the HDLC message, and hence the CRC-16 is optionally added after that sub-packet.

To meet the requirements of the CPU interface, the TX FDL always waits until a sub-packet is completely written before beginning to process that sub-packet. If an HDLC message spans more than one sub-packet, the CPU must ensure that subsequent sub-packets are available to the VSC9670 in time, or an underflow will result. Therefore it is recommended that the first sub-packet should be made as large as possible to allow adequate time for the CPU to insert additional sub-packets into the TX FDL.

The Register Map section has more information on the bit field encoding of the ID byte. The various command and status bits are also documented there.

2.10 Signaling State Machine

The signaling state machine and RAM provide a centralized resource for both RX and TX direction processing of robbed bit signaling extraction and insertion.

For the RX direction, per channel, there are 24 DS0s for which robbed bit signaling needs to be extracted, buffered, and presented to the CPU. Per DS0 standards, there are two options: signaling bit delay mode, and signaling bit filter mode.

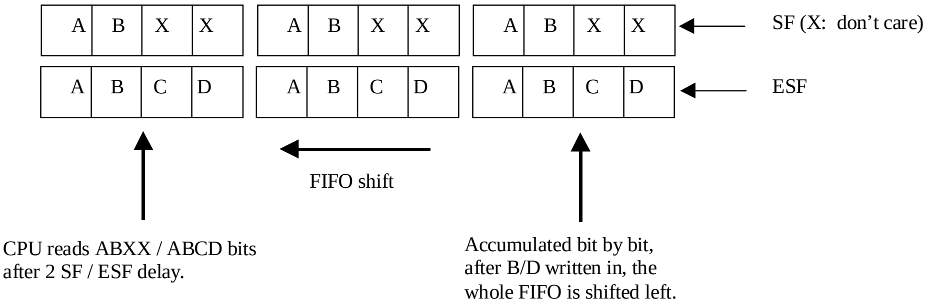


Figure 2.10 RX Signaling, Delay Mode

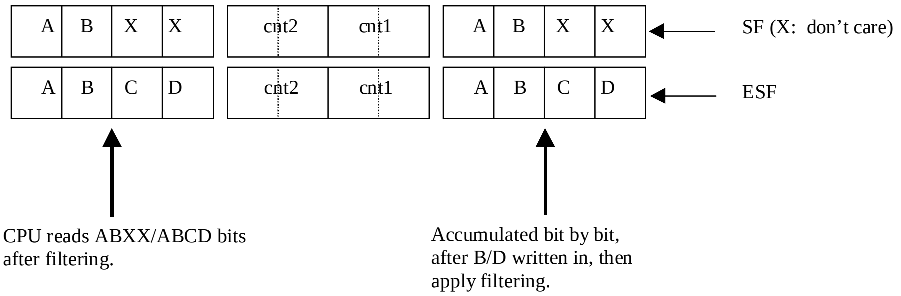


Figure 2.11 RX Signaling, Filter (Debounce) Mode

As shown, there are two options for signaling bit conditioning in the RX direction: a simple delay of 2 super-frames or extended-super-frames, or a more complex debounce. In either case, updating the signaling bits from the RX state machine stops when the DS1 is OOF so that the previously stable AB or ABCD bits can be frozen.

In the delay mode, the robbed signaling bits are accumulated one by one into the accumulator location (the A bit is received first, D bit is received last, in ESF mode). After the last signaling bit is received, the accumulated nibble is shifted as shown. If the CPU-output nibble location changes as a result of this shift, an interrupt is optionally generated. In SF mode, where there are only 2 signaling bits A and B; the C and D bit locations shown in the figure are to be ignored.

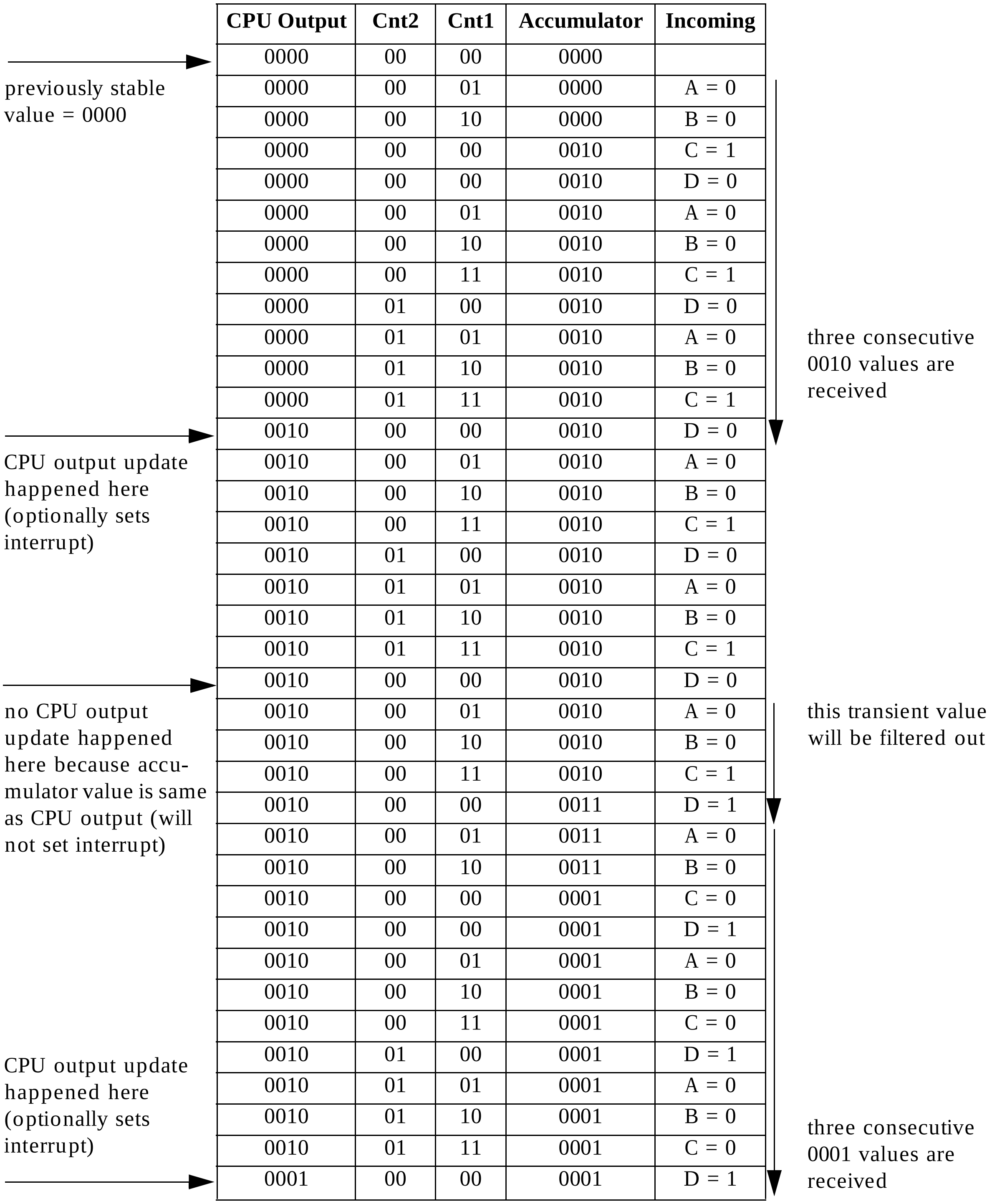
In filtering (debounce) mode, the accumulator is used to collect signaling bits one by one, and there are two 2-bit counters - cnt1 and cnt2. When a stable, filtered AB or ABCD value is present in the accumulator, it is copied into the CPU-output location.

An example of the filtering operation is shown in Figure 2.12. In this example, THRESHOLD was set to 1, meaning that three consecutive and identical new values of ABCD bits have to be received before the CPU is updated (with an optional interrupt). The filtering threshold can be changed, so that the CPU is updated after 2 to 5 consecutive identical values are received.

If the DS1 is SF format rather than ESF, the processing shown as happening after the D bit is received will instead happen after the B bit is received.

For the TX side, the CPU simply writes a 4 bit field with the ABCD bits that must be injected into the outgoing frame, together with a 1 bit field which enables insertion. The hardware will copy the CPU-written ABCD bits into an active location in an SF or an ESF synchronized way prior to use. This avoids the danger that the outgoing robbed signaling bit stream will contain part of the previous value and part of the current value.

The signaling RAM bits are described in further detail in Section 3.6.2 Signaling RAM Registers.



CPU Output	Cnt2	Cnt1	Accumulator	Incoming
0000	00	00	0000	
0000	00	01	0000	A = 0
0000	00	10	0000	B = 0
0000	00	00	0010	C = 1
0000	00	00	0010	D = 0
0000	00	01	0010	A = 0
0000	00	10	0010	B = 0
0000	00	11	0010	C = 1
0000	01	00	0010	D = 0
0000	01	01	0010	A = 0
0000	01	10	0010	B = 0
0000	01	11	0010	C = 1
0010	00	00	0010	D = 0
0010	00	01	0010	A = 0
0010	00	10	0010	B = 0
0010	00	11	0010	C = 1
0010	01	00	0010	D = 0
0010	01	01	0010	A = 0
0010	01	10	0010	B = 0
0010	01	11	0010	C = 1
0010	00	00	0010	D = 0
0010	00	01	0010	A = 0
0010	00	10	0010	B = 0
0010	00	11	0010	C = 1
0010	00	00	0011	D = 1
0010	00	01	0011	A = 0
0010	00	10	0011	B = 0
0010	00	00	0001	C = 0
0010	00	00	0001	D = 1
0010	00	01	0001	A = 0
0010	00	10	0001	B = 0
0010	00	11	0001	C = 0
0010	01	00	0001	D = 1
0010	01	01	0001	A = 0
0010	01	10	0001	B = 0
0010	01	11	0001	C = 0
0001	00	00	0001	D = 1

previously stable value = 0000

three consecutive 0010 values are received

CPU output update happened here (optionally sets interrupt)

no CPU output update happened here because accumulator value is same as CPU output (will not set interrupt)

this transient value will be filtered out

CPU output update happened here (optionally sets interrupt)

three consecutive 0001 values are received

Figure 2.12 RX Signaling Debounce Example

2.10.1 Signaling Bus

RX signaling information can be driven to VSC9670 signaling bus pins, and TX signaling information can be received from an external device using VSC9670 pins. This is useful for architectures where the signaling information is not intended to be terminated (as in the CPU access method above), but rather continued through in a TDM fashion (in a DACS, for example).

The VSC9670 signaling bus feature can only be used in Backplane Slip Buffer mode because of pin count limits. It is expected that most signaling bus users will have a common backplane clock/sync architecture.

As with the data pins, the signaling bus pins can be run at 1.544 MHz, 2.048 MHz, 4.096 MHz or 8.192 MHz. This provides a great deal of flexibility in passing TDM data and robbed bit signaling through similar switching paths.

The RX (TX) signaling bus uses SYSRXCCLK and SYSRXCSYNC (SYSTXCCLK and SYSTXCSYNC). The format and timing of the signaling bus is identical to the format and timing of the data bus SYSRXDATA or SYSTXDATA in backplane mode, as shown in Figure 2.5.

The data that is presented in a given DS0 is
:

SF	-	-	-	-	A	B	-	-

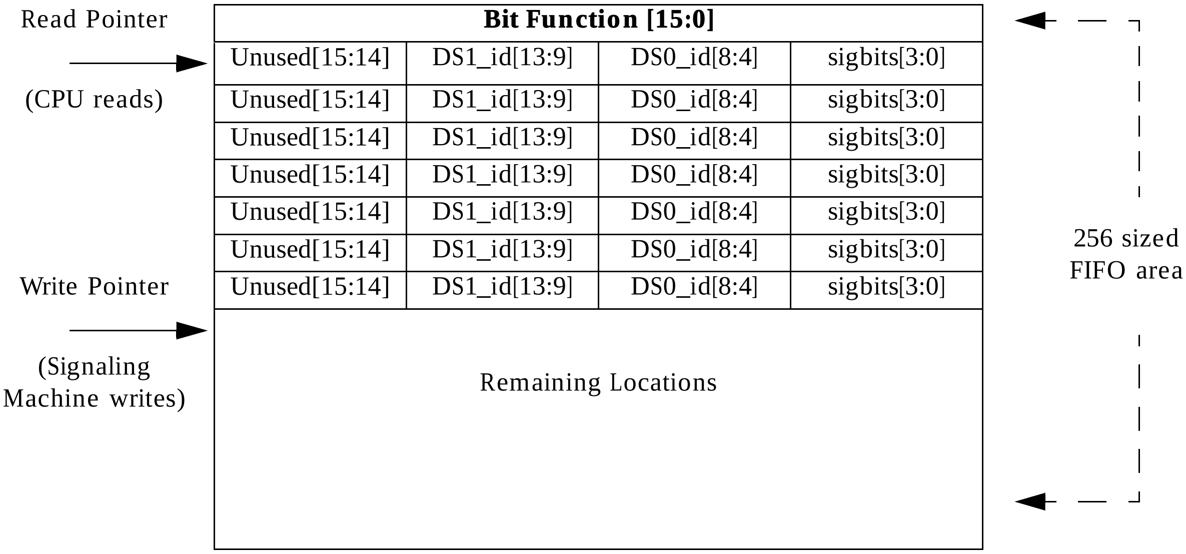
("-" indicates "don't care")

ESF	-	-	-	-	A	B	C	D

first bit last bit

2.10.2 RX Signaling FIFO

The RX Signaling FIFO is a way for the programmer to obtain quickly an update of recently received signaling messages. A 256 entry FIFO is maintained by the hardware, with the write pointer driven by received signaling (after delay or debounce), and the read pointer maintained by the CPU.



Bit Function [15:0]			
Unused[15:14]	DS1_id[13:9]	DS0_id[8:4]	sigbits[3:0]
Unused[15:14]	DS1_id[13:9]	DS0_id[8:4]	sigbits[3:0]
Unused[15:14]	DS1_id[13:9]	DS0_id[8:4]	sigbits[3:0]
Unused[15:14]	DS1_id[13:9]	DS0_id[8:4]	sigbits[3:0]
Unused[15:14]	DS1_id[13:9]	DS0_id[8:4]	sigbits[3:0]
Unused[15:14]	DS1_id[13:9]	DS0_id[8:4]	sigbits[3:0]
Unused[15:14]	DS1_id[13:9]	DS0_id[8:4]	sigbits[3:0]

Figure 2.13 On-chip RX Signaling FIFO Block

The signaling state machine updates the signaling FIFO using the same conditions that cause an update of the CPU-active read nibble location. In other words, in delay mode the condition is the 2 multiframe delay, and in the debounce mode the condition is the required number of consecutive matching nibbles.

The CPU drives the read pointer after reading the FIFO information at the current location. The format of the stored information is:

Bit Function															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Unused		DS1_id[4:0]					DS0_id[4:0]					sigbits[3:0]			

Thus the RX signaling FIFO provides a chronological history of received signaling messages. Note that at times of peak activity, depending upon the rate of CPU reads, the FIFO may saturate at a fill level of 255 and subsequent signaling messages will be lost. (Hardware automatically protects against a FIFO overflow by saturating instead.)

An error status indication is provided in this case, and normal operation automatically resumes as the CPU frees up FIFO space by advancing the read pointer. The SIG_FIFO_STAT, SIG_FIFO_1 and SIG_FIFO_0 registers are used to access and control the RX signaling FIFO block as described in Section 3.6.1 Per Chip Registers.

2.11 Loopback Modes

The following types of loopback are supported, with per DS1 (or per DS0) enables.

- Local loopback: system TX data outgoing to the line is copied back into the system RX direction
- Line loopback: RX data incoming from the line side is copied back into the TX direction
- DS0 resolution line loopback: On a DS0 basis, selected DS0s out of the incoming DS1 RX data are copied into the corresponding TX direction DS0 timeslots.

The following figures illustrate the three loopback modes for a single DS1 slice.

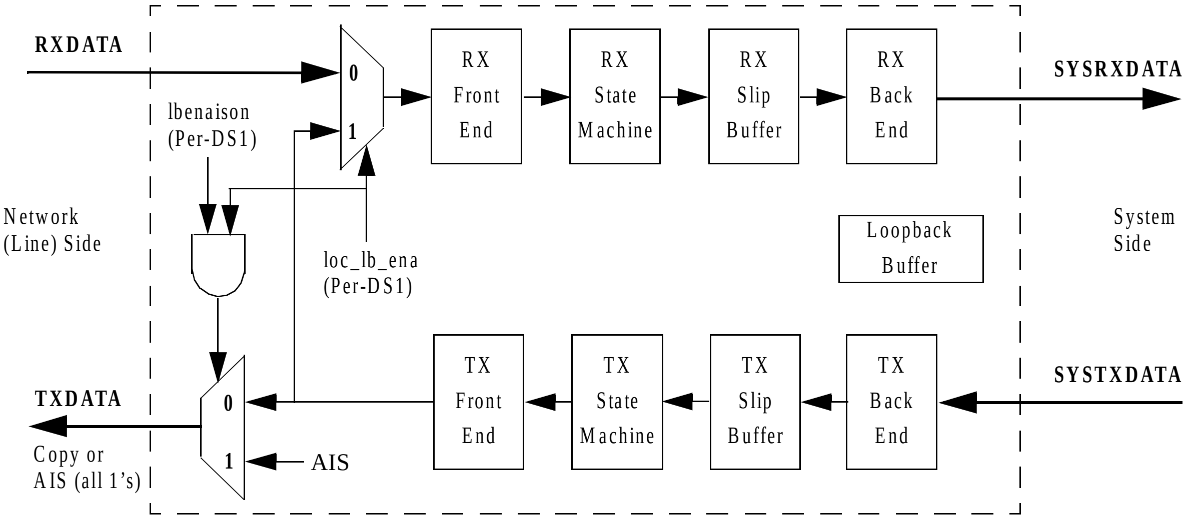


Figure 2.14 Per-DS1 Compliant Local Loopback

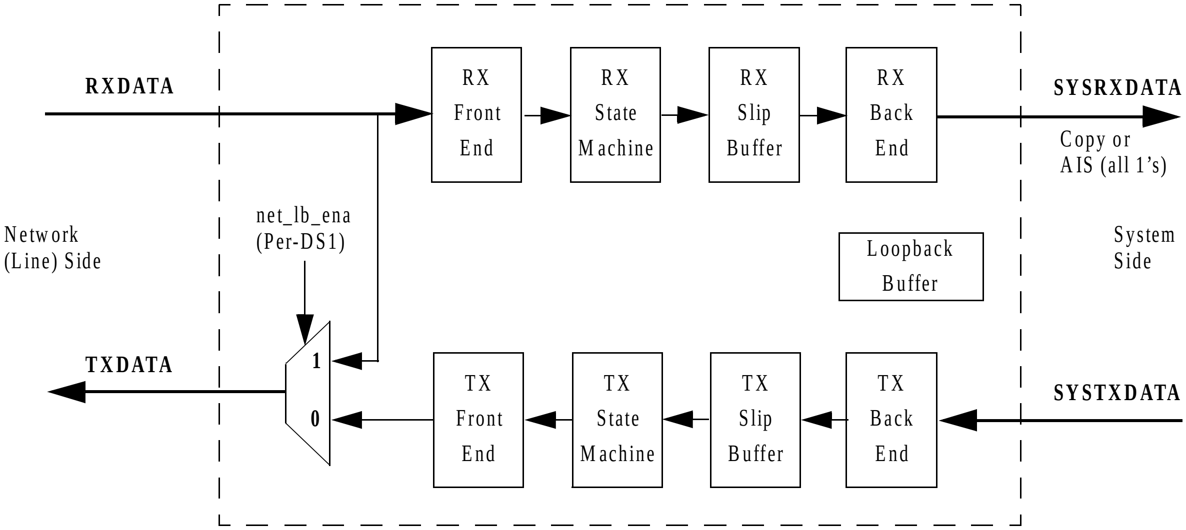


Figure 2.15 Per-DS1 Compliant Network Loopback

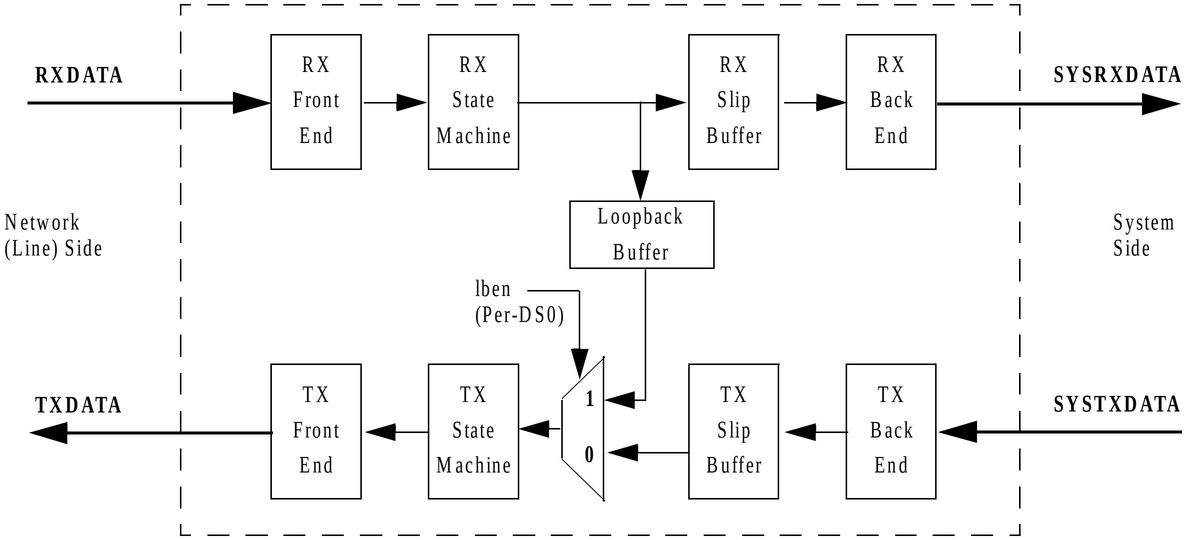


Figure 2.16 Per-DS0 Compliant Network Loopback

The per-DS1 compliant local loopback is implemented very close to the line-side pins. Outgoing TX direction traffic (data and timing) can be copied, on a per-DS1 compliant basis, into the incoming RX direction. The outgoing TX data can be left unmodified or converted into all ‘1’s (AIS).

The per-DS1 compliant network (line) loopback is also implemented close to the line-side pins. Incoming RX direction data and timing can be copied, on a per-DS1 compliant basis, to the outgoing TX direction pins. The RX traffic going to the RX system side can be left undisturbed or converted to all ‘1’s.

The per-DS0 compliant network loopback is implemented in a powerful and general way by using a dedicated loopback buffer, similar to a slip buffer. This buffer contains 2 frames of per-DS1 compliant buffering, independently for all 28 DS1 streams. RX direction traffic is copied into this loopback buffer in parallel with the traffic’s normal path to the back end pins.

In the TX direction, if loopback is enabled on a particular DS0, the TX state machine takes data from the loopback buffer instead of the regular TX back end or TX slip buffer. This is similar to a payload loopback if all DS0s in a DS1 are set to loopback.

The per-DS0 compliant loopback is allowed for all clocking modes (refer to Section 2.5 Clocking Schemes for more details). If a clock speed difference causes the loopback buffer read and write pointers to cross over, the buffer will re-center itself, though not necessarily in a frame-aligned way.

Note that SF or ESF level alignment (12-frame or 24-frame alignment) between RX and TX sides can not be guaranteed, which means that DS0 compliant loopback or full payload loopback will not preserve robbed bit signaling.

2.12 Alarms and Interrupts

The VSC9670 interrupt structure provides a single interrupt pin to the CPU, with programmable polarity (active high level or active low level). The following events can cause CPU interruption:

Alarms: Alarms are defined as unexpected transmission conditions that require immediate CPU attention. The following alarms are detected per DS1, and the CPU is optionally interrupted.

- RX into LOS, out of LOS
- RX into AIS, out of AIS
- RX into OOF, out of OOF
- RX into RED, out of RED
- RX into YELLOW, out of YELLOW
- RX FDL ESF YELLOW
- RX JT1 YELLOW
- RX Slip Buffer, frame aligned slip
- TX Slip Buffer, frame aligned slip

Errors: Errors are considered less severe than alarms, and also include some hardware architectural conditions such as internal FIFO overflows. The following errors are detected per DS1, and CPU interrupt with masking capability is provided.

- RX Front End, overflow
- TX Front End, underflow

Events: Events are not exceptional conditions, and maskable interrupts are provided so that the software can use an interrupt driven methodology.

- 1 second counter event
- RX FDL FIFO reached High Water Mark or on every sub-packet
- TX FDL FIFO reached Low Water Mark or on every sub-packet
- RX robbed bit signaling logic found new AB or ABCD value
- RX pattern recognized (loop-up / loop-down / pseudo-random sync)

The CPU interrupt service routine must take the following steps:

1. Read a single-byte register, the chip interrupt summary register, which shows whether an alarm, error, or event is pending.
2. Read some or all of the 28 interrupt source registers, which show, on a per-DS1 basis, which section (RX FSM, TX FSM, FDL, signaling, etc.) needs attention.
3. If necessary, read register(s) in a specific DS1's section to further isolate the problem.
4. If necessary, address the underlying cause (for example, read FDL data out of the RX FDL FIFO, and bring it below the high water mark).
5. If necessary, write-1-to-clear the per-section interrupt flags. Some interrupt conditions automatically clear themselves after Step 4.
6. Repeat until all interrupt flags are clear.

The register map section describes all relevant bits.

2.13 CPU Interface

The CPU interface communicates read/write updates to each of the other sections by means of an internal 16-bit address / 32-bit data bus. The CPU interface logic adapts the internal address and data buses to a simple generic 8-bit micro-controller port. This method allows the VSC9670 to interface to a wide variety of CPUs (both Intel and Motorola families) and also conserves chip pins.

All simple registers (non-RAM locations) inside the VSC9670 are located in the lower 256 byte addresses (address 0x0000 to 0x0100). The internal RAMs in the VSC9670 are accessed through indirect addressing.

Byte sized indirect address / indirect data / indirect command registers are provided in the lower address range (see Section 3.1 CPU Interface Section Registers for details).

Since CPU access to the internal RAMs is low priority (the TDM processes have the highest priority), the indirect access polling-based method is also valuable for decoupling the microcontroller from the delays in internal RAM access.

The state machines are designed so that state RAMs are not completely occupied by the TDM process, and the CPU can usually get access within 8 clock cycles at 44.736 MHz. Given that the external microcontroller usually runs much slower than this, the desired internal RAM access will immediately poll ‘finished’. Also note that since the internal data bus is 32 bits wide, all indirect reads provide 4 bytes of read data, and writes require 4 bytes of write data.

Byte enables are also provided where appropriate.

2.14 JTAG Functional Description

The VSC9670 supports the Boundary Scan Specification as described in the IEEE 1149.1 standards. The TAP (Test Access Port) controller is a standard 5 wire serial test interface. This TAP controller supports all the required and optional instructions of the IEEE 1149.1 specification. It allows the user to control the width of the instruction register so that additional ones can be added. The architecture of the TAP Controller is shown below in Figure 2.17.

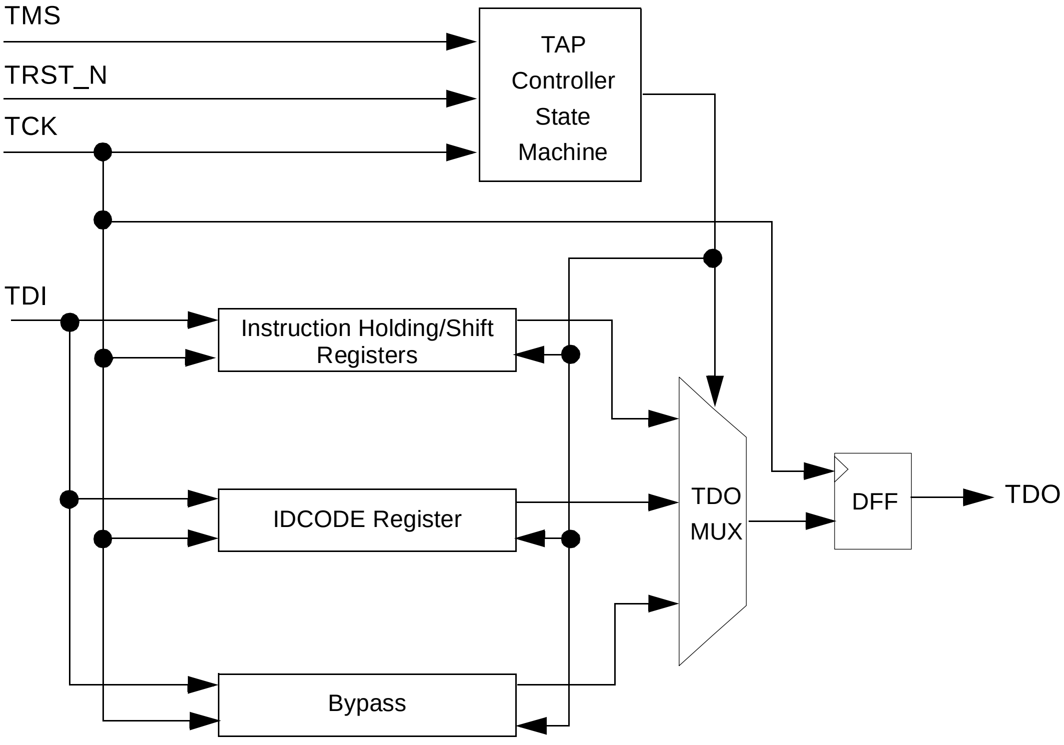


Figure 2.17 TAP Controller Architecture

2.14.1 Detailed Block Diagram for TAP Controller

The instruction shift register is used to shift in a new instruction to the TAP controller while the instruction holding register holds the current instruction. As with all the shift registers in the IEEE 1149.1 specification, the shift is performed on the rising edge of TCK. The instruction shift register shifts only when the state machine is in the Shift-IR state. The capture value for the instruction shift register is always 0x01 when in the Capture-IR state.

When TRST_N is asserted or the state machine moves to the Test-Logic/Reset state, the selected instruction is reset to the IDCODE instruction if present, otherwise it is set to the BYPASS instruction.

Like all other update registers in the IEEE 1149.1 specification, the instruction holding register is updated on the falling edge of the TCK. The instruction holding register is only updated when the state machine is in the Update-IR state.

The BYPASS register is a one bit shift register that shifts in the Shift-DR state and captures the Capture-DR state with a value of 0x1. The BYPASS instruction will always select this register, as will any other undefined instruction.

The IDCODE register is 32 bit register that shifts out the identification code for the device. The register shifts in the Shift-DR state and loads the programed ID code in the Capture-DR state.

The entire TAP controller is controlled by a single state machine with 16 states. The state diagram for this state machine is shown in Figure 2.18. All state changes are with respect to the rising edge of the clock TCK.

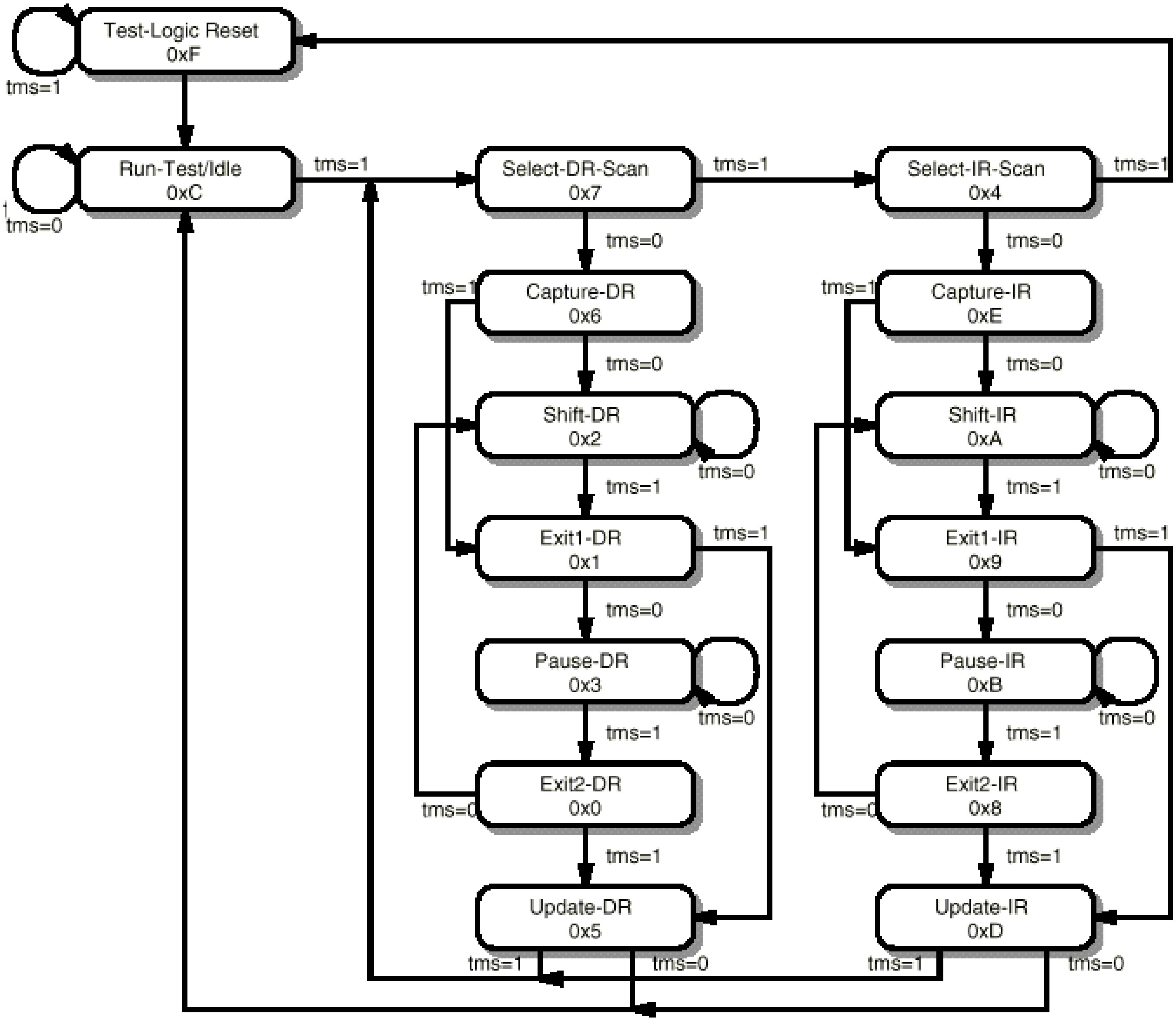


Figure 2.18 TAP Controller State Machine Diagram

3.0 Register Map

This section describes the registers used by the different subsystems of the VSC9670, such as the CPU interface, the RX state machine, the TX state machine, the error machine, and so on. Registers can be categorized as scalar registers (directly mapped into the bottom 256 addresses) and RAM registers (accessible through indirect read/write).

The following abbreviations are used to describe register bits:

R/W	Read/Write bit
RO	Read Only (CPU writes are ignored)
WO	Write Only (CPU reads return undefined values)
W1C	Write-1-to-Clear (writing 0 leaves value unchanged)
P	State machine private bits. (CPU must write them only during provisioning.)

Unless otherwise specified, all bits are read/write, the power-up default value of all scalar registers is 0, and the power-up default value of all RAM registers is unknown.

All the directly accessible registers (the bottom 256 addresses) in the system are listed below, with their address locations and register names.

Table 3.1 List Of Directly Mapped Registers

Address (Hex)	Register Name	Function
0xFF	CPUDIAG	CPU Diagnostic Control
0xFE - 0xF2		Undefined (reserved)
0xF1	HALFMS_HSCALER	Half Millisecond High Byte Scaler
0xF0	HALFMS_LSCALER	Half Millisecond Low Byte Scaler
0xEF - 0xE9		Undefined (reserved)
0xE8	IDLECODE	Idle Code Value (for TX Direction)
0xE7 - 0xE0	DMWVAL_[7:0]	Digital Milliwatt Octets Value (for TX Direction)
0xDF - 0xD8		Undefined (reserved)
0xD7	INTIDT3	Interrupt Indirect Data bits [31:24]
0xD6	INTIDT2	Interrupt Indirect Data bits [23:16]
0xD5	INTIDT1	Interrupt Indirect Data bits [15:08]
0xD4	INTIDT0	Interrupt Indirect Data bits [07:00]
0xD3	INTIAD1	Interrupt Indirect 16-bit Address bits [15:08] (see Table 3.3)
0xD2	INTIAD0	Interrupt Indirect 16-bit Address bits [07:00] (see Table 3.3)
0xD1	INTICMD	Interrupt Indirect Command/Status Control
0xD0		Undefined (reserved)
0xCF	RX2REF_SEL	Select RXCLK to drive REF1544
0xCE	DJB_EN_27_24	De-Jitter Buffer Enables (DS 1 Channel [27:24] for either TX or RX direction)
0xCD	DJB_EN_23_16	De-Jitter Buffer Enables (DS 1 Channel [23:16] for either TX or RX direction)
0xCC	DJB_EN_15_08	De-Jitter Buffer Enables (DS 1 Channel [15:08] for either TX or RX direction)
0xCB	DJB_EN_07_00	De-Jitter Buffer Enables (DS 1 Channel [07:00] for either TX or RX direction)
0xCA	DJB_RX_27_24	De-Jitter Buffer RX Control (DS 1 Channel [27:24])
0xC9	DJB_RX_23_16	De-Jitter Buffer RX Control (DS 1 Channel [23:16])
0xC8	DJB_RX_15_08	De-Jitter Buffer RX Control (DS 1 Channel [15:08])
0xC7	DJB_RX_07_00	De-Jitter Buffer RX Control (DS 1 Channel [07:00])

Table 3.1 List of Directly Mapped Registers (continued)

0xC6	DJB_TX_27_24	De-Jitter Buffer TX Control (DS 1 Channel [27:24])
0xC5	DJB_TX_23_20	De-Jitter Buffer TX Control (DS 1 Channel [23:20])
0xC4	DJB_TX_19_16	De-Jitter Buffer TX Control (DS 1 Channel [19:16])
0xC3	DJB_TX_15_12	De-Jitter Buffer TX Control (DS 1 Channel [15:12])
0xC2	DJB_TX_11_08	De-Jitter Buffer TX Control (DS 1 Channel [11:08])
0xC1	DJB_TX_07_04	De-Jitter Buffer TX Control (DS 1 Channel [07:04])
0xC0	DJB_TX_03_00	De-Jitter Buffer TX Control (DS 1 Channel [03:00])
0xBF - 0xB7		Undefined (reserved)
0xB6	IDLE_SET_THR	Idle Code Set Threshold Timing for RX Direction
0xB5	PAT5_SET_THR	Pattern=10100 Set Threshold Timing for RX Direction
0xB4	NILD_SET_THR	NI LoopDown Set threshold Timing for RX Direction
0xB3	NILU_SET_THR	NI LoopUp Set threshold Timing for RX Direction
0xB2	CSULD_SET_THR	CSU LoopDown Set threshold Timing for RX Direction
0xB1	CSULU_SET_THR	CSU LoopUp Set threshold Timing for RX Direction
0xB0	YEL_SET_THR	Yellow Detection Set Threshold Timing for RX Direction
0xAF - 0xA9		Undefined (reserved)
0xA8	RED_CLR_SUB0	RED Clear Threshold (in RED)
0xA7	AIS_CLR_THR	AIS Clear Threshold Pattern Match
0xA6	AIS_SET_THR	AIS Set Threshold Pattern Match
0xA5	RED_CLR_SUB	RED Clear Threshold Pattern Match (not in RED)
0xA4	RED_SET_ADD	RED Set Threshold Pattern Match
0xA3		Undefined (reserved)
0xA2	MIS_WIN16	Mismatch Pattern Window Threshold
0xA1 - 0xA0	MIS_3MS_[1:0]	Mismatch Pattern Threshold
0x9F - 0x9C		Undefined (reserved)
0x9B - 0x80	INTSOURCE[27:0]	Interrupt Source
0x7F - 0x7C		Undefined (reserved)
0x7B - 0x60	CONTROL_BE[27:0]	Back End System Interface Control
0x5F - 0x5C		Undefined (reserved)
0x5B - 0x40	CONTROL_FE[27:0]	Front End Line Interface Control

Table 3.1 List of Directly Mapped Registers (continued)

0x3F - 0x3C	LBENAISON[3:0]	Local Loopback Enable & AIS ON Control
0x3B - 0x38	LBENAN[3:0]	Network Loopback Enable Control
0x37 - 0x34	LBENAL[3:0]	Local Loopback Enable Control
0x33 - 0x31		Undefined (reserved)
0x30	LBENAMISC	Loopback Enable Miscellaneous Control
0x2F	RXFDL_ESFYEL[27:24]	RX FDL ESF Yellow Status for DS 1 channels [27:24]
0x2E	RXFDL_ESFYEL[23:16]	RX FDL ESF Yellow Status for DS 1 channels [23:16]
0x2D	RXFDL_ESFYEL[15:08]	RX FDL ESF Yellow Status for DS 1 channels [15:08]
0x2C	RXFDL_ESFYEL[07:00]	RX FDL ESF Yellow Status for DS 1 channels [07:00]
0x2B	RX_JT1_SFYEL[27:24]	RX JT1 SF Yellow Status for DS 1 channels [27:24]
0x2A	RX_JT1_SFYEL[23:16]	RX JT1 SF Yellow Status for DS 1 channels [23:16]
0x29	RX_JT1_SFYEL[15:08]	RX JT1 SF Yellow Status for DS 1 channels [15:08]
0x28	RX_JT1_SFYEL[07:00]	RX JT1 SF Yellow Status for DS 1 channels [07:00]
0x27 - 0x21		Undefined (reserved)
0x20	ERRMISC	Error Miscellaneous Control
0x1F	SIG_FIFO_0	RX Signaling FIFO Data Low Byte
0x1E	SIG_FIFO_1	RX Signaling FIFO Data High Byte
0x1D	SIG_FIFO_STAT	RX Signaling FIFO Status/ Control
0x1C	SIG_CNTTHR_OVRIDE	RX Signaling FSM Threshold Override
0x1B - 0x19		Undefined (reserved)
0x18	TXMISC	TX Miscellaneous Control
0x17 - 0x11		Undefined (reserved)
0x10	RXMISC	RX Miscellaneous Control
0x0F - 0x0A		Undefined (reserved)
0x09	INTMASK	Interrupt Mask Control
0x08	INTSUMMARY	Interrupt Summary Status
0x07	CPUIDT3	CPU Indirect 32-bit Data bits [31:24]
0x06	CPUIDT2	CPU Indirect 32-bit Data bits [23:16]
0x05	CPUIDT1	CPU Indirect 32-bit Data bits [15:08]
0x04	CPUIDT0	CPU Indirect 32-bit Data bits [07:00]
0x03	CPUIAD1	CPU Indirect 16-bit Address bits [15:08] (see Table 3.3)
0x02	CPUIAD0	CPU Indirect 16-bit Address bits [07:00] (see Table 3.3)
0x01	CPUICMD	CPU Indirect Command/Status Control
0x00	CPUIDRST	CPU Indirect Reset/Interrupt Control & Chip ID

All the RAMs in the system are listed below, with their sizes. The indirect address range of longword locations within each RAM is also listed.

Table 3.2 List of RAMs

Name	Per-Channel Size Longwords/ bytes	Total Size Longwords/ bytes	Indirect Address Location, Range
RX RAM, history	32 LW / 128 B	1K LW / 4K B	0x1000 - 0x1FFF
RX RAM, state	8 LW / 32 B	256 LW / 1K B	0x2000 - 0x23FF
TX RAM	8 LW / 32 B	256 LW / 1K B	0x4000 - 0x43FF
Performance-monitor flags RAM	2 LW / 8 B	64 LW / 256 B	0x5000 - 0x50FF
Performance-monitor counts RAM	8 LW / 32 B	256 LW / 1K B	0x5400 - 0x57FF
FDL RAM, RX	32 LW / 128 B	1K LW / 4K B	0x6000 - 0x6FFF
FDL RAM, TX	32 LW / 128 B	1K LW / 4K B	0x8000 - 0x8FFF
Signaling RAM, RX, TX	32 LW / 128 B	1K LW / 4K B	0xA000 - 0xAFFF
Slip buffer RAM, RX	16 LW / 64 B	512 LW / 2K B	0xB000 - 0xB7FF
Slip buffer RAM, TX	16 LW / 64 B	512 LW / 2K B	0xC000 - 0xC7FF
Loopback buffer RAM	16 LW / 64 B	512 LW / 2K B	0xD000 - 0xD7FF

Table 3.3 List Of Indirectly Mapped Registers

Address (Hex)	Register Name	Function
0x1000 - 0x1D80	RXH_RAM_CH[0:27]	RX History RAM
0x2000 - 0x2360	RXS_RAM_CH[0:27]	RX State RAM
0x4000 - 0x4360	TXS_RAM_CH[0:27]	TX State RAM
0x5000 - 0x50D8	FLAG_RAM_CH[0:27]	Performance Monitor Flags RAM
0x5400 - 0x5760	COUNT_RAM_CH[0:27]	Performance Monitor Count RAM
0x6000 - 0x6D80	FDL_RX_RAM_CH[0:27]	Facility Data Link RX RAM
0x8000 - 0x8D80	FDL_TX_RAM_CH[0:27]	Facility Data Link TX RAM
0xA000 - 0xA6C0	SIG_RX_RAM_CH[0:27]	RX Signaling RAM
0xA800 - 0xAEC0	SIG_TX_RAM_CH[0:27]	TX Signaling RAM
0xB000 - 0xB6C0	SLIP_RX_RAM_CH[0:27]	RX Slip Buffer RAM
0xC000 - 0xC6C0	SLIP_TX_RAM_CH[0:27]	TX Slip Buffer RAM
0xD000 - 0xD6C0	LBACK_RAM_CH[0:27]	Network Loopback RAM (per-DS0)

3.1 CPU Interface Section Registers

Register Name: CPUIDRST - CPU Indirect Reset/Interrupt Control & Chip ID
Register Address: 0x00

Bit	Type	Function	Reset
7	R/W	soft_rst	0
6	R/W	cpu_ind_rst	0
5	R/W	int_level	0
4	RO	int	0
3:0	RO	id[3:0]	0x2

soft_rst:
The soft_rst bit implements a soft reset. When this bit is set to logical 1, the state machines and RAMs are initialized. This bit is self-clearing to logical 0 after the reset operation is finished. The RESETN input pin completely resets all.

cpu_ind_rst:
The cpu_ind_rst bit implements a CPU indirect reset. When this bit is set to logical 1, the CPU indirect read/write protocol state machines are cleared. This bit is self-clearing to logical 0 after the CPU indirect reset is finished.

int_level:
This bit allows software to program the polarity of the interrupt output pin, CPU_INT. If this bit is set to logical 0, then CPU_INT signal is active low. If this bit is set to logical 1, then CPU_INT signal is active high.

int:
This bit is a copy of CPU_INT pin (useful for polling).

id[3:0]:
These bits are read only. They are used to identify the version of VSC9670 device.

Register Name: CPUICMD - CPU Indirect Command/Status Control
Register Address: 0x01

Register Name: INTICMD - Interrupt Indirect Command/Status Control
Register Address:0xD1

(CPUICMD and INTICMD registers are identical and share bit functionality.)

Bit	Type	Function	Reset
7:4	R/W	bmask[3:0]	0x0
3	R/W	undef_fdl	0
2	R/W	err	0
1	R/W	wcmd	0
0	R/W	rcmd	0

The CPUICMD register is used to launch commands or monitor the status of indirect write or read sequences while the code flow is outside an ISR (Interrupt Service Routine). The INTICMD register is used to start and check indirect write or read sequences from inside an interrupt flow. The CPUICMD register uses the CPUIAD and CPUIDT registers for address and data; while the INTICMD uses the INTIAD and INTIDT registers for address and data. Only one read or write sequence may be active at a time and the sequences are atomic. So, for example, once a write to the CPUICMD register occurs, the read or write sequence initiated by that register write will run to completion. If an INTICMD register write is issued while a CPUICMD sequence is active, the err bit in INTICMD register is set and the INTICMD initiated sequence is aborted. Prior to launching a new request to the CPUICMD or INTICMD register, the software should poll either the CPUICMD or the INTICMD registers to determine if the previous read or write operation has completed.

bmask[3:0]:
These bmask[3:0] bits implement a byte mask operation. They indicate to the VSC9670 which bytes of the 32 bit word on the CPUIDT/INTIDT bus are valid data. For four-bit mode, each bit set to logical 1 indicates a valid byte; 0 indicates an invalid byte.

bmask[0] affects CPUIDT0[7:0]/INTIDT0[7:0] of a 32-bit location
bmask[1] affects CPUIDT1[15:8]/INTIDT1[15:8] of a 32-bit location
bmask[2] affects CPUIDT2[23:16]/INTIDT2[23:16] of a 32-bit location
bmask[3] affects CPUIDT3[31:24]/INTIDT3[31:24] of a 32-bit location

undef_fdl:
The undef_fdl bit implements different functions during CPU access. During a CPU read/write operation to any register (except RX/TX FDL FIFO - 4 bytes of FDL data registers), the undef_fdl bit reads 0 if the CPU accessed a legitimate address. Otherwise this bit reads 1 if the CPU attempted to access an undefined address.

If the CPU writes this bit to logical 0, then the CPU indirect access targets RAM locations. If CPU writes this bit to logical 1, then the CPU indirect access targets RX/TX FDL FIFO (only some addresses are legal, see the section for the FDL).

err:
The err bit is used to indicate an error condition. If the CPU indirect read/write finished error-free, this bit reads 0. If any error occurred this bit reads 1. The CPU must write to 0 to clear.

wcmd:
The wcmd bit enables the CPU write command. This bit is cleared to 0 when the write command is completed. Upon read, if both wcmd and rcmd bits are 0, then wcmd is set to 1, and the VSC9670 initiates a write to the register selected by the 16-bit indirect address registers CPUIAD[15:0]/INTIAD[15:0].

rcmd:
The rcmd bit enables the CPU read command. This bit is cleared to 0 when the read command is completed. Upon read, if both wcmd and rcmd bits are 0, then rcmd is set to 1, and the VSC9670 initiates a read to the register selected by the 16-bit indirect address registers CPUIAD[15:0]/INTIAD[15:0].

Register Name: CPUIAD0, CPUIAD1 - CPU Indirect Address
Register Address: 0x02, 0x03

Bit	Type	Function	Reset
7:0	R/W	CPUIAD0 [7:0]	0x00
7:0	R/W	CPUIAD1 [15:8]	0x00

CPUIAD0 [7:0]:
These bits CPUIAD0 [7:0] contain the lower 8 bits of the 16-bit indirect address.

CPUIAD1 [15:8]:
These bits CPUIAD1 [15:8] contain the higher 8 bits of the 16-bit indirect address.

Register Name: CPUIDT0, CPUIDT1, CPUIDT2, CPUIDT3 - CPU Indirect Data
Register Address: 0x04, 0x05, 0x06, 0x07

Bit	Type	Function	Reset
7:0	R/W	CPUIDT0 [7:0]	0x00
7:0	R/W	CPUIDT1 [15:8]	0x00
7:0	R/W	CPUIDT2 [23:16]	0x00
7:0	R/W	CPUIDT3 [31:24]	0x00

These registers CPUIDT3, CPUIDT2, CPUIDT1, CPUIDT0 are filled by the CPU with the 32-bit indirect write data before initiating an indirect write. They are also filled by the VSC9670 with the 32-bit indirect read data after the CPU initiates an indirect read.

CPUIDT0 [7:0]:
These bits CPUIDT0 [7:0] contain the first 8 bits of the 32-bit indirect data.

CPUIDT1 [15:8]:
These bits CPUIDT1 [15:8] contain the second 8 bits of the 32-bit indirect data.

CPUIDT2 [23:16]:
These bits CPUIDT2 [23:16] contain the third 8 bits of the 32-bit indirect data.

CPUIDT3 [31:24]:
These bits CPUIDT3 [31:24] contain the fourth 8 bits of the 32-bit indirect data.

Register Name: INTIAD0, INTIAD1 - Interrupt Indirect Address
Register Address: 0xD2, 0xD3

Bit	Type	Function	Reset
7:0	R/W	INTIAD0 [7:0]	0x00
7:0	R/W	INTIAD1 [15:8]	0x00

INTIAD0 [7:0]:
When processing interrupts, these bits INTIAD0 [7:0] contain the lower 8 bits of the 16-bit indirect address.

INTIAD1 [15:8]:
When processing interrupts, these bits INTIAD1 [15:8] contain the higher 8 bits of the 16-bit indirect address.

Register Name: INTIDT0, INTIDT1, INTIDT2, INTIDT3
- Interrupt Indirect Data
Register Address:0xD4, 0xD5, 0xD6, 0xD7

Bit	Type	Function	Reset
7:0	R/W	INTIDT0 [7:0]	0x00
7:0	R/W	INTIDT1 [15:8]	0x00
7:0	R/W	INTIDT2 [23:16]	0x00
7:0	R/W	INTIDT3 [31:24]	0x00

When processing interrupts, these registers INTIDT3, INTIDT2, INTIDT1, INTIDT0 are filled by the CPU with the 32-bit indirect write data before initiating an indirect write. They are also filled by the VSC9670 with the 32-bit indirect read data after the CPU initiates an indirect read.

INTIDT0 [7:0]:
These bits INTIDT0 [7:0] contain the first 8 bits of the 32-bit indirect data.

INTIDT1 [15:8]:
These bits INTIDT1 [15:8] contain the second 8 bits of the 32-bit indirect data.

INTIDT2 [23:16]:
These bits INTIDT2 [23:16] contain the third 8 bits of the 32-bit indirect data.

INTIDT3 [31:24]:
These bits INTIDT3 [31:24] contain the fourth 8 bits of the 32-bit indirect data.

Register Name: INTSUMMARY - Interrupt Summary
Register Address:0x08

Bit	Type	Function	Reset	Source
7	RO, W1C	one_sec	0	One Second RollOver
6	RO	any_rx_fdl	0	FIFO High WaterMark or on every sub-packet, ESF Yellow Interrupt
5	RO	any_tx_fdl	0	FIFO Low Water Mark or on every sub-packet
4	RO	any_rx_sig	0	per-DS0 Interrupt
3	RO	any_tx_err	0	tx_fe_ufl
2	RO	any_rx_err	0	rx_fe_ofl
1	RO	any_tx_alm	0	tx_slip
0	RO	any_rx_alm	0	red, los, rx_slip, AIS, OOF Pattern Recognition: YELLOW, CSU_LOOP_UP, CSU_LOOP_DN, NI_LOOP_UP, NI_LOOP_DN, IDLE

INTSUMMARY holds the status of each interrupt-causing event in an OR of corresponding bits in INTSOURCE[0:27] registers. When an interrupt causing event occurs, the corresponding bit is set to 1, and stays 1 until the underlying condition is addressed. Bit 7 (one_sec) is write-1-to-clear.

one_sec:
The one_sec bit is used for an internal 1-second counter. If the internal 1-second counter rolls over, one_sec is set to logical 1. This bit is write-1-to-clear.

any_rx_fdl:
This bit indicates new RX FDL data. If one or more DS1s have new RX FDL data, this bit is set to ‘1’.

This bit also indicates RX FDL ESF Yellow Interrupt.

any_tx_fdl:
The any_tx_fdl bit indicates that new TX FDL data should be accepted. If one or more DS1s can accept new TX FDL data, this bit is set to logical 1.

any_rx_sig:
The any_rx_sig bit indicates new RX signaling bits. If one or more DS1s have new RX signaling bits, this bit is set to logical 1.

any_tx_err:
Indicates a TX error condition. If one or more DS1s have a TX error condition, this bit is set to ‘1’.

any_rx_err:
Indicates RX error condition. If one or more DS1s have an RX error condition, this bit is set to ‘1’.

any_tx_alm:
This bit indicates a TX alarm condition. If one or more DS1s have a TX alarm condition, then this bit is set to logical 1.

any_rx_alm:
This bit indicates an RX alarm condition. If one or more DS1s have an RX alarm condition, this bit is set to logical 1.

Register Name: INTMASK - Interrupt Mask
Register Address: 0x09

Bit	Type	Function	Reset
7	R/W	mask_one_sec	0
6	R/W	mask_rx_fdl	0
5	R/W	mask_tx_fdl	0
4	R/W	mask_rx_sig	0
3	R/W	mask_tx_err	0
2	R/W	mask_rx_err	0
1	R/W	mask_tx_alm	0
0	R/W	mask_rx_alm	0

This is the interrupt mask register corresponding to the interrupt summary status register (INTSUMMARY). Each bit in the mask register independently masks the corresponding bit in the summary status register (i.e. the bit with the same number and the same sub-name). When a mask bit is set to 0, the interrupt due to the corresponding event is masked. When a mask bit is set to 1, the interrupt due to the corresponding event is unmasked. Note that the mask bit does not affect the setting of the summary status bit, it only affects whether or not the summary status bit will cause the CPU_INT pin to be asserted.

Register Name: INTSOURCE0 .. INTSOURCE27 - Interrupt Source (28 DS1's)
Register Address: 0x80 .. 0x9B (28 locations)

Bit	Type	Function	Reset	Source
7	RO,W1C	rx_fdl_esf	0	RX FDL ESF Yellow Interrupt
6	RO, W1C	rx_fdl	0	FIFO High WaterMark or on every sub-packet
5	RO, W1C	tx_fdl	0	FIFO Low Water Mark or on every sub-packet
4	RO, W1C	rx_sig	0	Per-DS0 Interrupt
3	RO, W1C	tx_err	0	tx_fe_ufl
2	RO, W1C	rx_err	0	rx_fe_ofl
1	RO, W1C	tx_alm	0	tx_slip
0	RO, W1C	rx_alm	0	red, los, rx_slip, AIS, OOF Pattern Recognition: YELLOW, CSU_LOOP_UP, CSU_LOOP_DN, NI_LOOP_UP, NI_LOOP_DN, IDLE

The above registers allow software to determine the source which produced the interrupt on the CPU_INT output pin. These interrupt source register bits [6:0] are read only and write-1-to-clear. Each bit in the interrupt source register corresponds to the summary status register bit (i.e. the bit with the same number and the same sub-name). Once a bit is set, it can only be cleared by performing the appropriate actions that address the underlying cause. Specifically, for bit [4:0], the per-DS1 interrupt bits in the underlying RAM's (RX, TX, ERR, or SIG) must be cleared by writing 1.

rx_fdl_esf:
The rx_fdl_esf bit indicates a new RX FDL ESF Yellow Interrupt. If this DS1 has a new RX FDL ESF Yellow Interrupt, then this bit is set to logical 1.

rx_fdl:
The rx_fdl bit indicates new RX FDL data. If this DS1 has new RX FDL data, then this bit is set to logical 1. Once set, it can only be cleared by reading RX FDL FIFO to bring it below its high water mark.

tx_fdl:
The tx_fdl bit indicates that new TX FDL data can be accepted. If this DS1 can accept new TX FDL data, then this bit is set to logical 1. Once set, it can only be cleared by writing TX FDL FIFO to bring it above its low water mark.

rx_sig:
The rx_sig bit indicates new RX signaling bits. If this DS1 has new RX signaling bits, then this bit is set to logical 1.

tx_err:
The tx_err bit indicates a TX error condition. If this DS1 has a TX error condition, then this bit is set to logical 1.

rx_err:
The rx_err bit indicates an RX error condition. If this DS1 has an RX error condition, then this bit is set to logical 1.

tx_alm:
The tx_alm bit indicates a TX alarm condition. If this DS1 has a TX alarm condition, then this bit is set to logical 1.

rx_alm:
The rx_alm bit indicates an RX alarm condition. If this DS1 has an RX alarm condition, then this bit is set to logical 1.

Register Name: CPUDIAG - CPU Diagnostic Control
Register Address: 0xFF

Bit	Type	Function	Reset
7:0	P	Reserved	0x00

This register contains reserved bits for internal test CPU and other diagnostic use. It must not be written to other than 0x00.

3.2 RX State Machine, Registers and RAM

3.2.1 Per Chip Registers

Register Name: RXMISC - RX Miscellaneous Control
Register Address: 0x10

Bit	Type	Function	Reset
7		Unused	X
6	R/W	rx_fdl_ena	0
5	R/W	rx_sigfreeze_dis	0
4	R/W	rx_sigbus_ena	0
3:2	R/W	ssp[1:0]	0x0
1:0	RO	sysclkmode[1:0]	0x0

rx_fdl_ena:
The rx_fdl_ena bit is used to enable the RX FDL state machine. When this bit is set to logical 1, the RX FDL state machine is enabled. When this bit is set to logical 0, the RX_FDL state machine is disabled. The rx_ena bit in FDL_RX_RAM must also be set to enable a particular DS1 (Refer to Section 3.5 FDL Machine, RAM Bits).

rx_sigfreeze_dis:
The rx_sigfreeze_dis bit is used to disable RX Per-DS1 Signaling Freeze. When this bit is set to logical 1, the RX Per-DS1 Signaling Freeze is disabled. When this bit is set to logical 0, the RX Per-DS1 Signaling Freeze is enabled.

rx_sigbus_ena:
The rx_sigbus_ena bit is used to enable the RX signaling bus. When this bit is set to logical 1, the pins SYSRXSIG[27:0] are used as the System Receive Signaling Output Bus in Slip Buffer mode.

ssp[1:0]:
These bits ssp[1:0] allow software to program the system backplane common clock speed. This is valid only in Slip Buffer mode.

ssp[1:0]	System Backplane Clock/sync
00	1.544 MHz
01	2.048 MHz
10	4.096 MHz
11	8.192 MHz

sysclkmode[1:0]:
These bits are read only. The value of the sysclkmode[1:0] bits indicates the status of the system clock mode selection input pins SYSCLKMODE[1:0].

sysclkmode[1:0]	Operation Mode
00	Data Termination
01	Low Latency
10	Pin Efficient
11	Slip Buffer

Register Name: MIS_3MS_0 - Mismatch Pattern Threshold 0
Register Address: 0xA0

Bit	Type	Function	Reset
7:5		Unused	X
4:0	R/W	mis_3ms_0[4:0]	0x00

mis_3ms_0[4:0]:
These bits mis_3ms_0[4:0] are used to set the mismatch threshold for repetitive pattern detection within 3 milliseconds. They are enabled when pat_mis_thr_sel = 0 in RXS_RAM_BASE register. The recommended value in this register is 5.

Register Name: MIS_3MS_1 - Mismatch Pattern Threshold 1
Register Address: 0xA1

Bit	Type	Function	Reset
7:5		Unused	X
4:0	R/W	mis_3ms_1[4:0]	0x00

mis_3ms_1[4:0]:
These bits mis_3ms_1[4:0] are used to set the mismatch threshold for repetitive pattern detection within 3 milliseconds. They are enabled when pat_mis_thr_sel = 1 in RXS_RAM_BASE register. The recommended value in this register is 5.

Register Name: MIS_WIN16 - Mismatch Pattern Window Threshold
Register Address: 0xA2

Bit	Type	Function	Reset
7:4	R/W	mis_win16_1[3:0]	0x0
3:0	R/W	mis_win16_0[3:0]	0x0

mis_win16_1[3:0]:
These bits mis_win16_1[3:0] are used to set the mismatch threshold for repetitive pattern detection over a sliding window of 16 3-ms periods. They are enabled when pat_mis_thr_sel = 1 in RXS_RAM_BASE register. The recommended value in this register is 8.

mis_win16_0[3:0]:
These bits mis_win16_0[3:0] are used to set the mismatch threshold for repetitive pattern detection over a sliding window of 16 3-ms periods. They are enabled when pat_mis_thr_sel = 0 in RXS_RAM_BASE register. The recommended value in this register is 8.

3.2.2 RX RAM Registers

The RX History RAM is 32 bits wide, and 32 longwords (128 bytes) are used per DS1 channel, leading to a total size of 896 (32 x 28 channels) longwords. If the RX History RAM is located at absolute address RXH_RAM_BASE in the CPU's memory space, a channel n (from 0 to 27) will have a base address of [RXH_RAM_BASE + (n x 128)].

The RX State RAM is 32 bits wide, and 8 longwords (32 bytes) are used per DS1 channel, leading to a total size of 224 longwords (8 x 28 channels). If the RX State RAM is located at absolute address RXS_RAM_BASE in the CPU's memory space, a channel n (from 0 to 27) will have a base address of [RXS_RAM_BASE + (n x 32)].

As described earlier,
RXH_RAM_BASE = 0x1000 (this is a base address for channel 0).
RXS_RAM_BASE = 0x2000 (this is a base address for channel 0).

The following table is valid for writes to location 7C of each channel during CPU indirect write access to this RXH_RAM. (x = don't care).

bmask[3:0] (in CPUICMD bits [7:4])	RXH_RAM bits are written
xxxx (Byte enables are ignored)	Bits [31:00]

The following table is valid for writes to locations 0C, 10, 14 of each channel during CPU indirect write access to this RXS_RAM. (x = don't care).

bmask[3:0] (in CPUICMD bits [7:4])	RXS_RAM bits are written
xxxx (Byte enables are ignored)	Bits [31:00]

Register Name: **RXH_RAM_CH0 .. RXH_RAM_CH27 (28 DS1's) - RX History RAM**
Register Address: **0x1000, 0x1080, 0x1100, 0x1180, 0x1200, 0x1280, 0x1300, 0x1380, 0x1400, 0x1480, 0x1500, 0x1580, 0x1600, 0x1680, 0x1700, 0x1780, 0x1800, 0x1880, 0x1900, 0x1980, 0x1A00, 0x1A80, 0x1B00, 0x1B80, 0x1C00, 0x1C80, 0x1D00, 0x1D80 (28 Locations)**

<table>
<tr><th colspan="2"></th><th colspan="32">Bit Function</th></tr>
<tr><th colspan="2"></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td rowspan="32">Address Location for 32 Longwords (Hex)</td><td>00</td><td colspan="8">Row 3, Byte 0</td><td colspan="8">Row 2, Byte 0</td><td colspan="8">Row 1, Byte 0</td><td colspan="8">Row 0, Byte 0</td></tr>
<tr><td>04</td><td colspan="8">Row 3, Byte 1</td><td colspan="8">Row 2, Byte 1</td><td colspan="8">Row 1, Byte 1</td><td colspan="8">Row 0, Byte 1</td></tr>
<tr><td>08</td><td colspan="8">Row 3, Byte 2</td><td colspan="8">Row 2, Byte 2</td><td colspan="8">Row 1, Byte 2</td><td colspan="8">Row 0, Byte 2</td></tr>
<tr><td>0C</td><td colspan="8">Row 3, Byte 3</td><td colspan="8">Row 2, Byte 3</td><td colspan="8">Row 1, Byte 3</td><td colspan="8">Row 0, Byte 3</td></tr>
<tr><td>10</td><td colspan="8">Row 3, Byte 4</td><td colspan="8">Row 2, Byte 4</td><td colspan="8">Row 1, Byte 4</td><td colspan="8">Row 0, Byte 4</td></tr>
<tr><td>14</td><td colspan="8">Row 3, Byte 5</td><td colspan="8">Row 2, Byte 5</td><td colspan="8">Row 1, Byte 5</td><td colspan="8">Row 0, Byte 5</td></tr>
<tr><td>18</td><td colspan="8">Row 3, Byte 6</td><td colspan="8">Row 2, Byte 6</td><td colspan="8">Row 1, Byte 6</td><td colspan="8">Row 0, Byte 6</td></tr>
<tr><td>1C</td><td colspan="8">Row 3, Byte 7</td><td colspan="8">Row 2, Byte 7</td><td colspan="8">Row 1, Byte 7</td><td colspan="8">Row 0, Byte 7</td></tr>
<tr><td>20</td><td colspan="8">Row 3, Byte 8</td><td colspan="8">Row 2, Byte 8</td><td colspan="8">Row 1, Byte 8</td><td colspan="8">Row 0, Byte 8</td></tr>
<tr><td>24</td><td colspan="8">Row 3, Byte 9</td><td colspan="8">Row 2, Byte 9</td><td colspan="8">Row 1, Byte 9</td><td colspan="8">Row 0, Byte 9</td></tr>
<tr><td>28</td><td colspan="8">Row 3, Byte 10</td><td colspan="8">Row 2, Byte 10</td><td colspan="8">Row 1, Byte 10</td><td colspan="8">Row 0, Byte 10</td></tr>
<tr><td>2C</td><td colspan="8">Row 3, Byte 11</td><td colspan="8">Row 2, Byte 11</td><td colspan="8">Row 1, Byte 11</td><td colspan="8">Row 0, Byte 11</td></tr>
<tr><td>30</td><td colspan="8">Row 3, Byte 12</td><td colspan="8">Row 2, Byte 12</td><td colspan="8">Row 1, Byte 12</td><td colspan="8">Row 0, Byte 12</td></tr>
<tr><td>34</td><td colspan="8">Row 3, Byte 13</td><td colspan="8">Row 2, Byte 13</td><td colspan="8">Row 1, Byte 13</td><td colspan="8">Row 0, Byte 13</td></tr>
<tr><td>38</td><td colspan="8">Row 3, Byte 14</td><td colspan="8">Row 2, Byte 14</td><td colspan="8">Row 1, Byte 14</td><td colspan="8">Row 0, Byte 14</td></tr>
<tr><td>3C</td><td colspan="8">Row 3, Byte 15</td><td colspan="8">Row 2, Byte 15</td><td colspan="8">Row 1, Byte 15</td><td colspan="8">Row 0, Byte 15</td></tr>
<tr><td>40</td><td colspan="8">Row 3, Byte 16</td><td colspan="8">Row 2, Byte 16</td><td colspan="8">Row 1, Byte 16</td><td colspan="8">Row 0, Byte 16</td></tr>
<tr><td>44</td><td colspan="8">Row 3, Byte 17</td><td colspan="8">Row 2, Byte 17</td><td colspan="8">Row 1, Byte 17</td><td colspan="8">Row 0, Byte 17</td></tr>
<tr><td>48</td><td colspan="8">Row 3, Byte 18</td><td colspan="8">Row 2, Byte 18</td><td colspan="8">Row 1, Byte 18</td><td colspan="8">Row 0, Byte 18</td></tr>
<tr><td>4C</td><td colspan="8">Row 3, Byte 19</td><td colspan="8">Row 2, Byte 19</td><td colspan="8">Row 1, Byte 19</td><td colspan="8">Row 0, Byte 19</td></tr>
<tr><td>50</td><td colspan="8">Row 3, Byte 20</td><td colspan="8">Row 2, Byte 20</td><td colspan="8">Row 1, Byte 20</td><td colspan="8">Row 0, Byte 20</td></tr>
<tr><td>54</td><td colspan="8">Row 3, Byte 21</td><td colspan="8">Row 2, Byte 21</td><td colspan="8">Row 1, Byte 21</td><td colspan="8">Row 0, Byte 21</td></tr>
<tr><td>58</td><td colspan="8">Row 3, Byte 22</td><td colspan="8">Row 2, Byte 22</td><td colspan="8">Row 1, Byte 22</td><td colspan="8">Row 0, Byte 22</td></tr>
<tr><td>5C</td><td colspan="8">Row 3, Byte 23</td><td colspan="8">Row 2, Byte 23</td><td colspan="8">Row 1, Byte 23</td><td colspan="8">Row 0, Byte 23</td></tr>
<tr><td>60</td><td colspan="8">Row 3, Byte 24</td><td colspan="8">Row 2, Byte 24</td><td colspan="8">Row 1, Byte 24</td><td colspan="8">Row 0, Byte 24</td></tr>
<tr><td>64</td><td colspan="32">Unused</td></tr>
<tr><td>68</td><td colspan="32">Unused</td></tr>
<tr><td>6C</td><td colspan="32">Unused</td></tr>
<tr><td>70</td><td colspan="32">Unused</td></tr>
<tr><td>74</td><td colspan="32">Unused</td></tr>
<tr><td>78</td><td colspan="2">sig_mf [1:0]</td><td colspan="30">Unused</td></tr>
<tr><td>7C</td><td>dont_frame</td><td>use_crc</td><td colspan="2">sf_esf [1:0]</td><td colspan="2">frm_thresh [1:0]</td><td colspan="2">oof_thresh [1:0]</td><td colspan="24">pn_ena [23:0]</td></tr>
</table>

Row[0:3], Byte[0:24]: (RO)
In the RX history RAM, the first 25 longwords per DS1 channel are used as three or four (ESF or SF respectively) one-DS1-frame sized buffers. When acquiring frame sync, 193 bits at a time are stored in the column marked Row0, Byte[0:24]; then another 193 bits (separated by 193*4 bits in the case of ESF) are stored in the column marked Row1, Byte[0:24]. There is a similar process for the Row 2 column. Three rows' worth of storage is adequate for ESF frame sync acquisition, four for SF.

pn_ena[23:0]: (R/W)
Each of these pn_ena[23:0] bits selects one of the 24 DS0s in the PN pattern checking function. If a bit is set to 1, the corresponding DS0 is fed to the PN checker. If a bit is set to 0, the corresponding DS0 is ignored in PN pattern checking. If the pn_fullt1 bit is set to 1 in the RXH_RAM, pn_ena is overridden and the full T1 is selected. The pn_m64_56 bit in the RXH_RAM is used to select either 8-bit or 7-bit to apply to each of the selected DS0's.

oof_thresh[1:0]: (R/W)
The oof_thresh[1:0] bits are used to select the threshold for the number of error F-bits needed to set the OOF condition from a normal framed state.

oof_thresh[1:0]	**Select Function**
00	Two F-bit errors in four consecutive F bits
01	Two F-bit errors in five consecutive F bits
10	Two F-bit errors in six consecutive F bits
11	Three F-bit errors in five consecutive F bits

frm_thresh[1:0]: (R/W)
The frm_thresh[1:0] bits are used to select the threshold for the number of correct F-bits needed to clear the OOF condition from the OOF state.

frm_thresh[1:0]	**F-bits**	**Equivalent number of frames**		
		ESF/FPS	**SF/Ft**	**SF/Ft+Fs**
00	6	24	12	6
01	12	48	24	12
10	18	72	36	18
11	24	96	48	24

sf_esf[1:0]: (R/W)
The sf_esf[1:0] bits are used to select the RX frame format. In Superframe (SF) format, both Ft and Fs bits are used for framing. Once framed, both Ft and Fs errors are reported.

sf_esf[1:0]	**Frame Format**
00	Extended Superframe (ESF) format
01	Unused
10	Unused
11	Superframe (SF) format

use_crc: (R/W)
This bit is used to enable checking CRC-6 in ESF format. If use_crc is set to 0, the CRC-6 checking is omitted during a frame-acquisition condition. If use_crc is set to 1 while framing, after the F-bit sequence is acquired, RX checks for CRC-6 before clearing OOF.

dont_frame: (R/W)
This bit is used to control the framing operation. If dont_frame is set to 0, the RX will try to re-frame after it is in OOF mode. If dont_frame is set to 1, the RX will not try to re-frame after it is in OOF mode, and passes through data. Briefly, setting dont_frame during a framed operation causes re-acquisition of the frame.

sig_mf[1:0]: (Reserved)
These reserved bits are used for RX State RAM implementation, so these locations should not be programmed. The sig_mf[1:0] field contains reserved bits and is used by the hardware during DS1 signaling freeze processing.

Register Name: **RXS_RAM_CH0 .. RXS_RAM_CH27 (28 DS1s) - RX State RAM**
Register Address: **0x2000, 0x2020, 0x2040, 0x2060, 0x2080, 0x20A0, 0x20C0, 0x20E0, 0x2100, 0x2120, 0x2140, 0x2160, 0x2180, 0x21A0, 0x21C0, 0x21E0, 0x2200, 0x2220, 0x2240, 0x2260, 0x2280, 0x22A0, 0x22C0, 0x22E0, 0x2300, 0x2320, 0x2340, 0x2360 (28 Locations)**

Address Location for 8 Longwords (Hex)	Bit Function																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
00	Reserved																															
04	Reserved																															
08	Reserved																															
0C	Reserved										pn_length[4:0]					pn_tap[4:0]					Unused	exz_ena	pn_fullt1	pn_invert	pn_m64_56	pn_qrss	pn_fixed	pat_kill_los	pat_kill_loc	pat_mis_thr_sel	pat_unframed	soft_reset
10	P-N_Shift[31:0]																															
14	offset_ss [1:0]		auto_ais_los	auto_ais_red	sig_inj1	sync_sf_esf	inj_ais	inj_idle	Reserved																							
18	Unused																															
1C	Unused																															

Reserved:
These reserved bits are used for RX State RAM implementation, so these locations should not be programmed.

soft_reset: (R/W)
The soft_reset bit is set to 1 to initialize the RX framer. The bit auto-clears itself when the operation is complete.

pat_unframed: (R/W)
The pat_unframed bit is used to control RX patterns detection. If this bit is set to 0, the RX detects repetitive patterns 001, 00111, 00011, 00001 and 10100 with the framing column present. If this bit is set to 1, the RX detects the above patterns over-writing framing.

pat_mis_thr_sel: (R/W)
The pat_mis_thr_sel bit is used to select the RX pattern mismatch thresholds for this DS1 channel. If this bit is set to 0, the RX selects the mis_3ms_0[4:0] and mis_win16_0[3:0]. If this bit is set to 1, the RX selects the mis_3ms_1[4:0] and mis_win16_1[3:0].

pat_kill_loc: (R/W)
The pat_kill_loc bit affects LOS detection. See details in Section 2.

pat_kill_los: (R/W)
The pat_kill_los bit affects LOS detection. See details in Section 2.

pn_fixed: (R/W)
The pn_fixed bit is used to control RX pattern detection. If this bit is set to 0, the RX detects XOR type patterns in the PN (Pseudo-random Number) machine. If this bit is set to 1, the RX detects repetitive patterns in the PN machine.

pn_qrss: (R/W)
The pn_qrss bit is used to control RX pattern detection. If this bit is set to 0, there is no action. If this bit is set to 1, there is suppression of a 14 consecutive zeros (QRSS) pattern.

pn_m64_56: (R/W)
The pn_m64_56 bit is used to control RX pattern detection. If this bit is set to 0, the RX uses all 8 bits in a DS0 for PN matching. If this bit is set to 1, the RX treats the DS0 as a 56K channel (does not use LSB).

pn_invert: (R/W)
The pn_invert bit is used to control RX pattern detection. If this bit is set to 0, the RX does not invert line data before PN pattern detection. If this bit is set to 1, the RX inverts line data before PN pattern detection.

pn_fullt1: (R/W)
The pn_fullt1 bit is used to control RX pattern detection. If this bit is set to 0, the RX uses the pn_ena[23:0] bits specification for each DS0. If this bit is set to 1, the RX uses the entire T1 (1.544 Mbps) in PN pattern detection.

exz_ena: (R/W)
The exz_ena bit is used to control the RX coding error count, coderr[20:0], in COUNT_RAM register. If this bit is set to 0, the RX only counts BPVs as coding violations. If this bit is set to 1, the RX enables counting of excess zeros as coding violations (more than 7 zeros in B8ZS, more than 15 zeros in AMI).

pn_tap[4:0], pn_length[4:0]: (R/W)
The pn_tap[4:0] bits determine the feedback tap position where the RX XOR tap should be inserted.

The pn_length[4:0] bits determine the length of the RX PN (pseudo random pattern) polynomial being detected.

The RX can detect the following standardized pseudo-random patterns:

Pattern Type	pn_length[4:0](Hex)	pn_tap[4:0](Hex)	pn_invert
2^3 -1	0x02	0x00	0
2^4 -1	0x03	0x00	0
2^5 -1	0x04	0x01	0
2^6 -1	0x05	0x04	0
2^7 -1	0x06	0x00	0
2^7 -1 (Fractional T1 LB Activate)	0x06	0x03	0
2^7 -1 (Fractional T1 LB Deactivate)	0x06	0x03	1
2^9 -1 (O.153)	0x08	0x04	0
2^{10} -1	0x09	0x02	0
2^{11} -1 (O.152, O.153)	0x0A	0x08	0
2^{15} -1 (O.151)	0x0E	0x0D	1
2^{17} -1	0x10	0x02	0
2^{18} -1	0x11	0x06	0
2^{20} -1 (O.153)	0x13	0x02	0
2^{20} -1 (O.151, pn_qrss = 1)	0x13	0x10	0
2^{21} -1	0x14	0x01	0
2^{22} -1	0x15	0x00	0
2^{23} -1 (O.151)	0x16	0x11	1
2^{25} -1	0x18	0x02	0
2^{28} -1	0x1B	0x02	0
2^{29} -1	0x1C	0x01	0
2^{31} -1	0x1E	0x02	0

P-N_Shift[31:0]: (R/W)
This 32-bit register is used for recognizing pseudo random number patterns in incoming RX data. This register is for state machine private access; no CPU initialization is needed.

inj_idle: (R/W)
The inj_idle bit controls insertion of the IDLE code for this particular RX DS1 channel. If this bit is set to 0, there is no action. If this bit is set to 1, the IDLE code is inserted downstream for this DS1 channel.

inj_ais: (R/W)
The inj_ais bit controls insertion of the AIS (unframed all 1) code for this particular RX DS1 channel. If this bit is set to 0, there is no action. If this bit is set to 1, the AIS code is inserted downstream for this DS1 channel.

sync_sf_esf: (R/W)
The sync_sf_esfbit is used to select the output timing for system receive sync SYSRXSYNC. If this bit is set to 0, the SYSRXSYNC output is 125 us. If this bit is set to 1, the SYSRXSYNC output is either 1.5 ms for SF format, or 3 ms for ESF format.

sig_inj1: (R/W)
The sig_inj1 bit controls the insertion of 1’s into signaling bit positions for this particular RX DS1 channel. If this bit is set to 0, there is no action. If this bit is set to 1, 1’s are inserted into signaling bit positions of RX data being sent to the system side.

auto_ais_red: (R/W)
The auto_ais_red bit controls the automatic insertion of AIS (all 1's pattern) into RX data for this particular RX DS1. If this bit is set to 0, there is no action. If this bit is set to 1, the AIS (all 1's pattern) is inserted automatically into RX data being sent to the system side upon a RED alarm condition.

auto_ais_los: (R/W)
The auto_ais_los bit controls the automatic insertion of AIS (all 1's pattern) into RX data for this particular RX DS1. If this bit is set to 0, there is no action. If this bit is set to 1, the AIS (all 1's pattern) is inserted automatically into RX data being sent to the system side upon an LOS alarm condition.

offset_ss[1:0]: (R/W)
The offset_ss[1:0] bits are used to select the system receive sync SYSRXSYNC offset from the normal position, as shown in the following table:

offset_ss[1:0]	SYSRXSYNC Offset
00	No offset
01	+1
10	-2
11	-1

3.3 TX State Machine, Registers and RAM

3.3.1 Per Chip Registers

Register Name: TXMISC - Tx Miscellaneous Control
Register Address:0x18

Bit	Type	Function	Reset
7	R/W	frc_dtm_mode	0
6	R/W	tx_fdl_ena	0
5	R/W	tx_tdme_bypass	0
4	R/W	tx_sigbus_ena	0
3:2	R/W	ssp[1:0]	0x0
1:0	RO	sysclkmode[1:0]	0x0

frc_dtm_mode:
The frc_dtm_mode bit is used to force the TX side into Data Termination Mode while the RX side remains in SlipBuffer Mode (SYSCLKMODE[1:0] = 11, see Table 1.9 and Table 1.10).

tx_fdl_ena:
The tx_fdl_ena bit is used to enable the TX FDL. When this bit is set to logical 1, the TX FDL is enabled. When this bit is set to logical 0, the TX_FDL is disabled.

tx_tdme_bypass:
The tx_tdme_bypass bit is used to bypass a stage of data delay (one of 45 MHz clock) in the on-chip TX TDMe system side incoming data flow. This bit changes the relationship of MUXTXDATA to the MUXTXSYNC45 pulse in the pin-efficient muxed data mode. When this bit is set to logical 1, the muxtxdata staging register is bypassed. When this bit is set to logical 0, the muxtxdata staging register is part of the internal TX TDMe system side data flow.

tx_sigbus_ena:
The tx_sigbus_ena bit is used to enable the TX signaling bus. When this bit is set to logical 1, the pins SYSTXSIG[27:0] are used as the System Transmit Signaling Input Bus in Slip Buffer mode.

ssp[1:0]:
The bits ssp[1:0] allow software to program the system backplane clock speed. This is valid only in Slip Buffer mode.

ssp[1:0]	System Backplane Clock/sync
00	1.544 MHz
01	2.048 MHz
10	4.096 MHz
11	8.192 MHz

sysclkmode[1:0]:
These bits are read only. Their value indicates the status of the system clock mode selection input pins SYSCLKMODE[1:0] as follows: 00 = Data Termination, 01 Low Latency, 10 = Pin Efficient and 11 = Slip Buffer.

3.3.2 TX RAM Registers

The TX State RAM is 32 bits wide. Eight longwords (32 bytes) are used per DS1 channel, for a total size of 224 longwords (8 x 28 channels). If the TX State RAM is located at absolute address TXS_RAM_BASE in the CPU's memory space, a channel n (from 0 to 27) will have a base address of [TXS_RAM_BASE + (n x 32)]. As described earlier, TXS_RAM_BASE = 0x4000 (this is a base address for channel 0).

During CPU indirect write access to this TXS_RAM. The following table is valid for writes to locations 08, 0C, 10, 14, 18 of each channel.

bmask[3:0] (in CPUICMD bits [7:4])	**TXS_RAM bits are written**
0000 - 1110	Invalid byte enable combination, no writes occurs.
1111	Bits [31:00]

Register Name: TXS_RAM_CH0 .. TXS_RAM_CH27 (28 DS1's) - TX State RAM
Register Address: 0x4000, 0x4020, 0x4040, 0x4060, 0x4080, 0x40A0, 0x40C0, 0x40E0, 0x4100, 0x4120, 0x4140, 0x4160, 0x4180, 0x41A0, 0x41C0, 0x41E0, 0x4200, 0x4220, 0x4240, 0x4260, 0x4280, 0x42A0, 0x42C0, 0x42E0, 0x4300, 0x4320, 0x4340, 0x4360 (28 Locations)

Address Location for 8 Longwords (Hex)	Bit Function																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
00	Reserved																															
04	Reserved																															
08	pn_shr [31:0]																															
0C	ais_force	yel_force	fbit_pass	cbit_pass	dbit_pass	offset_ss		sync_sf_esf	alt_yel	auto_yel_red	auto_yel_los	pn_fixed	m64_56	Unused	sf_esf	pn_qrss	pn_length[4:0]					pn_tap[4:0]					pn_run	mkbpverr	mkcbiterr	mkfbiterr	ins_pnerr	soft_reset
10	frc_ds0_0 [3:0]				frc_ds0_1 [3:0]				frc_ds0_2 [3:0]				frc_ds0_3 [3:0]				frc_ds0_4 [3:0]				frc_ds0_5 [3:0]				frc_ds0_6 [3:0]				frc_ds0_7 [3:0]			
14	frc_ds0_8 [3:0]				frc_ds0_9 [3:0]				frc_ds0_10 [3:0]				frc_ds0_11 [3:0]				frc_ds0_12 [3:0]				frc_ds0_13 [3:0]				frc_ds0_14 [3:0]				frc_ds0_15 [3:0]			
18	frc_ds0_16 [3:0]				frc_ds0_17 [3:0]				frc_ds0_18 [3:0]				frc_ds0_19 [3:0]				frc_ds0_20 [3:0]				frc_ds0_21 [3:0]				frc_ds0_22 [3:0]				frc_ds0_23 [3:0]			
1C	Unused																														j1_mode	pn_ftoo

Reserved:
The reserved bits are used for TX State RAM implementation; these locations should not be programmed.

pn_shr[31:0]: (R/W)
The pn_shr[31:0] bits are used for XOR tree type or repetitive pattern type generation.

When a new PN pattern is programmed, all 1's must be written to this register. When a new repetitive pattern is programmed, the pattern to be transmitted must be written to this location.

soft_reset: (R/W)
The soft_reset bit is set to 1 to initialize the TX framer. The bit auto-clears itself when initialization is complete.

ins_pnerr: (R/W)
The ins_pnerr bit controls the insertion of a programmable bit error. If this bit is set to 0, there is no action. If this bit is set to 1, the TX inserts a single error in the PN pattern. This bit auto-clears itself when the operation is done.

mkfbiterr: (R/W)
The mkfbiterr bit controls the insertion of a programmable F-bit error. If this bit is set to 0, there is no action. If this bit is set to 1, the TX creates a single F-bit error to the line side. This bit auto-clears itself when the operation is done.

mkcbiterr: (R/W)
The mkcbiterr bit controls the insertion of a programmable CRC6-bit error. If this bit is set to 0, there is no action. If this bit is set to 1, the TX creates a single CRC6-bit error (valid in ESF mode only) to the line side. This bit auto-clears itself when the operation is done.

mkbpverr: (R/W)
The mkbpverr bit controls the insertion of a programmable BPV encoding error. If this bit is set to 0, there is no action. If this bit is set to 1, the TX creates a single BPV encoding error to the line side (valid in bipolar rail mode only). This bit auto-clears itself when the operation is done.

pn_run: (R/W)
The pn_run bit controls PN machine operation. If this bit is set to 0, the TX PN machine is idle. If this bit is set to 1, the TX PN machine insertion begins to run.

pn_tap[4:0], pn_length[4:0]: (R/W)
The pn_tap[4:0] bits determine the feedback tap position where the TX XOR tap should be inserted.

The pn_length[4:0] bits determine the length of the TX generated PN (pseudo random pattern) polynomial.

The user can generate the following standardized pseudo random patterns:

Pattern Type	**pn_length[4:0] (Hex)**	**pn_tap[4:0] (Hex)**	**frc_ds0_X[3:0] (Binary)**
2^3 -1	0x02	0x00	1001
2^4 -1	0x03	0x00	1001
2^5 -1	0x04	0x01	1001
2^6 -1	0x05	0x04	1001
2^7 -1	0x06	0x00	1001
2^7 -1 (Fractional T1 LB Activate)	0x06	0x03	1001
2^7 -1 (Fractional T1 LB Deactivate)	0x06	0x03	1000
2^9 -1 (O.153)	0x08	0x04	1001
2^{10} -1	0x09	0x02	1001
2^{11} -1 (O.152, O.153)	0x0A	0x08	1001
2^{15} -1 (O.151)	0x0E	0x0D	1000
2^{17} -1	0x10	0x02	1001
2^{18} -1	0x11	0x06	1001
2^{20} -1 (O.153)	0x13	0x02	1001
2^{20} -1 (O.151, pn_qrss = 1)	0x13	0x10	1001
2^{21} -1	0x14	0x01	1001
2^{22} -1	0x15	0x00	1001
2^{23} -1 (O.151)	0x16	0x11	1000
2^{25} -1	0x18	0x02	1001
2^{28} -1	0x1B	0x02	1001
2^{29} -1	0x1C	0x01	1001
2^{31} -1	0x1E	0x02	1001

pn_qrss: (R/W)
The pn_qrss bit is used to control TX pattern generation. If this bit is set to 0, there is no action. If this bit is set to 1, there is suppression of a 14 consecutive zeros (QRSS) pattern.

sf_esf: (R/W)
The sf_esf bit is used to select the TX frame format. If this bit is set to 0, the TX selects SF mode. If this bit is set to 1, the TX selects ESF mode.

m64_56: (R/W)
The m64_56 bit is used to select the TX rate for all 24 DS0's. If this bit is set to 0, the TX selects 64 Kbps mode. If this bit is set to 1, the TX selects 56 Kbps mode.

pn_fixed: (R/W)
The pn_fixed bit is used to control TX pattern generation. If this bit is set to 0, the TX generates XOR type patterns in the PN (Pseudo-random Number) machine. If this bit is set to 1, the TX generates repetitive and fixed patterns in the PN machine.

auto_yel_los: (R/W)
The auto_yel_los bit controls the automatic transmission of a YELLOW alarm condition. If this bit is set to 0, transmission requires CPU intervention. If this bit is set to 1, the TX output automatically transmits SF YELLOW if the RX side is LOS.

auto_yel_red: (R/W)
The auto_yel_red bit controls the automatic transmission of a YELLOW alarm condition. If this bit is set to 0, transmission requires CPU intervention. If this bit is set to 1, the TX output automatically transmits SF YELLOW if the RX side is RED.

alt_yel: (R/W)
The alt_yel bit controls the transmission format of a YELLOW alarm. If this bit is set to 0, the YELLOW alarm insertion in SF is: the bit 2 in every DS0 channel is 0. If this bit is set to 1, the YELLOW alarm insertion in SF is: the 6th Fs bit is 1 instead of 0 (Japanese standard).

sync_sf_esf: (R/W)
The sync_sf_esf bit is used to select the output timing for system transmit sync SYSTXSYNC. If this bit is set to 0, the SYSTXSYNC output is 125 us. If this bit is set to 1, the SYSTXSYNC output is either 1.5 ms for SF format, or 3 ms for ESF format.

offset_ss[1:0]: (R/W)
The offset_ss[1:0] bits are used to select the system transmit sync SYSTXSYNC offset from the normal position, as shown in the following table:

offset_ss[1:0]	**SYSTXSYNC Offset (in Data Termination and Pin Efficient Mode)**	**SYSTXSYNC Offset (in Low Latency Mode)**
00	No offset	No offset
01	+1 (one clock delay after)	+1 (one clock delay after)
10	-2 (two clocks in advance)	+2 (two clocks delay after)
11	-1 (one clock in advance)	-1 (one clock in advance)

dbit_pass: (R/W)
The dbit_pass bit is used to control the TX d-bit (facility data link) function. If this bit is set to 0, the TX overwrites the system side transmit data d-bit positions with HDLC data (valid in ESF mode only). If this bit is set to 1, the TX passes on system side data in the d-bit positions.

cbit_pass: (R/W)
The cbit_pass bit is used to control the TX CRC-6 bits function. If this bit is set to 0, the TX overwrites system side transmit data c-bit positions with the correct CRC-6 (valid in ESF mode only). If this bit is set to 1, the TX passes through system side data in the c-bit positions.

fbit_pass: (R/W)
The fbit_pass bit is used to control the TX F-bit function. If this bit is set to 0, the TX overwrites system side transmit data F-bit positions with correct SF or ESF framing bits. If this bit is set to 1, the TX passes on system side data in the F-bit positions.

yel_force: (R/W)
The yel_force bit controls forced transmission of a YELLOW alarm condition. If this bit is set to 0, there is no action. If this bit is set to 1, the TX forces transmission of an SF or alternate (Japanese) YELLOW alarm.

ais_force: (R/W)
The ais_force bit controls forced transmission of an AIS (unframed all 1's) condition. If this bit is set to 0, there is no action. If this bit is set to 1, the TX forces transmission of an AIS condition.

frc_ds0_0[3:0] .. frc_ds0_23[3:0]: (R/W)
The frc_ds0_X[3:0] (where X indicates the DS0 number) bits are used to control the TX per-DS0 data format, as shown in the table below. Normally, DS0s that are configured to be forced to the PN machine output will use either an all-inverted or all-non-inverted PN pattern.

frc_ds0_X[3:0] (X = 0 .. 23)		
frc_ds0_X[3] - Function	frc_ds0_X[2] - Function	frc_ds0_X[1:0] - Function
0 - Transmit data from system side	0 - Don’t invert TX data	00 - Normal transmit per-ds0 data
		01 - Bit 8 always forced to 1
	1 - invert TX data	10 - GTE
		11 - Bell
1 - Force transmission of a pattern (overwriting system side TX data)	0 - System side TX clock (SYSTXCLK) is present	00 - Invert PN machine output
		01 - Don’t invert PN machine output
	1 - Reserved (don’t use)	10 - TX Digital milliwatt pattern
		11 - TX Idle code

Notes:

GTE: Bit 8 of an all-zero octet is forced to 1, except in signaling bytes, where bit 7 is forced to 1.
Bell: Bit 7 of an all-zero octet is forced to 1.

pn_ftoo: (R/W)
The pn_ftoo bit is used to control the TX F-bit operation. If this bit is set to 0, the F-bit is not included in the PN pattern generation. If this bit is set to 1, the F-bit is included in the PN pattern generation. Note: When both pn_ftoo and pn_fixed bits are set to 1, the pn_tap[4:1] bits are ignored. If pn_tap[0] is also set to 1, the F-bit overwrites the pattern.

j1_mode:(R/W)
The j1_mode bit selects either T1 or J1 mode. If this bit is set to 0 then T1 is selected. If this bit is set to 1 then J1 is selected.

3.4 Error Machine Registers and RAM

3.4.1 Per Chip Registers

Register Name: ERRMISC - Error Miscellaneous Control
Register Address: 0x20

Bit	Type	Function	Reset
7	RO	shdw_sel	0
6	R/W	use_cpu_addr	0
5	R/W	omit_ais	0
4	R/W	omit_los	0
3	R/W	enb_wkr	0
2		Unused	X
1	R/W	force_dbuf	0
0	R/W	auto_dbuf	1

shdw_sel:
The shdw_sel bit is read only and shows the current shadow bank location that the CPU should access (the one other than the bank being used by the hardware). When this bit is logical 0, the current shadow bank location is 0 to 3. When this bit is logical 1, the current shadow bank location is 4 to 7. A program can always automatically read the correct (not-in-use) bank by setting the use_cpu_addr = 0.

use_cpu_addr:
The use_cpu_addr bit is used to select the correct bank. If this bit is set to logical 0, the CPU will automatically select the correct bank to read. If this bit is set to logical 1, the CPU address is used to select the bank to read.

omit_ais:
Controls the ability of AIS to stop incrementation of the RX red accumulator. If set to ‘0’ and if the channel is in-OOF and not in-AIS, the red accumulator will increment (if not saturated). If set to ‘1’ and if the channel is in-OOF, the RX red accumulator will increment (if not saturated).

omit_los:
Used to enable the red_accum[20:0] RED accumulator in the FLAG_RAM_BASE register. If this bit is set to logical 1, the LOS is omitted. If this bit is set to logical 0, the LOS is not omitted.

enb_wkr:
The enb_wkr bit is used to enable the err_walker_fsm state machine to monitor the error conditions of all 28 DS1s every 10 mS. If this bit is set to logical 1, the err_walker_fsm is enabled. If this bit is set to logical 0, the err_walker_fsm is disabled.

force_dbuf, auto_dbuf:
Used to control the internal double buffers operation, as shown in the following table:

force_dbuf, auto_dbuf	Double Buffers Operation
00	NOP (No Operation)
01	Use internal 1-second counter to switch double buffered counters
10	CPU forced switch, and clears to 00 (NOP) when done
11	Use external signal pin (CPU_DBUF) to switch double buffered counters

Register Name: HALFMS_LSCALER, HALFMS_HSCALER - Half Millisecond Low/High Byte Scaler
Register Address: 0xF0, 0xF1

Bit	Type	Function	Reset
7:0	R/W	halfms_lscaler[7:0]	0xE6
7:0	R/W	halfms_hscaler[7:0]	0x57

halfms_lscaler[7:0], halfms_hscaler[7:0:
The contents of these two registers are used for a one-half millisecond timer. The value {hscaler, lscaler} controls the divisor of the 44.736 MHz clock to generate the 0.5 mS internal clock used for pattern timing. For write accesses, the halfms_lscaler low byte must be written firs, followed by the halfms_hscaler high byte.

Register Name: RED_SET_ADD - RED Set Threshold Pattern Match
Register Address:0xA4

Bit	Type	Function	Reset
7		Unused	X
6:0	R/W	red_set_add [6:0]	0x00

red_set_add[6:0]:
The contents of this register are used for declaring and clearing various pattern match conditions. (See detailed description in Section 2.8.2)

Register Name: RED_CLR_SUB - RED Clear Threshold Pattern Match
Register Address: 0xA5

Bit	Type	Function	Reset
7		Unused	X
6:0	R/W	red_clr_sub [6:0]	0x00

red_clr_sub[6:0]:
The contents of this register are used for declaring and clearing various pattern match conditions. (See detailed description in Section 2.8.2)

Register Name: AIS_SET_THR - AIS Set Threshold Pattern Match
Register Address: 0xA6

Bit	Type	Function	Reset
7		Unused	X
6:0	R/W	ais_set_thr [6:0]	0x00

ais_set_thr[6:0]:
The contents of this register are used for declaring and clearing various pattern match conditions. (See detailed description in Section 2.8.3)

Register Name: AIS_CLR_THR - AIS Clear Threshold Pattern Match
Register Address: 0xA7

Bit	Type	Function	Reset
7		Unused	X
6:0	R/W	ais_clr_thr [6:0]	0x00

ais_clr_thr[6:0]:
The contents of this register are used for declaring and clearing various pattern match conditions. (See detailed description in Section 2.8.3)

Register Name: RED_CLR_SUB0 - RED Clear Threshold Pattern Match 0
Register Address: 0xA8

Bit	Type	Function	Reset
7		Unused	X
6:0	R/W	red_clr_sub0 [6:0]	0x00

red_clr_sub0[6:0]:
The contents of this register are used for declaring and clearing various pattern match conditions. (See detailed description in Section 2.8.2)

Register Name: YEL_SET_THR
- Yellow Detection Set Threshold Timing for RX Direction
Register Address:0xB0

Bit	Type	Function	Reset
7:6		Unused	X
5:0	R/W	yel_set_thr [5:0]	0x00

yel_set_thr[5:0]:
The contents of this register are used for declaring and clearing various pattern match conditions. (See detailed description in Section 2.8.4)

YEL_SET_THR is used for SF Yellow detection timing.

Register Name: CSULU_SET_THR
- CSU LoopUp Set Threshold Timing for RX Direction
Register Address:0xB1

Bit	Type	Function	Reset
7:6		Unused	X
5:0	R/W	csulu_set_thr [5:0]	0x00

csulu_set_thr[5:0]:
The contents of this register are used for declaring and clearing various pattern match conditions. (See detailed description in Section 2.8.5)

CSULU_SET_THR is used for CSU loopup (00001) detection timing.

Register Name: CSULD_SET_THR
- CSU Loop-Down Set Threshold Timing for RX Direction
Register Address: 0xB2

Bit	Type	Function	Reset
7:6		Unused	X
5:0	R/W	csuld_set_thr [5:0]	0x00

csuld_set_thr[5:0]:
The contents of this register are used for declaring and clearing various pattern match conditions.

CSULD_SET_THR is used for CSU loopdown (001) detection timing.

Register Name: NILU_SET_THR
- NI LoopUp Set Threshold Timing for RX Direction
Register Address: 0xB3

Bit	Type	Function	Reset
7:6		Unused	X
5:0	R/W	nilu_set_thr [5:0]	0x00

nilu_set_thr[5:0]:
The contents of this register are used for declaring and clearing various pattern match conditions.

NILU_SET_THR is used for NI loopup (00011) detection timing.

Register Name: NILD_SET_THR
- NI Loop-Down Set Threshold Timing for RX Direction
Register Address: 0xB4

Bit	Type	Function	Reset
7:6		Unused	X
5:0	R/W	nild_set_thr [5:0]	0x00

nild_set_thr[5:0]:
The contents of this register are used for declaring and clearing various pattern match conditions.

NILD_SET_THR is used for NI loopdown (00111) detection timing.

Register Name: PAT5_SET_THR
- Pattern (10100) Set Threshold Timing for RX Direction
Register Address: 0xB5

Bit	Type	Function	Reset
7:6		Unused	X
5:0	R/W	pat5_set_thr [5:0]	0x00

pat5_set_thr[5:0]:
The contents of this register are used for declaring and clearing pattern (10100) match conditions.

PAT5_SET_THR is used for pattern (10100) detection timing.

Register Name: IDLE_SET_THR
- IDLE Code Set Threshold Timing for RX Direction
Register Address: 0xB6

Bit	Type	Function	Reset
7:6		Unused	X
5:0	R/W	idle_set_thr [5:0]	0x00

idle_set_thr[5:0]:
The contents of this register are used for declaring and clearing various pattern match conditions.

IDLE_SET_THR is used for idle code (DS0 aligned 00010111) detection timing.

Register Name: RX_JT1_SFYEL[07:00], RX_JT1_SFYEL[15:08], RX_JT1_SFYEL[23:16], RX_JT1_SFYEL[27:24]
- RX JT1 SF Yellow Status for DS1 Channels
Register Address: 0x28, 0x29, 0x2A, 0x2B

Register Name (Address)	Bit Function							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RX_JT1_SFYEL[07:00] (0x28)	jt1_sfyel_07	jt1_sfyel_06	jt1_sfyel_05	jt1_sfyel_04	jt1_sfyel_03	jt1_sfyel_02	jt1_sfyel_01	jt1_sfyel_00
RX_JT1_SFYEL[15:08] (0x29)	jt1_sfyel_15	jt1_sfyel_14	jt1_sfyel_13	jt1_sfyel_12	jt1_sfyel_11	jt1_sfyel_10	jt1_sfyel_09	jt1_sfyel_08
RX_JT1_SFYEL[23:16] (0x2A)	jt1_sfyel_23	jt1_sfyel_22	jt1_sfyel_21	jt1_sfyel_20	jt1_sfyel_19	jt1_sfyel_18	jt1_sfyel_17	jt1_sfyel_16
RX_JT1_SFYEL[27:24] (0x2B)	Unused	Unused	Unused	Unused	jt1_sfyel_27	jt1_sfyel_26	jt1_sfyel_25	jt1_sfyel_24

Each of these register bits is used to indicate RX JT1 SF Yellow status.

jt1_sfyel_[27:00]:
The contents of these (RO) registers are used for declaring Per-DS1 JT1 SF Yellow Alarm Condition.

3.4.2 Performance-monitor Flags and Counts RAMs

The Performance-monitor FLAG (Errors) RAM is 32 bits wide. Two longwords (8 bytes) are used per DS1 channel, for a total size of 56 (2 x 28 channels) longwords. If the FLAG (Errors) RAM is located at absolute address FLAG_RAM_BASE in the CPU's memory space, a channel n (from 0 to 27) has a base address of [FLAG_RAM_BASE + (n x 8)].

The Performance-monitor COUNT RAM is 32 bits wide. Eight longwords (32 bytes) are used per DS1 channel, for a total size of 224 (8 x 28 channels) longwords. If the COUNT RAM is located at absolute address COUNT_RAM_BASE in the CPU's memory space, a channel n (from 0 to 27) has a base address of [COUNT_RAM_BASE + (n x 32)]. As described earlier,

FLAG_RAM_BASE = 0x5000 (this is a base address for channel 0).
COUNT_RAM_BASE = 0x54000 (this is a base address for channel 0).

During CPU indirect write access to this FLAG_RAM, the following table is valid for writes to location 04 (intr_mask[24, 15:0]) of each channel (x = don't care).

bmask[3:0] (in CPUICMD bits [7:4])	FLAG_RAM bits are written
x000	No writes occurs
x001	Bits [07:00]
x010	Bits [15:08]
x011	Bits [15:08], [07:00]
x100	Bits [24]
x101	Bits [24], [07:00]
x110	Bits [24],[15:08]
x111	Bits [24], [15:08], [07:00]

The bmask[3:0] bits are ignored when writing to any of the following bits in the FLAG_RAM:

stky_los, stky_oof, fe_ofl, fe_ufl, rx_slip, rx_slip_ufl, tx_slip, tx_slip_ufl

This is a "write 1 to clear".

During CPU indirect write access to this COUNT_RAM, the following table is valid for writes to locations 00, 04, 08, 10, 14, 18 of each channel. In general a user should ONLY perform CPU indirect reads access to this COUNT_RAM. However, indirect write access is also supported as follows (note that valid error count values present will be overwritten).

bmask[3:0] (in CPUICMD bits [7:4])	**COUNT_RAM bits are written**
0000	No writes occurs
0001	Bits [07:00]
0011	Bits [15:08], [07:00]
0111	Bits [23:16], [15:08], [07:00]
1111	Bits [31:24], [23:16], [15:08], [07:00]

Note: For these bmask[3:0] enable bits, each bit set to ‘1’ indicates a byte write enable; ‘0’ indicates a byte write disable. Valid for byte enable combination up to 15 different byte writes.

Register Name: FLAG_RAM_CH0 .. FLAG_RAM_CH27 (28 DS1's) - Performance Monitor Flags RAM

Register Address: 0x5000, 0x5008, 0x5010, 0x5018, 0x5020, 0x5028, 0x5030, 0x5038, 0x5040, 0x5048, 0x5050, 0x5058, 0x5060, 0x5068, 0x5070, 0x5078, 0x5080, 0x5088, 0x5090, 0x5098, 0x50A0, 0x50A8, 0x50B0, 0x50B8, 0x50C0, 0x50C8, 0x50D0, 0x50D8 (28 Locations)

		Bit Function																															
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Two LW address	00	ptimed[3:0]				ais	pfound[3:0]				los	p_tstamp[5:0]						stky_los	rxfdl_esfyel	jt1_sfyel	red_accum[10:0]											oof	red
	04	tx_slip	rx_slip	fe_ufl	fe_ofl	stky_oof	tx_slip_ufl	rx_slip_ufl	intr_mask[24]	ais_pfound	ais_tstamp[6:0]							intr_mask[15:0]															

red: (RO)
The red bit is set to 1 when this DS1 channel received a RED alarm (persistent LOS or non-AIS-caused OOF).

oof: (RO)
This bit is set to 1 when the RX FSM directs readout of a detected OOF condition in this DS1 channel.

red_accum[10:0]: (RO)
These bits are used as an accumulator for state machine private use.

jt1_sfyel: (RO)
The jt1_sfyel bit is set when the RSM detects SF Yellow in JT1 Mode.

rxfdl_esfyel: (RO)
The rxfdl_esfyel bit is set when the RX_FDL detects BOC code ESF Yellow.

stky_los: (RO, W1C)
The stky_los (sticky) bit is set to 1 when RX FSM detects an RX LOS; it is write-1-to-clear by the CPU.

p_tstamp[5:0]: (RO)
The p_tstamp[5:0] bits are used as a pattern timestamp for state machine private use.

los: (RO)
Set to 1 when the RX FSM directs readout of a detected LOS condition in this DS1 channel.

pfound[3:0]: (RO)
These bits are used to indicate a raw (untimed) pattern detected by RX FSM, for state machine reserved use.

ais: (RO)
The ais bit is set to 1 when the RX FSM detects an RX AIS (unframed all 1’s) state in this DS1 channel.

ptimed[3:0]:
(RO) The ptimed[3:0] bits are used to indicate a pattern match acquired in this DS1 channel, as shown in the following table:

ptimed[3:0] (Hex)	**Pattern match acquired**
0x0	SF Yellow detected (bit2=0)
0x1	CSU loop-up pattern detected (00001..)
0x2	CSU loop-down pattern detected (001…)
0x3	NI loop-up pattern detected (00011…)
0x4	NI loop-down pattern detected (00111…)
0x5	Pattern 10100 detected
0x6	Idle code (00010111 octet aligned)
0x7 - 0xE	Ignore
0xF	NOP (no pattern detected)

intr_mask[24, 15:0]: (R/W)
Each bit in the intr_mask[24, 15:0] independently masks the corresponding bit as shown in the table below. When a mask bit is set to 1, any change in the bit being monitored causes an interrupt. For ptimed[3:0] monitoring, any change into or out of a specific encoding triggers an interrupt. For rx_slip, tx_slip, tx_slip_ufl, rx_slip_ufl, fe_ofl and fe_ufl, only the rising edge triggers the interrupt.

	intr_mask[24],intr_mask[15:0]																
Interrupt Mask Bit	24	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	jt1_sfyel	red	los	fe_ufl	fe_ofl	tx_slip	rx_slip	ais	oof	ptimed[3:0]=0xf	ptimed[3:0]=0x6	ptimed[3:0]=0x5	ptimed[3:0]=0x4	ptimed[3:0]=0x3	ptimed[3:0]=0x2	ptimed[3:0]=0x1	ptimed[3:0]=0x0

ais_tstamp[6:0]: (RO)
The ais_tstamp[6:0] bits are used as an AIS timer, for state machine private use.

ais_pfound: (RO)
The ais_pfound bit is used as an AIS pattern encoding, for state machine private use.

rx_slip_ufl: (RO, W1C)
The rx_slip_ufl bit is set to 1 when the RX slip buffer underflows.

tx_slip_ufl: (RO, W1C)
The tx_slip_ufl bit is set to 1 when the TX slip buffer underflows.

stky_oof: (RO, W1C)
The stky_oof (sticky) bit is set to 1 when the RX FSM detects an OOF; it is write-1-to-clear by the CPU.

fe_ofl: (RO, W1C)
The fe_ofl bit is set to 1 when the RX front end reports an overflow error (RX machine did not service requests soon enough).

fe_ufl: (RO, W1C)
The fe_ufl bit is set to 1 when the TX front end reports an underflow error (TX machine did not service requests soon enough).

rx_slip: (RO, W1C)
The rx_slip bit is set to 1 when the RX slip buffer suffers a slip.

tx_slip: (RO, W1C)
The tx_slip bit is set to 1 when the TX slip buffer suffers a slip.

Register Name: **COUNT_RAM_CH0 .. COUNT_RAM_CH27 (28 DS1's) - Performance Monitor Count RAM**

Register Address: **0x5400, 0x5420, 0x5440, 0x5460, 0x5480, 0x54A0, 0x54C0, 0x54E0, 0x5500, 0x5520, 0x5540, 0x5560, 0x5580, 0x55A0, 0x55C0, 0x55E0, 0x5600, 0x5620, 0x5640, 0x5660, 0x5680, 0x56A0, 0x56C0, 0x56E0, 0x5700, 0x5720, 0x5740, 0x5760 (28 Locations)**

	Bit Function																															
Address Location for 8 Longwords (Hex)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
00	e_i	e_o	Unused		cofa[3:0]				Unused			pn_err[20:0]																				
04	sefe	Unused	oof_event[5:0]						Unused			coderr[20:0]																				
08	Unused				crc_errcnt[11:0]												Unused			ferrcnt[12:0]												
0C	Unused																															
10	e_i	e_o	Un-used		cofa[3:0]				Unused			pn_err[20:0]																				
14	sefe	Unused	oof_event[5:0]						Unused			coderr[20:0]																				
18	Unused				crc_errcnt[11:0]												Unused			ferrcnt[12:0]												
1C	Unused																															

There are two identical banks of counters. One bank is used as a working (active: 0x00, 0x04, 0x08) bank and the other as a frozen (shadow: 0x10, 0x14, 0x18) copy. When an internal 1-second counter expires (or when an external 1-second pin CPU_DBUF is pulsed, or upon CPU command), the VSC9670 switches counter banks between working and frozen, and optionally interrupts the CPU. The CPU then has up to 1 second to read the frozen counts before they are overwritten. This provides enough time for the CPU to read error counters. The counters are saturating, and clear themselves at each switchover.

pn_err[20:0]: (RO)
The pn_err[20:0] bits are used to count the number of PN bit errors received in the RX direction.

cofa[3:0]: (RO)
The cofa[3:0] bits are used as a change-of-frame-alignment events counter.

e_o: (RO)
The e_o bit is set to 1 if the RX PN receiver was ever in out-of-sync mode (in the last 1 second or time elapsed from the CPU / pin trigger).

e_i: (RO)
The e_i bit is set to 1 if the RX PN receiver was ever in in-sync mode (in the last 1 second or time elapsed from the CPU / pin trigger).

coderr[20:0]: (RO)
The coderr[20:0] bits are used as a coding error counter (BPV violations; optionally, excess 0's included in count).

oof_event[5:0]: (RO)
The oof_event[5:0] bits are used as out-of-frame event counter. A non-zero count also can indicate a severely errored frame event.

sefe: (RO)
The sefe bit is set to 1 if more than 1 framing bit error was found in a multiframe (12 frames in SF, 24 in ESF).

ferrcnt[12:0]: (RO)
The ferrcnt[12:0] bits are used as an RX direction F-bit error counter.

crc_errcnt[11:0]: (RO)
The crc_errcnt[11:0] bits are used in ESF to indicate the number of multiframes during which a CRC6 error was found. The expected maximum count in a 1-second period is 333.

The pseudo-random pattern generator/receiver is intended for in-service or out-of-service line quality testing. Testing starts with this framer of TX circuitry generating a PN pattern. The remote end performs loopback, or generates a PN pattern according to the same rule. The framer of RX circuitry tries to establish pattern sync. When there is a 1-second interval with the e_i bit asserted, it means that the PN receiver was able to sync itself to the incoming pattern. If the e_o bit is asserted, this 1-second interval was so severely errored that the PN receiver dropped out of sync for some or all of the time. If the e_o bit is not asserted, the pn_err[11:0] count can be used as a quantitative measure of line quality.

3.5 FDL Machine, RAM Bits

The FDL (Facility Data Link) RAM is 32 bits wide. Thirty-two longwords (128 bytes) are used per DS1 channel, for a total size of 896 (32 x 28 channels) longwords for each of RX and TX. If the RX or TX FDL RAM is located at absolute address FDL_RX_RAM_BASE or FDL_TX_RAM_BASE in the CPU's memory space, the RX space for a channel n (from 0 to 27) has a base address of [FDL_RX_RAM_BASE + (n x 128)] and TX space for a channel n (from 0 to 27) has a base address of [FDL_TX_RAM_BASE + (n x 128)]. As described earlier,

FDL_RX_RAM_BASE = 0x6000 (this is a base address for RX channel 0).
FDL_TX_RAM_BASE = 0x8000 (this is a base address for TX channel 0).

During CPU indirect write access to this FDL_RX_RAM (undef_fdl = 0), the following table is valid for writes to location 00 of each channel.

bmask[3:0] (in CPUICMD bits [7:4])	**FDL_RX_RAM bits are written**
0000	No writes occurs
0001	Bits [07:00]
0011	Bits [15:08], [07:00]
0111	Bits [23:16], [15:08], [07:00]
1111	Bits [31:24], [23:16], [15:08], [07:00]

Note:
For these bmask[3:0] enable bits, each bit set to '1' indicates a byte write enable; '0' indicates a byte write disable. Valid for byte enable combination up to 15 different byte writes.

During CPU indirect write access to this FDL_TX_RAM (undef_fdl = 0), the following table is valid for writes to location 00 or location 08 of each channel.

bmask[3:0] (in CPUICMD bits [7:4])	**FDL_TX_RAM bits are written**
0000	No writes occurs
0001	Bits [07:00]
0011	Bits [15:08], [07:00]
0111	Bits [23:16], [15:08], [07:00]
1111	Bits [31:24], [23:16], [15:08], [07:00]

Note:
For these bmask[3:0] enable bits, each bit set to '1' indicates a byte write enable; '0' indicates a byte write disable. Valid for byte enable combination up to 15 different byte writes.

Register Name: RXFDL_ESFYEL[07:00], RXFDL_ESFYEL[15:08], RXFDL_ESFYEL[23:16], RXFDL_ESFYEL[27:24] - RX FDL ESF Yellow Status for DS1 Channels
Register Address: 0x2C, 0x2D, 0x2E, 0x2F

Register Name (Address)	Bit Function							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXFDL_ESFYEL[07:00] (0x2C)	esfyel_07	esfyel_06	esfyel_05	esfyel_04	esfyel_03	esfyel_02	esfyel_01	esfyel_00
RXFDL_ESFYEL[15:08] (0x2D)	esfyel_15	esfyel_14	esfyel_13	esfyel_12	esfyel_11	esfyel_10	esfyel_09	esfyel_08
RXFDL_ESFYEL[23:16] (0x2E)	esfyel_23	esfyel_22	esfyel_21	esfyel_20	esfyel_19	esfyel_18	esfyel_17	esfyel_16
RXFDL_ESFYEL[27:24] (0x2F)	Unused	Unused	Unused	Unused	esfyel_27	esfyel_26	esfyel_25	esfyel_24

Each of these register bits is used to indicate RX FDL ESF Yellow status.

esfyel[27:00]:
The contents of these (RO) registers are used for declaring Per-DS1 T1 ESF Yellow Alarm Condition.

Register Name: FDL_RX_RAM_CH0 .. FDL_RX_RAM_CH27 (28 DS1's - Facility Data Link RX RAM) (indirect access with undef_fdl = 0 in CPUICMD)

Register Address: 0x6000, 0x6080, 0x6100, 0x6180, 0x6200, 0x6280, 0x6300, 0x6380, 0x6400, 0x6480, 0x6500, 0x6580, 0x6600, 0x6680, 0x6700, 0x6780, 0x6800, 0x6880, 0x6900, 0x6980, 0x6A00, 0x6A80, 0x6B00, 0x6B80, 0x6C00, 0x6C80, 0x6D00, 0x6D80 (28 Locations)

Address Location for 32 Longwords (Hex)	Bit Function																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
00	Reserved (Set bmask[3:2]=00 in CPUICMD register when CPU writes to this location)																rx_ofl	rxfsize [6:0]							Un-used		esfyel_mask	int_mask	int_every	disc_err	rx_xpar	rx_ena
04	Reserved																															
08	Reserved																															
0C																																
10																																
14																																
18																																
1C																																
20																																
24																																
28																																
2C																																
30																																
34																																
38																																
3C																																
40																																
44																																
48																																
4C																																
50																																
54																																
58																																
5C																																
60																																
64																																
68																																
6C																																
70																																
74																																
78																																
7C																																

rx_ena: (R/W)
The rx_ena bit selects the RX FDL operation. When rx_ena is set to logical 1, the RX FDL state machine is enabled to recognize bit oriented or HDLC messages. When rx_ena is set to logical 0, the RX FDL state machine is disabled; no processing is performed and there is no CPU update. Toggling the rx_ena bit is a way to initialize pointers and the FIFO sub-packet protocol.

rx_xpar: (R/W)
The rx_xpar bit selects the RX FDL as transparent mode. When rx_xpar is set to logical 1, the RX FDL state machine is enabled as transparent mode, which hands off incoming FDL bits to the CPU without processing (using 31-byte sub-packets). When rx_xpar is set to logical 0, the RX FDL state machine operates using bit oriented / HDLC messages.

disc_err: (R/W)
The disc_err bit controls the RX FDL errored packets. When disc_err is set to logical 1, the RX FDL state machine discards errored packets which are less than 31 bytes. When disc_err is set to logical 0, the RX FDL state machine enters errored packets and status into the RX FDL FIFO.

int_every: (R/W)
The int_every bit controls the RX FDL interrupt. When int_every is set to logical 1, the RX FDL state machine will interrupt the CPU every time an entry is placed in the RX FDL FIFO. When int_every is set to logical 0, the RX FDL state machine will interrupt only when crossing the high water mark (half full); that is, when rxfsize[6:0] increases past 64 bytes.

int_mask: (R/W)
The int_mask bit masks the RX FDL interrupt. When int_mask is set to logical 1, the RX FDL state machine will generate an interrupt (according to int_every). When int_mask is set to logical 0, the RX FDL state machine will not generate an interrupt.

esfyel_mask: (R/W)
The esfyel_mask bit masks the RX FDL ESF Yellow interrupt. When esfyel_mask is set to logical 1, the RX FDL ESF state machine will generate an interrupt. When esfyel_mask is set to logical 0, the RX FDL ESF state machine will not generate an interrupt.

rxfsize[6:0]: (RO)
The rxfsize[6:0] bits indicate the RX FDL FIFO size in bytes. After a new sub-packet is completely added to this FIFO, rxfsize[6:0] is incremented by that sub-packet's size, and rxfsize[6:0] is decremented by a CPU indirect read.

rx_ofl: (RO, W1C)
The rx_ofl bit is set to 1 when the RX FDL FIFO overflows (sticky, write-1-to-clear). If RX FDL FIFO overflow is reached while a sub-packet is being inserted, the portion of the sub-packet already written into the RX FDL FIFO is abandoned; that is, the internal write pointer is moved back to just after the end of the previous sub-packet. Hence the programmer might find the rx_ofl bit set, even though rxfsize[6:0] is less than the maximum size. The rx_ofl bit is set to 0 when RX FDL FIFO has not overflowed.

Reserved bits are used for RX FDL FIFO implementation. They should not be programmed.

RX FDL FIFO: (RO)
The RX FDL FIFO is used for storing received FDL data, together with a sub-packet ID byte, or an error ID byte if rx_err bit is set. It can hold up to 116 bytes (29 LW x 4 bytes) per DS1 channel. The FDL (4 Kb/s) occupies the F bit of every odd frame of an extended superframe (ESF). The CPU can read up to 4 bytes of RX FDL FIFO data with undef_fdl = 1 in the CPUICMD register (see below).

Register Name: **FDL_RX_RAM_CH0 .. FDL_RX_RAM_CH27 (28 DS1s)**
- Facility Data Link RX RAM
(indirect access with undef_fdl = 1 in CPUICMD)
Register Address: **0x6000, 0x6080, 0x6100, 0x6180, 0x6200, 0x6280, 0x6300,**
0x6380, 0x6400, 0x6480, 0x6500, 0x6580, 0x6600, 0x6680,
0x6700, 0x6780, 0x6800, 0x6880, 0x6900, 0x6980, 0x6A00,
0x6A80, 0x6B00, 0x6B80, 0x6C00, 0x6C80, 0x6D00, 0x6D80
(28 Locations)

		Bit Function																															
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
One LW addr.	00	First_byte								Second_byte								Third_Byte								Fourth_byte							

It is legal for the CPU to perform an indirect read only with undef_fdl = 1 for addresses of the form [FDL_RX_RAM_BASE + (n x 128)], where n = 0 .. 27. This type of indirect read is used as a way of collecting RX FDL FIFO bytes, while maintaining pointers consistently.

First_byte, Second_byte, Third_byte, Fourth_byte: (RO)
When a CPU indirect read access is performed with undef_fdl = 1 in the CPUICMD register to this location, the returned longword contains up to 4 bytes of RX FDL FIFO data. Each time a read is performed, rxfsize[6:0] is decremented by the number of bytes handed off to the CPU section. If there are less than 4 bytes to give, the returned byte enables the bits bmask[3:0] in the CPUICMD register; these bits are used to indicate which bytes are valid.

The first octet of a CPU read will always be a sub-packet-ID byte. The ID byte contains a bit field that describes whether this sub-packet is a BOC type or HDLC type, and another bit field that specifies the number of bytes to follow in this sub-packet. This allows the CPU to determine the location of the next sub-packet ID byte.RX Sub-Packet ID Byte Format:

Bit	**Name**	**Function**
7	rx_type	0: This is an HDLC message. 1: This is a BOC (bit-oriented-code).
6	rx_err	In an HDLC message (rx_type=0), this bit is used to indicate an error condition. The rx_err bit is set to 0 when this sub-packet has no error. The rx_err bit is set to 1 when this sub-packet has any errors. If rx_err is set, this sub-packet ID byte is followed by another ID byte which is the ERROR ID byte (see next table). In a bit-oriented-code (rx_type=1), this bit is not significant.
5	rx_last	In an HDLC message where rx_type=0, this bit is used to indicate the last part of the message. The rx_last bit is set to 0 when this sub-packet is not the last portion of the HDLC message (more than 31 bytes). The rx_last bit is set to 1 when this sub-packet is the last portion of the HDLC message (less than 31 bytes). In a bit-oriented-code where rx_type=0, or a NOP where rx_type=1, the code is no longer detected.
4:0	size[4:0]/ count[4:0]	In an HDLC message (rx_type=0), the bits size[4:0] are used to indicate the size of the byte field to follow in this sub-packet. In a bit-oriented-code (rx_type=1), the bits count[4:0] are used to count the number of times the following BOC was received.

RX ERROR ID Byte Format:
If rx_err is set, this sub-packet ID byte is followed by another ID byte, the ERROR ID byte. The ERROR ID byte format is as follows. Note that in general, more than one of the error bits could be set.

Bit	**Name**	**Function**
7:6	Unused	None
5	sig_err	This incoming DS1 had an OOF or LOS condition (no equivalent in TX).
4	too_short	This HDLC message size is less than 4 octets.
3	rxfifo_ofl	During this HDLC message reception, the RX FIFO overflowed. (This situation is treated identically to abort reception: data is discarded until the next packet opening flag, then RX FIFO storage is re-attempted.)
2	abort_det	During this HDLC message reception, an abort code (7 or more ones) was detected.
1	aln_err	Alignment error: the received data bits were not a multiple of 8.
0	crc_err	CRC error: the received CRC-16 did not match computed CRC-16.

Register Name: **FDL_TX_RAM_CH0 .. FDL_TX_RAM_CH27 (28 DS1s) - Facility Data Link TX RAM (indirect access with undef_fdl = 0 in CPUICMD)**

Register Address: 0x8000, 0x8080, 0x8100, 0x8180, 0x8200, 0x8280, 0x8300, 0x8380, 0x8400, 0x8480, 0x8500, 0x8580, 0x8600, 0x8680, 0x8700, 0x8780, 0x8800, 0x8880, 0x8900, 0x8980, 0x8A00, 0x8A80, 0x8B00, 0x8B80, 0x8C00, 0x8C80, 0x8D00, 0x8D80 (28 Locations)

<table>
<tr><th></th><th></th><th colspan="32">Bit Function</th></tr>
<tr><th></th><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td rowspan="32">Address Location for 32 Longwords (Hex)</td><td>00</td><td colspan="16">Reserved
(Set bmask[3:2]=00 in CPUICMD register when CPU writes to this location)</td><td>ufl_err</td><td colspan="7">txfsize [6:0]</td><td>add_crc</td><td>int_mask</td><td colspan="2">contx[1:0]</td><td>int_every</td><td>Unused</td><td>tx_xpar</td><td>tx_ena</td></tr>
<tr><td>04</td><td colspan="32">Reserved</td></tr>
<tr><td>08</td><td colspan="16">Reserved</td><td colspan="16">fill_pattern[15:0]</td></tr>
<tr><td>0C</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>10</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>14</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>18</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>1C</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>20</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>24</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>28</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>2C</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>30</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>34</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>38</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>3C</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>40</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>44</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>48</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>4C</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>50</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>54</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>58</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>5C</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>60</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>64</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>68</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>6C</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>70</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>74</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>78</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td>7C</td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td><td colspan="8"></td></tr>
</table>

tx_ena: (R/W)
The tx_ena bit selects the TX FDL operation. When tx_ena is set to logical 1, the TX FDL state machine is enabled to send bit oriented or HDLC messages. When tx_ena is set to logical 0, the TX FDL state machine is disabled and initialized and then sends all 1's. Toggling tx_ena bit is a way to initialize FIFO pointers and the sub-packet parsing process.

tx_xpar: (R/W)
The tx_xpar bit selects the TX FDL as transparent mode. When tx_xpar is set to logical 1, the TX FDL state machine is enabled in transparent mode, which send sub-packets without any processing. When tx_xpar is set to logical 0, the TX FDL state machine operates using bit oriented / HDLC messages.

int_every: (R/W)
The int_every bit controls the TX FDL interrupt. When int_every is set to logical 1, the TX FDL state machine will interrupt the CPU every time an entry is removed from TX FDL FIFO (and not necessarily fully transmitted out to the line). When int_every is set to logical 0, the TX FDL state machine will interrupt only when crossing the low water mark (half full); that is, when txfsize[6:0] decreases past 64 bytes.

contx[1:0]: (R/W)
The contx[1:0] bits control the continuous repetition (cycle through) and transmission of the sub-packets that are in TX FIFO. For predictable operation, the programmer must set these bits as follows:

1. Initially, contx[1:0] = 00. Wait for txfsize to read back 0 (wait for TX FIFO to be cleared out).
2. Set contx[1:0] = 01. Write sub-packets into the FIFO.
3. Set contx[1:0] = 10. The sub-packets are continuously sent out.
4. Set contx[1:0] = 11. The machine finishes sending the current data, then txfsize is cleared.
5. Set contx[1:0] = 00. Either go to normal operation, or repeat from step 2 with new data.

int_mask: (R/W)
The int_mask bit masks the TX FDL interrupt. When int_mask is set to logical 1, the TX FDL state machine will generate an interrupt (according to int_every). When int_mask is set to logical 0, the TX FDL state machine will not generate an interrupt.

add_crc: (R/W)
The add_crc bit enables the addition of CRC-16 to an HDLC packet. When add_crc is set to logical 1, the TX FDL state machine will add CRC-16 to an outgoing HDLC packet. When add_crc is set to logical 0, the TX FDL state machine will not add CRC-16.

txfsize[6:0]: (RO)
The txfsize[6:0] bits indicate the TX FDL FIFO size in bytes. The txfsize[6:0] is decremented one byte at a time, and txfsize[6:0] is incremented by a CPU indirect write.

ufl_err: (RO, W1C)
The ufl_err bit is set to 1 when TX FDL FIFO has underflowed (sticky, write 1 to clear). The TX FDL machine processes one sub-packet at a time, after the complete sub-packet has been written in. The ufl_err bit is set if a sub-packet has tx_last = 0 (not last) and the next sub-packet has not been completely written in by the CPU before the hardware is ready to read it.

Reserved:
The reserved bits are used for TX FDL FIFO implementation. They should not be programmed.

fill_pattern[15:0]:
This is used as a 16-bit fill pattern, to be transmitted to a line that is idle (there are no BOCs or HDLC packets to send). Normally this should be set to either hex 0xFFFF or hex 0x7E7E. It can also be used to send alarms, such as YELLOW, as follows: set the fill pattern to hex 0xFF00, and use the TX FIFO for regular HDLC messages.

TX FDL FIFO: (R/W)

The TX FDL FIFO is used for storing transmit FDL data, together with a sub-packet ID byte. It can hold up to 116 bytes (29 LW x 4 bytes) per DS1 channel. The FDL (4 Kb/s) occupies the F bit of every odd frame of an extended superframe (ESF). The CPU can write up to 4 bytes of TX FDL FIFO data with undef_fdl = 1 in the CPUICMD register (see below).

Register Name: **FDL_TX_RAM_CH0 .. FDL_TX_RAM_CH27 (28 DS1's)**
- Facility Data Link TX RAM
(indirect access with undef_fdl = 1 in CPUICMD)
Register Address: **0x8000, 0x8080, 0x8100, 0x8180, 0x8200, 0x8280, 0x8300, 0x8380, 0x8400, 0x8480, 0x8500, 0x8580, 0x8600, 0x8680, 0x8700, 0x8780, 0x8800, 0x8880, 0x8900, 0x8980, 0x8A00, 0x8A80, 0x8B00, 0x8B80, 0x8C00, 0x8C80, 0x8D00, 0x8D80**
(28 Locations)

It is legal for the CPU to perform an indirect write only with undef_fdl = 1 for addresses of the form [FDL_TX_RAM_BASE + (n x 128)], where n = 0 .. 27. This type of indirect write is used as a way of inserting TX FDL FIFO bytes, while maintaining pointers consistently.

		Bit Function																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
One LW addr.	00	First_byte								Second_byte								Third_Byte								Fourth_byte							

First_byte, Second_byte, Third_byte, Fourth_byte: (WO)
When a CPU indirect write access is performed with undef_fdl = 1 in the CPUICMD register to this location, the written longword contains up to 4 bytes of TX FDL FIFO data. Each time a write is performed, txfsize[6:0] is incremented by the number of bytes handed off by the CPU section. If there are less than 4 bytes to give, the returned byte enables the bits bmask[3:0] in the CPUICMD register; these bits are used to indicate which bytes are valid.

The first octet of a CPU write will always be a sub-packet ID byte. The ID byte contains a bit field that describes whether this sub-packet is a BOC type or HDLC type, and another bit field that specifies the number of bytes to follow in this sub-packet. This allows the VSC9670 to determine the location of the next sub-packet ID byte.

TX Sub-packet ID Byte Format:

Bit	Name	Function
7	tx_type	0: This is a HDLC message. 1: This is a BOC (bit-oriented-code).
6	tx_ins_cerr/ tx_xparent	In an HDLC message (tx_type=0), this bit is used to force CRC-16 error insertion. The tx_ins_cerr bit is set to 0 when this sub-packet has no CRC-16 error insertion. The tx_ins_cerr bit is set to 1 when this sub-packet has a forced CRC-16 error insertion. In a bit-oriented-code (tx_type=1), this bit is used to select the BOC function. The tx_xparent bit is set to 0 when this BOC sends an octet with the initial 0xFF. The tx_xparent bit is set to 1 when this BOC sends an octet without the initial 0xFF.
5	tx_last	In an HDLC message (tx_type=0), this bit is used to indicate the last part of the message. The tx_last bit is set to 0 when this sub-packet is not the last portion of the HDLC message (more than 31 bytes). The tx_last bit is set to 1 when this sub-packet is the last portion of the HDLC message (less than 31 bytes). In a bit-oriented-code (tx_type=1), this bit is used to control the BOC operation. The tx_last bit is set to 0 when the next BOC sub-packet item is to be sent with 16 bits of fill pattern. The tx_last bit is set to 1 when the next BOC sub-packet item is to be sent without 16 bits of fill pattern.
4:0	size[4:0]/ count[4:0]	In an HDLC message (tx_type=0), the bits size[4:0] are used to indicate the size of the byte field to follow in this sub-packet. In a bit-oriented-code (tx_type=1), the bits count[4:0] are used to count the number of times the following BOC must be sent. (Note: Setting this field to 0x00 is illegal.)

The tx_last feature in BOC mode allows a BOC to be sent more than 31 times consecutively. Note that for much longer durations of BOC sending, it is recommended to modify the fill_pattern[15:0] as described earlier.

The tx_xparent feature in BOC mode allows sending a precisely measured number of flags (hex 7E's) in between two consecutive HDLC messages. By default, in between two sub-packets (the last portion of an HDLC message and the initial portion of another HDLC message), the TX machine will send out a fill_pattern[15:0] once. If fill_pattern = 0x7E7E, that means there is a default of 2 flags between HDLC packets.

If more flags are desired between HDLC packets, insert a tx_xparent type of BOC sub-packet with BOC = 0x7E between the end-of-message sub-packet and beginning-of-message sub-packet.

3.6 Signaling Machine, RAM Bits

3.6.1 Per Chip Registers

Register Name: SIG_CNTTHR_OVRIDE - RX Signaling FSM Threshold Override
Register Address: 0x1C

Bit	Type	Function	Reset
7		Unused	X
6	R/W	sig_cntthr_ena_delay	0
5:4	R/W	sig_cntthr_delay_mode[1:0]	0x0
3		Unused	X
2	R/W	sig_cntthr_ena_debounce	0
1:0	R/W	sig_cntthr_debounce_mode[1:0]	0x0

sig_cntthr_ena_delay:
The sig_cntthr_ena_delay bit is active when set to 1 and inactive when set to 0. Valid ONLY in ESF Mode when signalling state is DS0AIS (Unicode). When sig_cntthr_ena_delay is active use sig_cntthr_delay_mode[1:0] value as delay count.

sig_cntthr_delay_mode[1:0]:
These bits are used to count delay value when sig_cntthr_ena_delay bit is active.

sig_cntthr_delay_mode[1:0]	**Value for delay count**
00	2 ESF Frame Delay
01	3 ESF Frame Delay
10	4 ESF Frame Delay
11	5 ESF Frame Delay

sig_cntthr_ena_debounce:
The sig_cntthr_ena_debounce bit is active when set to 1 and inactive when set to 0. Valid ONLY in ESF Mode when signalling state is DS0AIS (Unicode). When sig_cntthr_ena_debounce is active use sig_cntthr_debounce_mode[1:0] value for CNT2 threshold, ignore the Per-DS0 Control[3:2] CNT2 threshold value.

sig_cntthr_debounce_mode[1:0]:
These bits are used to count delay value when sig_cntthr_ena_debounce bit is active.

sig_cntthr_debounce_mode[1:0]	**Value for delay count**
00	2 ESF Frame Delay
01	3 ESF Frame Delay
10	4 ESF Frame Delay
11	5 ESF Frame Delay

Register Name: SIG_FIFO_STAT - RX Signaling FIFO Status/Control
Register Address: 0x1D

Bit	Type	Function	Reset
7	R/W	sig_fifo_rptr_incr	0
6:4		Unused	X
3	RO	sig_fifo_err	0
2	RO	sig_fifo_empty	0
1	R/W	sig_fifo_rst	0
0	R/W	sig_fifo_ena	0

sig_fifo_rptr_incr:
The sig_fifo_rptr_incr bit will cause the signaling FIFO read address to increment upon completion of the current access. The sig_fifo_rptr_incr bit is written to 1 by the CPU. Automatically cleared to 0 by the hardware.

sig_fifo_err:
The sig_fifo_err bit indicates signaling FIFO full (saturated). When this bit is set to logical 1, the signaling FIFO is full. When this bit is set to logical 0, the signaling FIFO is not full.

sig_fifo_empty:
The sig_fifo_empty bit indicates signaling FIFO empty status. When this bit is set to logical 1, the signaling FIFO is empty. When this bit is set to logical 0, the signaling FIFO is not empty.

sig_fifo_rst:
The sig_fifo_rst bit will reset the signaling FIFO read/write address and control status bits. The sig_fifo_rst bit is written to ‘1’ by the CPU. Automatically cleared to ‘0’ by the hardware.

sig_fifo_ena:
The sig_fifo_ena bit is used to enable signaling FIFO operation. When this bit is set to logical 1, the signaling FIFO operation is enabled. When this bit is set to logical 0, the signaling FIFO operation is disabled.

Register Name: SIG_FIFO_1 - RX Signaling FIFO Data High Byte
Register Address: 0x1E

Bit	Type	Function	Reset
7	RO	sig_fifo_valid	0
6	RO	sig_fifo_not_empty	0
5:1	RO	sig_fifo_ds1_id[4:0]	0x0
0	RO	sig_fifo_ds0_id[4]	0

sig_fifo_valid:
The sig_fifo_valid bit indicates that SIG_FIFO_1 and SIG_FIFO_0 have valid data. When this bit is set to logical 1, the signaling FIFO has valid data. When this bit is set to logical 0, the signaling FIFO has no valid data.

sig_fifo_not_empty:
The sig_fifo_not_empty bit indicates the Signaling FIFO has at least one valid entry. When this bit is set to logical 1, the signaling FIFO has valid entry. When this bit is set to logical 0, the signaling FIFO has no valid entry.

sig_fifo_ds1_id[4:0]:
The sig_fifo_ds1_id[4:0] bits indicate the signaling FIFO data belongs to which DS1.

sig_fifo_ds0_id[4]:
See sig_fifo_ds0_id[3:0] description in SIG_FIFO_0 register below.

Register Name: SIG_FIFO_0 - RX Signaling FIFO Data Low Byte
Register Address: 0x1F

Bit	Type	Function	Reset
7:4	RO	sig_fifo_ds0_id[3:0]	0x0
3:0	RO	sig_fifo_bits[3:0]	0x0

sig_fifo_ds0_id[3:0]:
The sig_fifo_ds0_id[3:0] bits concatenated with the sig_fifo_ds0_id[4] bit in SIG_FIFO_1 register indicate the signaling FIFO data belongs to which DS0.

sig_fifo_bits[3:0]:
The sig_fifo_bits[3:0] bits indicate the value of the robbed bits. The sig_fifo_bits[3:0] = ABCD in ESF mode or sig_fifo_bits[3:0] = ABxx in SF mode.

3.6.2 Signaling RAM Registers

The signaling RAM is 32 bits wide; 16 longwords (64 bytes) are used per DS1 channel, for a total size of 448 (16 x 28 channels) longwords for each of RX and TX. If the RX or TX signaling RAM is located at absolute address SIG_RX_RAM_BASE or SIG_TX_RAM_BASE in the CPU's memory space, the RX space for a channel n (from 0 to 27) has a base address of [SIG_RX_RAM_BASE + (n x 64)] and the TX space for a channel n (from 0 to 27) has a base address of [SIG_TX_RAM_BASE + (n x 64)].

As described earlier,

SIG_RX_RAM_BASE = 0xA000 (this is a base address for RX channel 0).
SIG_TX_RAM_BASE = 0xA800 (this is a base address for TX channel 0).

During CPU indirect write access to this SIG_RX_RAM, the following table is valid for writes to locations 00, 04, 08, 0C, 10, 14, 18, 1C, 20, 24, 28, 2C of each channel (x = don't care).

bmask[3:0] (in CPUICMD bits [7:4])	**SIG_RX_RAM bits are written**
x0x0	No writes occurs
x0x1	Bits [03:00]
x1x0	Bits [19:16]
x1x1	Bits [19:16], [03:00]

During CPU indirect write access to this SIG_RX_RAM, the following table is valid for writes to location 3C (sig_int[23:0]) of each channel (x = don't care).

bmask[3:0] (in CPUICMD bits [7:4])	**SIG_RX_RAM bits are written**
x000	No writes occurs
x001	Bits [07:00] write 1 to clear
x011	Bits [15:08], [07:00] write 1 to clear
x111	Bits [23:16], [15:08], [07:00] write 1 to clear

Note: For these bmask[2:0] enable bits, each bit set to one indicates a byte write enable; zero indicates a byte write disable. Valid for byte enable combination up to 7 different byte writes.

During CPU indirect write access to this SIG_TX_RAM, the following table is valid for writes to locations 00, 04, 08, 0C, 10, 14, 18, 1C, 20, 24, 28, 2C of each channel (x = don't care).

bmask[3:0] (in CPUICMD bits [7:4])	**SIG_TX_RAM bits are written**
x0x0	No writes occurs
x0x1	Bits [07:00]
x1x0	Bits [23:16]
x1x1	Bits [23:16], [07:00]

Register Name: **SIG_RX_RAM_CH0 .. SIG_RX_RAM_CH27 (28 DS1's)**
- RX Signaling RAM
Register Address: **0xA000, 0xA040, 0xA080, 0xA0C0, 0xA100, 0xA140, 0xA180, 0xA1C0, 0xA200, 0xA240, 0xA280, 0xA2C0, 0xA300, 0xA340, 0xA380, 0xA3C0, 0xA400, 0xA440, 0xA480, 0xA4C0, 0xA500, 0xA540, 0xA580, 0xA5C0, 0xA600, 0xA640, 0xA680, 0xA6C0 (28 Locations)**

<table>
<tr><th rowspan="2">Address Location for 16 Longwords (Hex)</th><th colspan="32">Bit Function</th></tr>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td>00</td><td colspan="4">RX FIFO (DS0_01)</td><td colspan="4">RX FIFO (DS0_01)</td><td colspan="4">RX FIFO (DS0_01)</td><td colspan="4">Control[3:0] (DS0_01)</td><td colspan="4">RX FIFO (DS0_00)</td><td colspan="4">RX FIFO (DS0_00)</td><td colspan="4">RX FIFO (DS0_00)</td><td colspan="4">Control[3:0] (DS0_00)</td></tr>
<tr><td>04</td><td colspan="4">RX FIFO (DS0_03)</td><td colspan="4">RX FIFO (DS0_03)</td><td colspan="4">RX FIFO (DS0_03)</td><td colspan="4">Control[3:0] (DS0_03)</td><td colspan="4">RX FIFO (DS0_02)</td><td colspan="4">RX FIFO (DS0_02)</td><td colspan="4">RX FIFO (DS0_02)</td><td colspan="4">Control[3:0] (DS0_02)</td></tr>
<tr><td>08</td><td colspan="4">RX FIFO (DS0_05)</td><td colspan="4">RX FIFO (DS0_05)</td><td colspan="4">RX FIFO (DS0_05)</td><td colspan="4">Control[3:0] (DS0_05)</td><td colspan="4">RX FIFO (DS0_04)</td><td colspan="4">RX FIFO (DS0_04)</td><td colspan="4">RX FIFO (DS0_04)</td><td colspan="4">Control[3:0] (DS0_04)</td></tr>
<tr><td>0C</td><td colspan="4">RX FIFO (DS0_07)</td><td colspan="4">RX FIFO (DS0_07)</td><td colspan="4">RX FIFO (DS0_07)</td><td colspan="4">Control[3:0] (DS0_07)</td><td colspan="4">RX FIFO (DS0_06)</td><td colspan="4">RX FIFO (DS0_06)</td><td colspan="4">RX FIFO (DS0_06)</td><td colspan="4">Control[3:0] (DS0_06)</td></tr>
<tr><td>10</td><td colspan="4">RX FIFO (DS0_09)</td><td colspan="4">RX FIFO (DS0_09)</td><td colspan="4">RX FIFO (DS0_09)</td><td colspan="4">Control[3:0] (DS0_09)</td><td colspan="4">RX FIFO (DS0_08)</td><td colspan="4">RX FIFO (DS0_08)</td><td colspan="4">RX FIFO (DS0_08)</td><td colspan="4">Control[3:0] (DS0_08)</td></tr>
<tr><td>14</td><td colspan="4">RX FIFO (DS0_11)</td><td colspan="4">RX FIFO (DS0_11)</td><td colspan="4">RX FIFO (DS0_11)</td><td colspan="4">Control[3:0] (DS0_11)</td><td colspan="4">RX FIFO (DS0_10)</td><td colspan="4">RX FIFO (DS0_10)</td><td colspan="4">RX FIFO (DS0_10)</td><td colspan="4">Control[3:0] (DS0_10)</td></tr>
<tr><td>18</td><td colspan="4">RX FIFO (DS0_13)</td><td colspan="4">RX FIFO (DS0_13)</td><td colspan="4">RX FIFO (DS0_13)</td><td colspan="4">Control[3:0] (DS0_13)</td><td colspan="4">RX FIFO (DS0_12)</td><td colspan="4">RX FIFO (DS0_12)</td><td colspan="4">RX FIFO (DS0_12)</td><td colspan="4">Control[3:0] (DS0_12)</td></tr>
<tr><td>1C</td><td colspan="4">RX FIFO (DS0_15)</td><td colspan="4">RX FIFO (DS0_15)</td><td colspan="4">RX FIFO (DS0_15)</td><td colspan="4">Control[3:0] (DS0_15)</td><td colspan="4">RX FIFO (DS0_14)</td><td colspan="4">RX FIFO (DS0_14)</td><td colspan="4">RX FIFO (DS0_14)</td><td colspan="4">Control[3:0] (DS0_14)</td></tr>
<tr><td>20</td><td colspan="4">RX FIFO (DS0_17)</td><td colspan="4">RX FIFO (DS0_17)</td><td colspan="4">RX FIFO (DS0_17)</td><td colspan="4">Control[3:0] (DS0_17)</td><td colspan="4">RX FIFO (DS0_16)</td><td colspan="4">RX FIFO (DS0_16)</td><td colspan="4">RX FIFO (DS0_16)</td><td colspan="4">Control[3:0] (DS0_16)</td></tr>
<tr><td>24</td><td colspan="4">RX FIFO (DS0_19)</td><td colspan="4">RX FIFO (DS0_19)</td><td colspan="4">RX FIFO (DS0_19)</td><td colspan="4">Control[3:0] (DS0_19)</td><td colspan="4">RX FIFO (DS0_18)</td><td colspan="4">RX FIFO (DS0_18)</td><td colspan="4">RX FIFO (DS0_18)</td><td colspan="4">Control[3:0] (DS0_18)</td></tr>
<tr><td>28</td><td colspan="4">RX FIFO (DS0_21)</td><td colspan="4">RX FIFO (DS0_21)</td><td colspan="4">RX FIFO (DS0_21)</td><td colspan="4">Control[3:0] (DS0_21)</td><td colspan="4">RX FIFO (DS0_20)</td><td colspan="4">RX FIFO (DS0_20)</td><td colspan="4">RX FIFO (DS0_20)</td><td colspan="4">Control[3:0] (DS0_20)</td></tr>
<tr><td>2C</td><td colspan="4">RX FIFO (DS0_23)</td><td colspan="4">RX FIFO (DS0_23)</td><td colspan="4">RX FIFO (DS0_23)</td><td colspan="4">Control[3:0] (DS0_23)</td><td colspan="4">RX FIFO (DS0_22)</td><td colspan="4">RX FIFO (DS0_22)</td><td colspan="4">RX FIFO (DS0_22)</td><td colspan="4">Control[3:0] (DS0_22)</td></tr>
<tr><td>30</td><td colspan="32">Unused</td></tr>
<tr><td>34</td><td colspan="32">Unused</td></tr>
<tr><td>38</td><td colspan="32">Unused</td></tr>
<tr><td>3C</td><td>sig_frz</td><td colspan="7">Unused</td><td colspan="24">sig_int[23:0]</td></tr>
</table>

For the RX direction, per DS1 channel, there are 24 DS0s for which robbed bit signaling needs to be extracted, buffered, and presented to the CPU. According to DS0 standards, there are two options: signaling bit delay mode, and signaling bit filter mode. The robbed bit signaling occupies the LSB bit of every timeslot in the 6th (A bit) and 12th (B bit) frames of a superframe (SF), or in the 6th (A bit), 12th (B bit), 18th (C bit) and 24th (D bit) frames of an extended superframe (ESF).

Control[3:0]:
The Control[3:0] bits are used to control the RX signaling operation.

The Control[0] bit selects mode operation. If this bit is set to 0, Delay mode is selected. If set to 1, Filter (de-bounce) mode is selected.

The Control[1] bit is used to control a CPU interrupt when the RX signaling nibble changes. If this bit is set to 0, CPU interrupt is disabled; if set to 1, CPU interrupt is enabled.

The Control[3:2] bits are used by the CPU to program the threshold in CNT2 for Filter (de-bounce) mode. In Delay Mode, when the bit sig_cntthr_ena_delay = 1 in SIG_CNTTHR_OVRIDE register, the Control[3:2] bits are used by the HW to perform a similar function.

Control[3:2] = CNT2 = 00, the filtering threshold is 2.
Control[3:2] = CNT2 = 01, the filtering threshold is 3.
Control[3:2] = CNT2 = 10, the filtering threshold is 4.
Control[3:2] = CNT2 = 11, the filtering threshold is 5.

RX FIFO:
The 12 bits of 3-stage RX FIFO nibbles are used to store received AB or ABCD robbed signaling bits per DS0 (see Section 2.10 Signaling State Machine for more details).

sig_int[23:0]:
Per-DS0 interrupts, 0 = inactive, 1 = active. Write 1 to clear.

sig_frz:
Per-DS1 signaling frozen indicator. 0 = not frozen, 1 = frozen. Write 1 to clear.

Register Name: **SIG_TX_RAM_CH0 .. SIG_TX_RAM_CH27 (28 DS1s) - TX Signaling RAM**

Register Address: **0xA800, 0xA840, 0xA880, 0xA8C0, 0xA900, 0xA940, 0xA980, 0xA9C0, 0xAA00, 0xAA40, 0xAA80, 0xAAC0, 0xAB00, 0xAB40, 0xAB80, 0xABC0, 0xAC00, 0xAC40, 0xAC80, 0xACC0, 0xAD00, 0xAD40, 0xAD80, 0xADC0, 0xAE00, 0xAE40, 0xAE80, 0xAEC0 (28 Locations)**

Address Location for 16 Longwords (Hex)	Bit Function							
	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
00	Unused	TX_ABCD (DS0_01)	TX_ABCD (DS0_01)	Control[3:0] (DS0_01)	Unused	TX_ABCD (DS0_00)	TX_ABCD (DS0_00)	Control[3:0] (DS0_00)
04	Unused	TX_ABCD (DS0_03)	TX_ABCD (DS0_03)	Control[3:0] (DS0_03)	Unused	TX_ABCD (DS0_02)	TX_ABCD (DS0_02)	Control[3:0] (DS0_02)
08	Unused	TX_ABCD (DS0_05)	TX_ABCD (DS0_05)	Control[3:0] (DS0_05)	Unused	TX_ABCD (DS0_04)	TX_ABCD (DS0_04)	Control[3:0] (DS0_04)
0C	Unused	TX_ABCD (DS0_07)	TX_ABCD (DS0_07)	Control[3:0] (DS0_07)	Unused	TX_ABCD (DS0_06)	TX_ABCD (DS0_06)	Control[3:0] (DS0_06)
10	Unused	TX_ABCD (DS0_09)	TX_ABCD (DS0_09)	Control[3:0] (DS0_09)	Unused	TX_ABCD (DS0_08)	TX_ABCD (DS0_08)	Control[3:0] (DS0_08)
14	Unused	TX_ABCD (DS0_11)	TX_ABCD (DS0_11)	Control[3:0] (DS0_11)	Unused	TX_ABCD (DS0_10)	TX_ABCD (DS0_10)	Control[3:0] (DS0_10)
18	Unused	TX_ABCD (DS0_13)	TX_ABCD (DS0_13)	Control[3:0] (DS0_13)	Unused	TX_ABCD (DS0_12)	TX_ABCD (DS0_12)	Control[3:0] (DS0_12)
1C	Unused	TX_ABCD (DS0_15)	RX FIFO (DS0_15)	Control[3:0] (DS0_15)	Unused	TX_ABCD (DS0_14)	TX_ABCD (DS0_14)	Control[3:0] (DS0_14)
20	Unused	TX_ABCD (DS0_17)	TX_ABCD (DS0_17)	Control[3:0] (DS0_17)	Unused	TX_ABCD (DS0_16)	TX_ABCD (DS0_16)	Control[3:0] (DS0_16)
24	Unused	TX_ABCD (DS0_19)	TX_ABCD (DS0_19)	Control[3:0] (DS0_19)	Unused	TX_ABCD (DS0_18)	TX_ABCD (DS0_18)	Control[3:0] (DS0_18)
28	Unused	TX_ABCD (DS0_21)	TX_ABCD (DS0_21)	Control[3:0] (DS0_21)	Unused	TX_ABCD (DS0_20)	TX_ABCD (DS0_20)	Control[3:0] (DS0_20)
2C	Unused	TX_ABCD (DS0_23)	TX_ABCD (DS0_23)	Control[3:0] (DS0_23)	Unused	TX_ABCD (DS0_22)	TX_ABCD (DS0_22)	Control[3:0] (DS0_22)
30	Unused							
34	Unused							
38	Unused							
3C	Unused							

For the TX direction, per DS1 channel, there are 24 DS0s, for which robbed bit signaling needs to be buffered and written by the CPU. The robbed bit signaling occupies the LSB bit of every timeslot in the 6th (A bit) and 12th (B bit) frames of a superframe (SF), or in the 6th (A bit), 12th (B bit), 18th (C bit) and 24th (D bit) frames of an extended superframe (ESF).

Control[3:0]:
The Control[3:0] bits are used to control the TX signaling operation.

The Control[0] bit 0 is used to enable the TX signaling operation. If this bit is set to 0, the TX will not insert robbed bit signaling. If this bit is set to 1, the TX will insert signaling into appropriate bits.

The Control[1] bit 1 is used to control the CPU write operation to the TX_ABCD nibble location. If the CPU sets this bit to 0, then new AB or ABCD signaling bits are inserted into the first nibble, TX_ABCD [7:4] or TX_ABCD [23:20]. If the CPU reads this bit as 1, the just-inserted AB or ABCD signaling bits in the first nibble are copied into the second nibble, TX_ABCD [11:8] or TX_ABCD [27:24].

The Control[3:2] bits 3 and 2 are not used.

TX_ABCD:
These 8 bits of 2-stage TX_ABCD nibbles are used to store transmit AB or ABCD robbed signaling bits per DS0. The first nibble, TX_ABCD [7:4] or TX_ABCD [23:20], is written by the CPU. The second nibble, TX_ABCD [11:8] or TX_ABCD [27:24], is the active copy sent on the outgoing TX line.

3.7 Slip Buffer, RAM Bits

The Slip Buffer RAM is 32 bits wide; 16 longwords (64 bytes) are used per DS1 channel to store two frames of DS1 data and control bits, for a total size of 448 (16 x 28 channels) longwords for each of RX and TX. If the RX or TX Slip Buffer RAM is located at absolute address SLIP_RX_RAM_BASE or SLIP_TX_RAM_BASE in the CPU's memory space, the RX space for a channel n (from 0 to 27) has a base address of [SLIP_RX_RAM_BASE + (n x 64)] and the TX space for a channel n (from 0 to 27) has a base address of [SLIP_TX_RAM_BASE + (n x 64)]. As described earlier,

SLIP_RX_RAM_BASE = 0xB000 (this is a base address for RX channel 0).
SLIP_TX_RAM_BASE = 0xC000 (this is a base address for TX channel 0).

During CPU indirect write access to this SLIP_RX_RAM or SLIP_TX_RAM, the following table is valid for writes to locations 00, 04, 08, 0C, 10, 14, 20, 24, 28, 2C, 30, 34, 3C of each channel.

bmask[3:0] (in CPUICMD bits [7:4])	**SLIP_RX/TX_RAM bits are written**
0000	No writes occurs
0001	Bits [07:00]
0011	Bits [15:08], [07:00]
0111	Bits [23:16], [15:08], [07:00]
1111	Bits [31:24], [23:16], [15:08], [07:00]

Note: For these bmask[3:0] enable bits, each bit set to '1' indicates a byte write enable; '0' indicates a byte write disable. Valid for byte enable combination up to 15 different byte writes.

Register Name: **SLIP_RX_RAM_CH0 .. SLIP_RX_RAM_CH27 (28 DS1s) - RX Slip Buffer RAM**

Register Address: **0xB000, 0xB040, 0xB080, 0xB0C0, 0xB100, 0xB140, 0xB180, 0xB1C0, 0XB200, 0xB240, 0xB280, 0xB2C0, 0xB300, 0xB340, 0xB380, 0xB3C0, 0xB400, 0xB440, 0xB480, 0xB4C0, 0xB500, 0xB540, 0xB580, 0xB5C0, 0xB600, 0xB640, 0xB680, 0xB6C0 (28 Locations)**

Address Location for 16 Longwords (Hex)	Bit Function			
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
00	ds0_00[7:0]	ds0_01[7:0]	ds0_02[7:0]	ds0_03[7:0]
04	ds0_04[7:0]	ds0_05[7:0]	ds0_06[7:0]	ds0_07[7:0]
08	ds0_08[7:0]	ds0_09[7:0]	ds0_10[7:0]	ds0_11[7:0]
0C	ds0_12[7:0]	ds0_13[7:0]	ds0_14[7:0]	ds0_15[7:0]
10	ds0_16[7:0]	ds0_17[7:0]	ds0_18[7:0]	ds0_19[7:0]
14	ds0_20[7:0]	ds0_21[7:0]	ds0_22[7:0]	ds0_23[7:0]
18	Unused			
1C	Unused			
20	ds0_00[7:0]	ds0_01[7:0]	ds0_02[7:0]	ds0_03[7:0]
24	ds0_04[7:0]	ds0_05[7:0]	ds0_06[7:0]	ds0_07[7:0]
28	ds0_08[7:0]	ds0_09[7:0]	ds0_10[7:0]	ds0_11[7:0]
2C	ds0_12[7:0]	ds0_13[7:0]	ds0_14[7:0]	ds0_15[7:0]
30	ds0_16[7:0]	ds0_17[7:0]	ds0_18[7:0]	ds0_19[7:0]
34	ds0_20[7:0]	ds0_21[7:0]	ds0_22[7:0]	ds0_23[7:0]
38	Unused			
3C	Unused	per_ds0_force[23:0]		

These RAM locations are R/W and contents are unknown after a power reset. Therefore, the user must program default values for control bits.

Shown above are the 16 longwords (64 bytes) assigned to a per-DS1 RX slip buffer. For each RX, there is a slip buffer read pointer or write pointer that is common to all 28 DS1's and is driven by the backplane clock. For the RX slip buffer, the RX FSM hands off an octet of data, together with a 0 to 192 count value, which identifies the DS1 frame position of the first bit in the octet.

ds0_00[7:0] .. ds0_23[7:0]:
These byte locations store two RX frames of DS1 data (frame n, frame n+1). Frame n's address starts at 0x00, bit[31:24] and goes through 0x14, bit[07:00] for one frame's worth of 24 DS0 octets. Frame n+1's address starts at 0x20, bit[31:24] and goes through 0x34, bit[07:00] for another one frame's worth of 24 DS0 octets (frame bit position not stored).

per_ds0_force[23:0]:
The per_ds0_force[23:0] bits are used to allow the CPU to write a byte value into the appropriate DS0 location, ds0_xx[7:0] (where xx indicates the DS0 number), in both frames. For instance, if the bit per_ds0_force[0] is set to 1, then the ds0_00[7:0] byte value will be written from the CPU and sent out constantly to the system RX side on the desired DS0 timeslot.

Register Name: **SLIP_TX_RAM_CH0 .. SLIP_TX_RAM_CH27 (28 DS1s) - TX Slip Buffer RAM**

Register Address: **0xC000, 0xC040, 0xC080, 0xC0C0, 0xC100, 0xC140, 0xC180, 0xC1C0, 0xC200, 0xC240, 0xC280, 0xC2C0, 0xC300, 0xC340, 0xC380, 0xC3C0, 0xC400, 0xC440, 0xC480, 0xC4C0, 0xC500, 0xC540, 0xC580, 0xC5C0, 0xC600, 0xC640, 0xC680, 0xC6C0 (28 Locations)**

Address Location for 16 Longwords (Hex)	Bit Function			
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
00	ds0_00[7:0]	ds0_01[7:0]	ds0_02[7:0]	ds0_03[7:0]
04	ds0_04[7:0]	ds0_05[7:0]	ds0_06[7:0]	ds0_07[7:0]
08	ds0_08[7:0]	ds0_09[7:0]	ds0_10[7:0]	ds0_11[7:0]
0C	ds0_12[7:0]	ds0_13[7:0]	ds0_14[7:0]	ds0_15[7:0]
10	ds0_16[7:0]	ds0_17[7:0]	ds0_18[7:0]	ds0_19[7:0]
14	ds0_20[7:0]	ds0_21[7:0]	ds0_22[7:0]	ds0_23[7:0]
18	Unused			
1C	Unused			
20	ds0_00[7:0]	ds0_01[7:0]	ds0_02[7:0]	ds0_03[7:0]
24	ds0_04[7:0]	ds0_05[7:0]	ds0_06[7:0]	ds0_07[7:0]
28	ds0_08[7:0]	ds0_09[7:0]	ds0_10[7:0]	ds0_11[7:0]
2C	ds0_12[7:0]	ds0_13[7:0]	ds0_14[7:0]	ds0_15[7:0]
30	ds0_16[7:0]	ds0_17[7:0]	ds0_18[7:0]	ds0_19[7:0]
34	ds0_20[7:0]	ds0_21[7:0]	ds0_22[7:0]	ds0_23[7:0]
38	Unused			
3C	Unused	per_ds0_force[23:0]		

These RAM locations are R/W, and their contents are unknown after a power reset. Therefore, the user must program default values for control bits.

Shown above are the 16 longwords (64 bytes) assigned to a per-DS1 TX slip buffer. For each TX, there is a slip buffer read pointer or write pointer that is common to all 28 DS1s and is driven by the backplane clock. For the TX slip buffer, the TX FSM hands off an octet of data, together with a 0 to 192 count value, which identifies the DS1 frame position of the first bit in the octet.

ds0_00[7:0] .. ds0_23[7:0]:
These byte locations store two TX frames of DS1 data (frame n, frame n+1). The address of frame n starts at 0x00, bit[31:24] and goes through 0x14, bit[07:00] for one frame's worth of 24 DS0 octets. Frame n+1's address starts at 0x20, bit[31:24] and goes through 0x34, bit[07:00] for another one frame's worth of 24 DS0 octets (frame bit position not stored).

per_ds0_force[23:0]:
The per_ds0_force[23:0] bits are used to allow the CPU to write a byte value into the appropriate DS0 location ds0_xx[7:0] (where xx indicates the DS0 number), in both frames. For instance, if the bit per_ds0_force[0] is set to 1, then the ds0_00[7:0] byte value from the TX system side data will be overwritten by the CPU in specific DS0 locations before sending it out to the line side.

3.8 Loopback, Registers and RAM

3.8.1 Per Chip Registers

Register Name: LBENAMISC - Loopback Enable Miscellaneous Control
Register Address: 0x30

Bit	Type	Function	Reset
7:2		Unused	X
1	R/W	global_lbena	0
0	R/W	per_ds0_ena	0

global_lbena, per_ds0_ena:
Both global_lbena and per_ds0_ena bits allow software to enable or disable per-DS0 network loopback in LBACK_RAM_CH[0:27] registers, as shown in the following table.

global_lbena, per_ds0_ena	Function
00	Disable all per-DS0 network loopback
01	Undefined
10	Undefined
11	Enable all per-DS0 network loopback

Register Name: LBENAL0, LBENAL1, LBENAL2, LBENAL3
- Local Loopback Enable Control
Register Address: 0x34, 0x35, 0x36, 0x37

Register Name (Address)	Bit Function							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LBENAL0 (0x34)	loc_lb_ena_07	loc_lb_ena_06	loc_lb_ena_05	loc_lb_ena_04	loc_lb_ena_03	loc_lb_ena_02	loc_lb_ena_01	loc_lb_ena_00
LBENAL1 (0x35)	loc_lb_ena_15	loc_lb_ena_14	loc_lb_ena_13	loc_lb_ena_12	loc_lb_ena_11	loc_lb_ena_10	loc_lb_ena_09	loc_lb_ena_08
LBENAL2 (0x36)	loc_lb_ena_23	loc_lb_ena_22	loc_lb_ena_21	loc_lb_ena_20	loc_lb_ena_19	loc_lb_ena_18	loc_lb_ena_17	loc_lb_ena_16
LBENAL3 (0x37)	Unused	Unused	Unused	Unused	loc_lb_ena_27	loc_lb_ena_26	loc_lb_ena_25	loc_lb_ena_24

Each of these register bits is used to enable per-DS1 local loopback.

loc_lb_ena_[27:00]:
Each of these bits enables one of the 28 DS1s local loopback. If this bit is set to ‘1’, this DS1’s local loopback is enabled. If this bit is set to ‘0’, this DS1’s local loopback is disabled. All of these registers can be read/write, and the reset value is 0x00.

Register Name: LBENAN0, LBENAN1, LBENAN2, LBENAN3
- Network Loopback Enable Control
Register Address: 0x38, 0x39, 0x3A, 0x3B

Register Name (Address)	Bit Function							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LBENAN0 (0x38)	net_lb_ena_07	net_lb_ena_06	net_lb_ena_05	net_lb_ena_04	net_lb_ena_03	net_lb_ena_02	net_lb_ena_01	net_lb_ena_00
LBENAN1 (0x39)	net_lb_ena_15	net_lb_ena_14	net_lb_ena_13	net_lb_ena_12	net_lb_ena_11	net_lb_ena_10	net_lb_ena_09	net_lb_ena_08
LBENAN2 (0x3A)	net_lb_ena_23	net_lb_ena_22	net_lb_ena_21	net_lb_ena_20	net_lb_ena_19	net_lb_ena_18	net_lb_ena_17	net_lb_ena_16
LBENAN3 (0x3B)	Unused	Unused	Unused	Unused	net_lb_ena_27	net_lb_ena_26	net_lb_ena_25	net_lb_ena_24

Each of these register bits is used to enable DS1 compliant network loopback.

net_lb_ena_[27:00]:
Each of these net_lb_ena_[27:00] bits enables one of the 28 DS1's network loopback. If this bit is set to '1', this DS1's network loopback is enabled. If this bit is set to '0', this DS1's network loopback is disabled. All of these registers can be read/write, and the reset value is 0x00.

Register Name: LBENAISON0, LBENAISON1, LBENAISON2, LBENAISON3
- Local Loopback Enable & AIS ON Control
Register Address: 0x3C, 0x3D, 0x3E, 0x3F

Register Name (Address)	Bit Function							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LBENAISON0(0x3C)	lben_ais_on _07	lben_ais_on_06	lben_ais_on_05	lben_ais_on_04	lben_ais_on_03	lben_ais_on_02	lben_ais_on_01	lben_ais_on_00
LBENAISON1(0x3D)	lben_ais_on_15	lben_ais_on_14	lben_ais_on_13	lben_ais_on_12	lben_ais_on_11	lben_ais_on_10	lben_ais_on_09	lben_ais_on_08
LBENAISON2(0x3E)	lben_ais ena_23	lben_ais_on_22	lben_ais_on_21	lben_ais_on_20	lben_ais_on_19	lben_ais_on_18	lben_ais_on_17	lben_ais_on_16
LBENAISON3(0x3F)	Unused	Unused	Unused	Unused	lben_ais_on_27	lben_ais_on_26	lben_ais_on_25	lben_ais_on_24

Each of these registers bits enables the insertion of AIS to an outgoing line during per-DS1 local loopback.

lben_ais_on_[27:00]:
Each of the lben_ais_on_[27:00] bits enables one of the 28 DS1s to insert AIS during local loopback enabled in LBENAL[0:3] registers. If this bit is set to '1', AIS insertion to this DS1 is enabled. If this bit is set to '0', AIS insertion to this DS1 is disabled. All of these registers can be

read/write, and the reset value is 0x00.

3.8.2 Per-DS0 Network Loopback RAM

The Loopback Buffer RAM is 32 bits wide; 16 longwords (64 bytes) are used per DS1 channel to store two frames of DS1 data and control bits, for a total size of 488 (16 x 28 channels) longwords. If the Loopback RAM is located at absolute address LBACK_RAM_BASE in the CPU's memory space, a channel n (from 0 to 27) will have a base address of [LBACK_RAM_BASE + (n x 64)].

As described earlier:

LBACK_RAM_BASE = 0xD000 (this is a base address for channel 0).

During CPU indirect write access to this LBACK_RAM. The following table is valid for writes to locations 00, 04, 08, 0C, 10, 14, 20, 24, 28, 2C, 30, 34, 38 of each channel.

bmask[3:0] (in CPUICMD bits [7:4])	**LBACK_RAM bits are written**
0000	No writes occurs
0001	Bits [07:00]
0011	Bits [15:08], [07:00]
0111	Bits [23:16], [15:08], [07:00]
1111	Bits [31:24], [23:16], [15:08], [07:00]

Note: For these bmask[3:0] enable bits, each bit set to ‘1’ indicates a byte write enable; ‘0’ indicates a byte write disable. Valid for byte enable combinations of up to 15 different byte writes.

Register Name: **LBACK_RAM_CH0 .. LBACK_RAM_CH27 (28 DS1s) - Network Loopback RAM**

Register Address: **0xD000, 0xD040, 0xD080, 0xD0C0, 0xD100, 0xD140, 0xD180, 0xD1C0, 0xD200, 0xD240, 0xD280, 0xD2C0, 0xD300, 0xD340, 0xD380, 0xD3C0, 0xD400, 0xD440, 0xD480, 0xD4C0, 0xD500, 0xD540, 0xD580, 0xD5C0, 0xD600, 0xD640, 0xD680, 0xD6C0 (28 Locations)**

<table>
<tr><th rowspan="2" colspan="2"></th><th colspan="32">Bit Function</th></tr>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><th rowspan="16">Address Location for 16 Longwords (Hex)</th><td>00</td><td colspan="8">ds0_00[7:0]</td><td colspan="8">ds0_01[7:0]</td><td colspan="8">ds0_02[7:0]</td><td colspan="8">ds0_03[7:0]</td></tr>
<tr><td>04</td><td colspan="8">ds0_04[7:0]</td><td colspan="8">ds0_05[7:0]</td><td colspan="8">ds0_06[7:0]</td><td colspan="8">ds0_07[7:0]</td></tr>
<tr><td>08</td><td colspan="8">ds0_08[7:0]</td><td colspan="8">ds0_09[7:0]</td><td colspan="8">ds0_10[7:0]</td><td colspan="8">ds0_11[7:0]</td></tr>
<tr><td>0C</td><td colspan="8">ds0_12[7:0]</td><td colspan="8">ds0_13[7:0]</td><td colspan="8">ds0_14[7:0]</td><td colspan="8">ds0_15[7:0]</td></tr>
<tr><td>10</td><td colspan="8">ds0_16[7:0]</td><td colspan="8">ds0_17[7:0]</td><td colspan="8">ds0_18[7:0]</td><td colspan="8">ds0_19[7:0]</td></tr>
<tr><td>14</td><td colspan="8">ds0_20[7:0]</td><td colspan="8">ds0_21[7:0]</td><td colspan="8">ds0_22[7:0]</td><td colspan="8">ds0_23[7:0]</td></tr>
<tr><td>18</td><td colspan="32">Unused</td></tr>
<tr><td>1C</td><td colspan="32">Unused</td></tr>
<tr><td>20</td><td colspan="8">ds0_00[7:0]</td><td colspan="8">ds0_01[7:0]</td><td colspan="8">ds0_02[7:0]</td><td colspan="8">ds0_03[7:0]</td></tr>
<tr><td>24</td><td colspan="8">ds0_04[7:0]</td><td colspan="8">ds0_05[7:0]</td><td colspan="8">ds0_06[7:0]</td><td colspan="8">ds0_07[7:0]</td></tr>
<tr><td>28</td><td colspan="8">ds0_08[7:0]</td><td colspan="8">ds0_09[7:0]</td><td colspan="8">ds0_10[7:0]</td><td colspan="8">ds0_11[7:0]</td></tr>
<tr><td>2C</td><td colspan="8">ds0_12[7:0]</td><td colspan="8">ds0_13[7:0]</td><td colspan="8">ds0_14[7:0]</td><td colspan="8">ds0_15[7:0]</td></tr>
<tr><td>30</td><td colspan="8">ds0_16[7:0]</td><td colspan="8">ds0_17[7:0]</td><td colspan="8">ds0_18[7:0]</td><td colspan="8">ds0_19[7:0]</td></tr>
<tr><td>34</td><td colspan="8">ds0_20[7:0]</td><td colspan="8">ds0_21[7:0]</td><td colspan="8">ds0_22[7:0]</td><td colspan="8">ds0_23[7:0]</td></tr>
<tr><td>38</td><td colspan="8">Unused</td><td colspan="24">lben[23:0]</td></tr>
<tr><td>3C</td><td colspan="32">Unused</td></tr>
</table>

These RAM contents are unknown after a power reset. Therefore, the user must program default values for control bits (lben). Shown above are the 16 longwords (64 bytes) assigned to a per-DS1 network loopback buffer.

ds0_00[7:0] .. ds0_23[7:0]:
These byte locations store two RX frames of DS1 data (frame n, frame n+1).Frame n’s address starts at 0x00, bit[31:24] and goes through 0x14, bit[07:00] for one frame’s worth of 24 DS0 octets. Frame n+1’s address starts at 0x20, bit[31:24] and goes through 0x34, bit[07:00] for another one frame’s worth of 24 DS0 octets (frame bit position not stored). All of these byte locations are RO (read only), and the reset value is unknown.

lben[23:0]:
Each of the lben[23:0] bits enables one of the 24 DS0s network loopback. If this bit is set to 1, this DS0’s network loopback is enabled. If this bit is set to 0, this DS0’s network loopback is disabled (see related bit settings in the LBENAMISC register). All of these bits are R/W.

3.9 Miscellaneous Scalar Registers

Register Name: CONTROL_FE0 .. CONTROL_FE27
- Front End Line Interface Control (28 DS1's)
Register Address:0x40 .. 0x5B (28 Locations)

Bit	Type	Function	Reset
7	R/W	use_inv_lc	0
6	R/W	tx_ena_rcntr	0
5	R/W	rx_ena_rcntr	0
4		Unused	X
3	R/W	inv_lc	0
2	R/W	inv_ld	0
1	R/W	use_pn_rail	0
0	R/W	ami_not_b8zs	0

These per-DS1 registers allow flexible control of low level interface details from the LIU to the RX and TX front end. Note that a single control bit affects both the RX and TX sides, except the use_inv_lc bit.

use_inv_lc:
The use_inv_lc bit, implemented only in bit 7 of the CONTROL_FE0 register, controls the RXCLK of all 28 DS1s. Implemented only in bit 7 of the CONTROL_FE1 register, it controls the TXCLK of all 28 DS1s. The remaining bit 7's of the CONTROL_FE[2:27] registers are unused. When the use_inv_lc bit is set to logical 1 in bit 7 of the CONTROL_FE0 register, the inv_lc bit function is inverted for RXCLK. When the use_inv_lc bit is set to logical 1 in bit 7 of the CONTROL_FE1 register, the inv_lc bit function is inverted for TXCLK. If both of these use_inv_lc bits are set to logical 0, then the inv_lc bit function is normal. See the inv_lc bit section for a table which details this interaction.

tx_ena_rcntr:
The tx_ena_rcntr bit is used to control the re-center operation when the VSC9670 detects a system transmit backend data overflow or underflow condition, in Low Latency mode only. If this bit is set to logical 1, the re-center operation is enabled when the tx_be_ufl_ofl bit is read as 1 in CONTROL_BE register. If this bit is set to logical 0, the re-center operation is disabled.

rx_ena_rcntr:
The rx_ena_rcntr bit is used to control the re-center operation when the VSC9670 detects a system receive backend data overflow or underflow condition, in Low Latency mode only. If this bit is set to logical 1, the re-center operation is enabled when the rx_be_ufl_ofl bit is read as 1 in CONTROL_BE register. If this bit is set to logical 0, the re-center operation is disabled.

inv_lc:
The inv_lc bit controls the front end line clock interface along with use_inv_lc bit. When this bit is set to logical 1, the TXCLK/RXCLK line clocks use the falling edge. When this bit is set to logical 0, the TXCLK/RXCLK line clocks use the rising edge. The use_inv_lc bit and the inv_lc bits interact according to the following table:

use_inv_lc (bit 7 in CONTROL_FE0)	**inv_lc**	**Select RXCLK edge**
0	0	Rising edge
0	1	Falling edge
1	0	Falling edge
1	1	Rising edge

use_inv_lc (bit 7 in CONTROL_FE1)	**inv_lc**	**Select TXCLK edge**
0	0	Rising edge
0	1	Falling edge
1	0	Falling edge
1	1	Rising edge

inv_ld:
The inv_ld bit controls the front end line data interface. When this bit is set to logical 1, the TXDATA/RXDATA line data is inverted. When this bit is set to logical 0, the TXDATA/RXDATA line data is not inverted.

use_pn_rail:
The use_pn_rail bit selects either the dual-rail formatted data or single-rail formatted data pin function for the front end line interface. In Slip Buffer mode or Pin Efficient mode, if this bit is set to logical 1, the VSC9670 can be used with 3 signal pins for the LIU interface: TXCLK/RXCLK, TXPRAIL/RXPRAIL and TXNRAIL/RXNRAIL. If this bit is set to logical 0, the VSC9670 can be used with 3 signal pins for the LIU interface: TXCLK/RXCLK, TXDATA/RXDATA and TXBPV/RXBPV. In Data Termination mode or Low Latency mode, the use_pn_rail bit must be set to 0, and the VSC9670 can only be used with 2 signal pins for the LIU interface: TXCLK/RXCLK and TXDATA/RXDATA.

ami_not_b8zs:
The ami_not_b8zs bit controls the front end line interface. When this bit is set to logical 1, the AMI line coding is used. When this bit is set to logical 0, the B8ZS line coding is used.

Register Name: CONTROL_BE0 .. CONTROL_BE27
- Back End System Interface Control (28 DS1s)
Register Address: 0x60 .. 0x7B (28 Locations)

Bit	Type	Function	Reset
7	R/W	use_inv_sc	0
6	R/W	tx_bigshr	0
5	R/W	rx_bigshr	0
4	R/W	rx_tx_cgap_fbit	0
3	RO, W1C	tx_be_ufl_ofl	0
2	RO, W1C	rx_be_ufl_ofl	0
1	R/W	inv_ss	0
0	R/W	inv_sc	0

These per-DS1 registers allow flexible control of the system side interface.

use_inv_sc:
The use_inv_sc bit, only implemented in bit 7 of the CONTROL_BE0 register, controls all 28 DS1s' SYSRXCLK. Implemented only in bit 7 of the CONTROL_BE1 register, it controls all 28 DS1s' SYSTXCLK. The remaining bit 7's of the CONTROL_BE[2:27] registers are unused. When the use_inv_sc bit is set to logical 1 in bit 7 of the CONTROL_BE0 register, the inv_sc bit function is inverted for SYSRXCLK. When the use_inv_sc bit is set to logical 1 in bit 7 of the CONTROL_BE1 register, the inv_sc bit function is inverted for SYSTXCLK. If both use_inv_sc bits are set to logical 0, the inv_sc bit function is normal.

tx_bigshr:
The tx_bigshr bit is used to select either a 32-bit or 72-bit size shift register in the TX system backend. If this bit is set to logical 1, the TX system backend uses a 72-bit size shift register. If this bit is set to logical 0, the TX system backend uses a 32-bit size shift register.

rx_bigshr:
The rx_bigshr bit is used to select either a 32-bit or 72-bit size shift register in the RX system backend. If this bit is set to logical 1, the RX system backend uses a 72-bit size shift register. If this bit is set to logical 0, the RX system backend uses a 32-bit size shift register.

rx_tx_cgap_fbit:
The rx_tx_cgap_fbit bit controls gapping of the SYSRXCLK and SYSTXCLK clocks at the framing bit position, in Data Termination mode only. When this bit is set to logical 1, the SYSRXCLK and SYSTXCLK clocks will be gapped at the framing bit position. When this bit is set to logical 0, the SYSRXCLK and SYSTXCLK clocks will be normal.

tx_be_ufl_ofl:
The tx_be_ufl_ofl bit (read only) indicates a system transmit backend data overflow or underflow condition, in Low Latency mode only. When this bit is read as logical 1, it means an overflow or underflow condition occurred in the system transmit backend data. When this bit is read as logical 0, it means the system transmit backend data is normal.
This bit is write-1-to-clear.

rx_be_ufl_ofl:
The rx_be_ufl_ofl bit (read only) indicates a system receive backend data overflow or underflow condition, in Low Latency mode only. When this bit is read as logical 1, it means an overflow or underflow condition occurred in the system receive backend data. When this bit is read as logical 0, the system receive backend data is normal. This bit is write-1-to-clear.

inv_ss:
The inv_ss bit controls both the TX and RX backend system sync interfaces. When this bit is set to logical 1, the SYSTXSYNC and SYSRXSYNC lines are active low. When this bit is set to logical 0, the SYSTXSYNC and SYSRXSYNC lines are active high.

In Slip Buffer mode only, the inv_ss bit is only implemented in bit 1 of the CONTROL_BE0 register, to control both the TX and RX backend system common sync interfaces. The remaining bit 1’s of the CONTROL_BE[1:27] registers are unused. When this bit is set to logical 1, the SYSTXCSYNC and SYSRXCSYNC lines are active low. When this bit is set to logical 0, the SYSTXCSYNC and SYSRXCSYNC lines are active high. These two common sync signals are shared by all 28 DS1s.

inv_sc:
The inv_sc bit controls both the TX and RX backend system clock interfaces along with use_inv_sc bit. When this bit is set to logical 1, the SYSTXCLK and SYSRXCLK clock pins use the falling clock edge. When this bit is set to logical 0, the SYSTXCLK and SYSRXCLK clock pins use the rising clock edge. The use_inv_sc bit and the inv_sc bit interact as follows:

use_inv_sc (bit 7 in CONTROL_BE0)	**inv_sc**	**Select SYSRXCLK edge**
0	0	Rising edge
0	1	Falling edge
1	0	Falling edge
1	1	Rising edge

use_inv_sc (bit 7 in CONTROL_BE1)	**inv_sc**	**Select SYSTXCLK edge**
0	0	Rising edge
0	1	Falling edge
1	0	Falling edge
1	1	Rising edge

In Slip Buffer mode only, the inv_sc bit is only implemented in bit 0 of the CONTROL_BE0 register, to control both the TX and RX backend system common clock interfaces. The remaining bit 0’s of the CONTROL_BE[1:27] registers are unused. When this bit is set to logical 1, the SYSTXCCLK and SYSRXCCLK clock pins use falling edge. When this bit is set to logical 0, the SYSTXCCLK and SYSRXCCLK clock pins use rising edge. These two common clock signals are shared by all 28 DS1s.

Register Name: **DJB_TX_03_00,DJB_TX_07_04,DJB_TX_11_08,DJB_TX_15_12, DJB_TX_19_16, DJB_TX_23_20, DJB_TX_27_24 - De-Jitter Buffer TX Control**

Register Address: **0xC0, 0xC1, 0xC2, 0xC3, 0xC4, 0xC5, 0xC6**

Register Name (Address)	Bit Function							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DJB_TX_03_00 (0xC0)	djbtx_s1_03	djbtx_s0_03	djbtx_s1_02	djbtx_s0_02	djbtx_s1_01	djbtx_s0_01	djbtx_s1_00	djbtx_s0_00
DJB_TX_07_04 (0xC1)	djbtx_s1_07	djbtx_s0_07	djbtx_s1_06	djbtx_s0_06	djbtx_s1_05	djbtx_s0_05	djbtx_s1_04	djbtx_s0_04
DJB_TX_11_08 (0xC2)	djbtx_s1_11	djbtx_s0_11	djbtx_s1_10	djbtx_s0_10	djbtx_s1_09	djbtx_s0_09	djbtx_s1_08	djbtx_s0_08
DJB_TX_15_12 (0xC3)	djbtx_s1_15	djbtx_s0_15	djbtx_s1_14	djbtx_s0_14	djbtx_s1_13	djbtx_s0_13	djbtx_s1_12	djbtx_s0_12
DJB_TX_19_16 (0xC4)	djbtx_s1_19	djbtx_s0_19	djbtx_s1_18	djbtx_s0_18	djbtx_s1_17	djbtx_s0_17	djbtx_s1_16	djbtx_s0_16
DJB_TX_23_2 (0xC5)	djbtx_s1_23	djbtx_s0_23	djbtx_s1_22	djbtx_s0_22	djbtx_s1_21	djbtx_s0_21	djbtx_s1_20	djbtx_s0_20
DJB_TX_27_24 (0xC6)	djbtx_s1_27	djbtx_s0_27	djbtx_s1_26	djbtx_s0_26	djbtx_s1_25	djbtx_s0_25	djbtx_s1_24	djbtx_s0_24

The following table shows clocking scheme choices for the VCS9670:

djbrx[xx]	djbrx[xx]	djbrx[xx]	RXCLK	TXCLK
0	0	0	RX Line Clock	SYSTXCLK
0	0	1	RX Line Clock	SYSTXCLK (De-Jittered if djbrx[xx] is set; Low Latency Mode only)
0	1	0	RX Line Clock	REF1544
0	1	1	RX Line Clock	INT1544
1	0	0	RX Line Clock (De-Jittered if djbrx[xx] is set)	SYSTXCLK
1	0	1	RX Line Clock (De-Jittered if djbrx[xx] is set	RX Line Clock (De-Jittered if djbrx[xx] is set)
1	1	0	RX Line Clock (De-Jittered if djbrx[xx] is set	REF1544
1	1	1	RX Line Clock (De-Jittered if djbrx[xx] is set	INT1544

Each DS1 channel uses two bits to select the De-Jitter Buffer function for the TX direction. If the De-Jitter Buffer is enabled for the TX direction, the user can't use the same De-Jitter Buffer for the RX direction.

djbtx_s1_[27:00], djbtx_s0_[27:00]:
Both djbtx_s1_xx and djbtx_s0_xx bits allow software to program one of the 28 DS1 De-Jitter Buffers and select the transmit clock source for the TX block, as shown in the following table. Each register controls four DS1channels. All these registers can be read/write, and the reset value is 0xAA (Transmit Clock Selected = REF1544).

djbtx_s1_xx, djbtx_s0_xx	**Transmit Clock Selection**
00	SYSTXCLK (external transmit input clock)
01	DJB1544 (De-Jitter Buffer output & input from SYSTXCLK)
10	REF1544 (external reference input clock) or one of 28 DJ_RXCLKS
11	INTREF1544 (De-Jitter Buffer output & input from CLK45)

Register Name: RX2REL_SEL - Select RXCLK to REF1544
Register Address:0xCF

Bit	Type	Function	Reset
7:6		Unused	X
5	R/W	rx2ref_sel_ena	0
4:0	R/W	rx2ref_sel[4:0]	0x00

rx2ref_sel_ena:
The rx2ref_sel_ena bit is used to enable RXCLK for REF1544. If this bit is set to 1, the RXCLK is enabled to drive REF1544. If this bit is set to 0, the RXCLK is disabled to drive REF1544.

rx2ref_sel[4:0]:
The rx2ref_sel[4:0] bits select one of 28 RXCLKs to drive REF1544 when the rx2ref_sel_ena bit is enabled.

Register Name: DJB_RX_07_00, DJB_RX_15_08, DJB_RX_23_16, DJB_RX_27_24 - De-Jitter Buffer RX Control
Register Address: 0xC7, 0xC8, 0xC9, 0xCA

Register Name (Address)	Bit Function							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DJB_RX_07_00(0xC7)	djbrx07	djbrx06	djbrx05	djbrx04	djbrx03	djbrx02	djbrx01	djbrx00
DJB_RX_15_08(0xC8)	djbrx15	djbrx14	djbrx13	djbrx12	djbrx11	djbrx10	djbrx09	djbrx08
DJB_RX_23_16(0xC9)	djbrx23	djbrx22	djbrx21	djbrx20	djbrx19	djbrx18	djbrx17	djbrx16
DJB_RX_27_24(0xCA)	Unused	Unused	Unused	Unused	djbrx27	djbrx26	djbrx25	djbrx24

Each of these register bits is used to enable a DS1 channel's De-Jitter Buffer for the RX direction. If the De-Jitter Buffer is enabled for the RX direction, the user can't use the same De-Jitter Buffer for the TX direction.

djbrx[27:00]:
Each of these djbrx[27:00] bits enables one of the 28 De-Jitter Buffer's function for the RX direction. If this bit is set to 1, the receive De-Jitter Buffer is enabled, and the DJB1544 internal clock which outputs from the De-Jitter Buffer is used for the RX block. If this bit is set to 0, the De-Jitter Buffer is disabled, and the RXCLK external input clock is used for the RX block.

All these registers can be read/write, and the reset value is 0x00.

Register Name: **DJB_EN_07_00, DJB_EN_15_08, DJB_EN_23_16, DJB_EN_27_24 - De-Jitter Buffer Enables**
Register Address: **0xCB, 0xCC, 0xCD, 0xCE**

Register Name (Address)	Bit Function							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DJB_EN_07_00(0xCB)	djbena07	djbena06	djbena05	djbena04	djbena03	djbena02	djbena01	djbena00
DJB_EN_15_08(0xCC)	djbena15	djbena14	djbena13	djbena12	djbena11	djbena10	djbena09	djbena08
DJB_EN_23_16(0xCD)	djbena23	djbena22	djbena21	djbena20	djbena19	djbena18	djbena17	djbena16
DJB_EN_27_24(0xCE)	Unused	Unused	Unused	Unused	djbena27	djbena26	djbena25	djbena24

Each of these register bits is used to enable a DS1 channel's De-Jitter Buffer for either the TX or RX direction. The De-Jitter Buffer TX Control Registers implement the transmit direction, and the De-Jitter Buffer RX Control Registers implement the receive direction. There is only one De-Jitter Buffer available for each DS1 channel; therefore, it is used in either the TX or RX direction.

djbena[27:00]:
Each of the djbena[27:00] bits enables one of the 28 De-Jitter Buffer's function. When this bit is set to 1, the De-Jitter Buffer is enabled. When this bit is set to 0, the De-Jitter Buffer is disabled.

All these registers can be read/write, and the reset value is 0x00.

Register Name: DMWVAL_0, DMWVAL_1, DMWVAL_2, DMWVAL_3, DMWVAL_4, DMWVAL_5, DMWVAL_6, DMWVAL_7 - Digital Milliwatt Octets Value
Register Address: 0xE0, 0xE1, 0xE2, 0xE3, 0xE4, 0xE5, 0xE6, 0xE7

Register Name (Address)	Type	Bit Function	Reset
DMWVAL_0 (0xE0)	R/W	DMWVAL_0 [7:0]	0x1E
DMWVAL_1 (0xE1)	R/W	DMWVAL_1 [7:0]	0x0B
DMWVAL_2 (0xE2)	R/W	DMWVAL_2 [7:0]	0x0B
DMWVAL_3 (0xE3)	R/W	DMWVAL_3 [7:0]	0x1E
DMWVAL_4 (0xE4)	R/W	DMWVAL_4 [7:0]	0x9E
DMWVAL_5 (0xE5)	R/W	DMWVAL_5 [7:0]	0x8B
DMWVAL_6 (0xE6)	R/W	DMWVAL_6 [7:0]	0x8B
DMWVAL_7 (0xE7)	R/W	DMWVAL_7 [7:0]	0x9E

The registers DMWVAL_[0:7] are used for PCM octets that represent a 1-KHz sine wave. Power up defaults are 0x1E, 0x0B, 0x0B, 0x1E, 0x9E, 0x8B, 0x8B, 0x9E. These values are used in the TX direction, per-DS0 or per-DS1 pattern insertion control logic in the TX_RAM register.

Register Name: IDLECODE - Idle Code Value
Register Address: 0xE8

Register Name (Address)	Type	Bit Function	Reset
IDLECODE (0xE8)	R/W	IDLECODE [7:0]	0x17

This register is used for TX IDLE code. The power up default is 0x17. When enabled on a per-DS0 or per-DS1 basis in the TX_RAM register, this value can be inserted in the TX direction (upstream to the network).

4.0 Electrical Specifications

All timing is given in nanoseconds (ns) and frequency is in MegaHertz (MHz) unless otherwise noted.

4.1 Absolute Maximum Ratings

Absolute maximum ratings are the worst-case limits that the device can withstand without sustaining permanent damage.

Table 4.1 Absolute Maximum Ratings

Ambient Temperature	-40° C TO +85° C
Storage Temperature	-40° C TO +125° C
Supply Voltage (Min, Max)	-0.5V TO +6.0V
Voltage On Any Pin	-0.3V TO V_{DD} + 0.3V
Dc Input Current On Any Pin	± 20 mA
Lead Temperature	300° C
Power Dissipation	2 W

4.2 D.C. Characteristics

T_A = -40° C to +85° C;

V_{DD} = 3.3V ± 10%;

V_{DD} <= V_{CC_BIAS} <= 5.5V

Table 4.2 D.C. Characteristics

Symbol	Parameter	Min	Typ	Max	Units	Conditions
V_{DD_Core} V_{DD_IO}	Power Supply		3.3		V	
V_{CC_BIAS}	5V Tolerant Bias		5.0		V	
I_{VCC_BIAS}	Current into 5V BIAS		6.0		µA	V_{BIAS} = 5.5V
V_{IL}	Input Low Voltage	-0.5		0.8	V	Guaranteed Input LOW Voltage
V_{IH}	Input High Voltage	2.0		V_{CC_BIAS}	V	Guaranteed Input HIGH Voltage
V_{OL}	Output or Bi-directional Low Voltage		0.25	0.4	V	V_{DD} = min I_{OL} = -8 mA for CPU interface bus and -4 mA for others
V_{OH}	Output or Bi-directional High Voltage	2.4			V	V_{DD} = min, I_{OL} = 8 mA for CPU interface bus and 4 mA for others
I_{IL}	Input Low Current	-10		-10	µA	V_{IL} = 0 V
I_{DD}	Active Operating Current			550	mA	V_{DD} = 3.63V,Outputs Unloaded, CLK45 = 44.736 MHz
C_{IN}	Input Capacitance		5		pF	Excluding Package, Package typically 2 pF
C_{OUT}	Output Capacitance		5		pF	
C_{IO}	Bi-directional Capacitance		5		pF	

Notes:

Negative currents flow out of device and positive currents flow into device.

4.3 A.C. Timing Characteristics

T_A = -40° C to +85° C; V_{DD} = 3.3V ± 10%.

4.3.1 Microprocessor Interface Timing

Table 4.3 Microprocessor Write and Read Access Parameters - Motorola Bus Mode

Symbol	Parameter	Min	Max
tIA_{SEL}	CPU_SELN InActive Time	45	
tS_{DSEL}	Data to CPU_SELN or CPU_RDN_RDWRN high Set-up Time	9	
tH_{DSEL}	Data Hold Time from CPU_SELN or CPU_RDN_RDWRN high	0	
tS_{ASEL}	Address to CPU_SELN or CPU_RDN_RDWRN low Set-up Time	9	
tH_{ASEL}	Address Hold Time from CPU_SELN or CPU_RDN_RDWRN low	0	
tV_{SELP}	Valid CPU_SELN, CPU_RDN_RDWRN Pulse Width	45	
tV_{SELA}	CPU_SELN low to CPU_RDY_DTACKN low or Data Valid		15.1
tP_{SELA}	CPU_SELN high to CPU_RDY_DTACKN high or Data Invalid		15.1

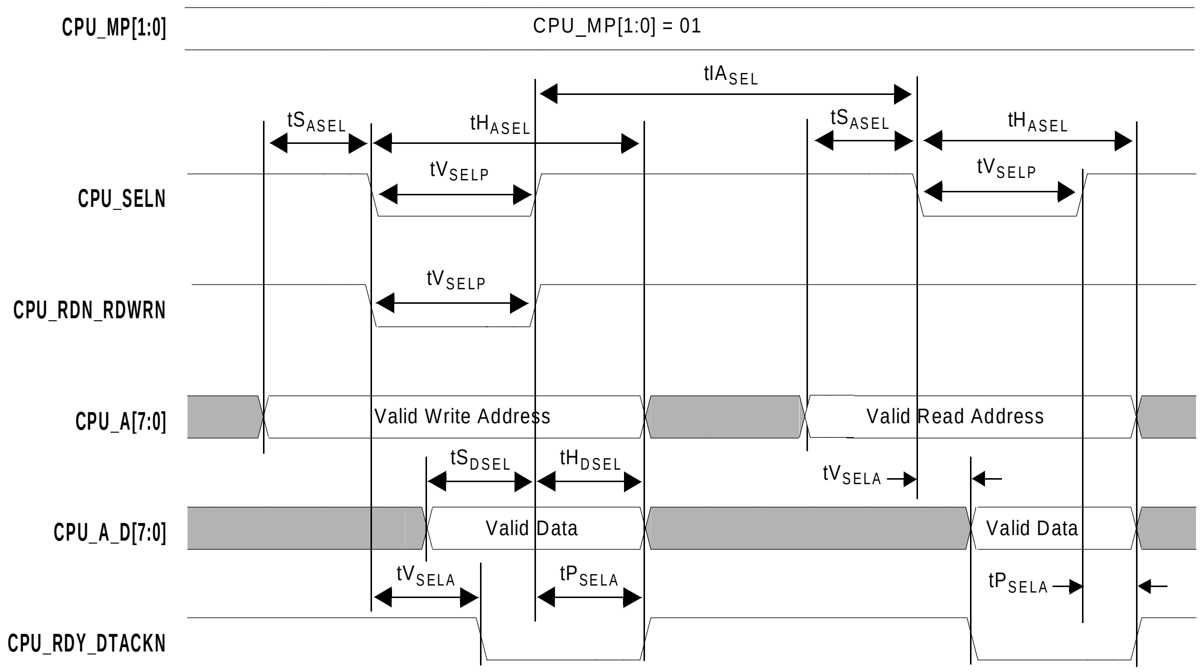


Figure 4.1 Microprocessor Write and Read Access Timing for Motorola Bus Mode

Table 4.4 Microprocessor Write and Read Access Parameters - Intel Bus Mode

Symbol	Parameter	Min	Max
tIA_{SEL}	CPU_SELN InActive Time	45	
tS_{DSEL}	Data to CPU_SELN or CPU_WRN high Set-up Time	9	
tH_{DSEL}	Data Hold Time from CPU_SELN or CPU_WRN high	0	
tS_{ASEL}	Address to CPU_SELN, CPU_WRN or CPU_RDN_RDWRN low Set-up Time	9	
tH_{ASEL}	Address Hold Time from CPU_SELN, CPU_WRN or CPU_RDN_RDWRN low	0	
tV_{SELP}	Valid CPU_SELN, CPU_WRN, CPU_RDN_RDWRN Pulse Width	45	
tV_{SELA}	CPU_SELN or CPU_RDN_RDWRN low to Data Valid		15.1
tP_{SELA}	CPU_SELN or CPU_RDN_RDWRN high to Data Invalid		15.1

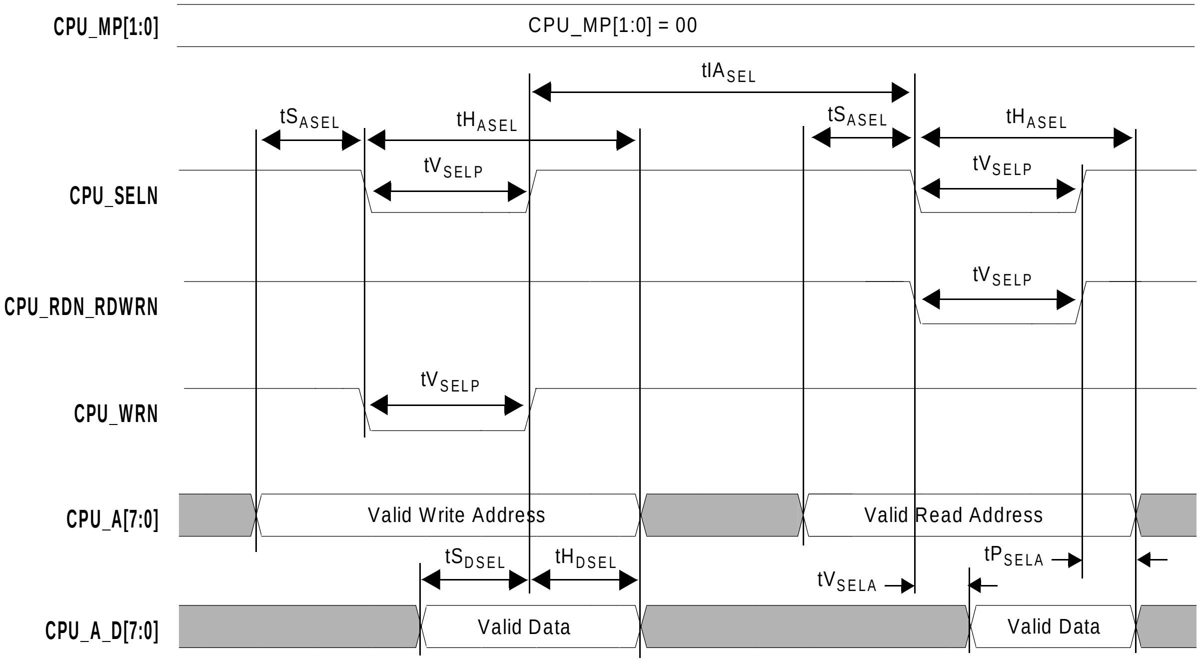


Figure 4.2 Microprocessor Write and Read Access Timing for Intel Bus Mode

Table 4.5 Microprocessor Write and Read Access Parameters - Multiplexed Bus Mode

Symbol	Parameter	Min	Max
tIA_{ALE}	CPU_ALE InActive Time	45	
tIA_{SEL}	CPU_SELN InActive Time	45	
tS_{DSEL}	Data to CPU_SELN or CPU_WRN high Set-up Time	6.8	
tH_{DSEL}	Data Hold Time from CPU_SELN or CPU_WRN high	0	
tS_{AALE}	Address to CPU_ALE low Set-up Time	6.8	
tH_{AALE}	Address Hold Time from CPU_ALE low	0	
tV_{SELP}	Valid CPU_SELN, , CPU_WRN, CPU_RDN_RDWRN Pulse Width	45	
tV_{SELA}	CPU_SELN or CPU_RDN_RDWRN low to Data Valid		15.1
tP_{SELA}	CPU_SELN or CPU_RDN_RDWRN high to Data Invalid		15.1

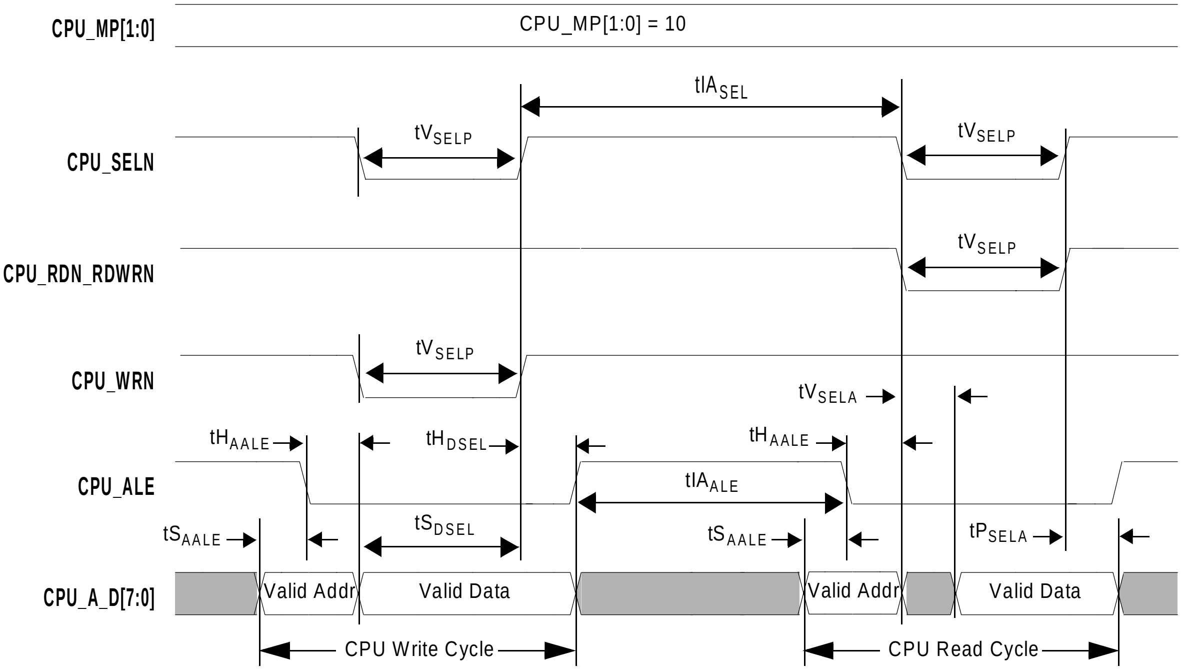


Figure 4.3 Microprocessor Write and Read Access Timing for Multiplexed Bus Mode

Table 4.6 Microprocessor Write and Read Access Parameters - Special Multiplexed Bus Mode

Symbol	Parameter	Min	Max
tIA_{ALE}	CPU_ALE InActive Time	45	
tIA_{SEL}	CPU_SELN InActive Time	45	
tS_{DWRN}	Data to CPU_WRN low Set-up Time	6.8	
tH_{DWRN}	Data Hold Time from CPU_WRN low	0	
tS_{AALE}	Address to CPU_ALE low Set-up Time	6.8	
tH_{AALE}	Address Hold Time from CPU_ALE low	0	
tV_{SELP}	Valid CPU_SELN, , CPU_WRN, CPU_RDN_RDWRN Pulse Width	45	
tV_{SELA}	CPU_SELN low to Data Valid		15.1
tP_{SELA}	CPU_SELN high to Data Invalid		15.1

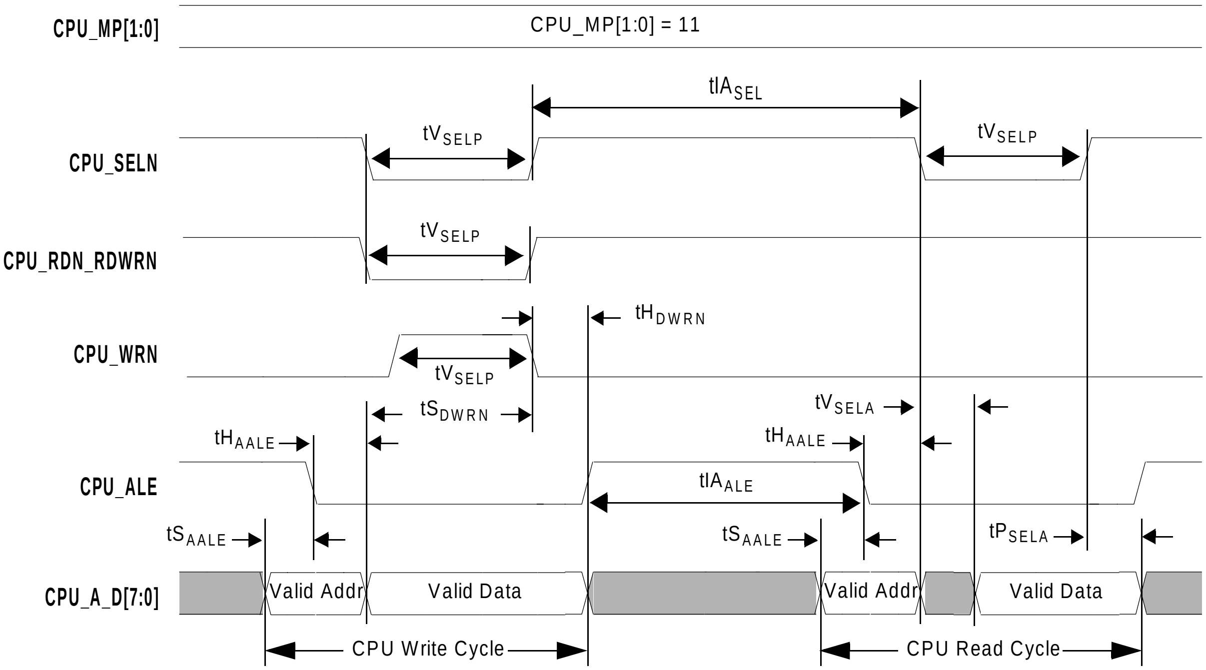


Figure 4.4 Microprocessor Write and Read Access Timing for Special Multiplexed Bus Mode

Notes on Microprocessor Write and Read Timing:

1. All input or output measurement points are at 1.4 volt.
2. Maximum output propagation delays are measured with a 50 pF load on the Microprocessor Interface.
3. In non-multiplexed address/data bus modes the CPU_ALE input shall be tied to high. All minimum output propagation delays are measured with a 5 pF load.

4.3.2 I/O Timing

T_A = -40° C to +85° C; V_{DD} = 3.3V ± 10%

Table 4.7 CLK45 (44.736 MHz) and REF1544 (1.544 MHz) Inputs

Symbol	Description	Min	Typ	Max
tL_{CLKW}	CLK45 Low Pulse Width (40% to 60% duty cycle)	8.9		13.4
tH_{CLKW}	CLK45 High Pulse Width (40% to 60% duty cycle)	8.9		13.4
tP_{CLK}	CLK45 Period (typically 44.736 MHz ± 32 ppm)		22.35	
tL_{REFW}	REF1544 Low Pulse Width (40% to 60% duty cycle)	259		388.6
tH_{REFW}	REF1544 High Pulse Width (40% to 60% duty cycle)	259		388.6
tP_{REF}	REF1544 Period (typically 1.544 MHz ± 25 ppm)		647.7	

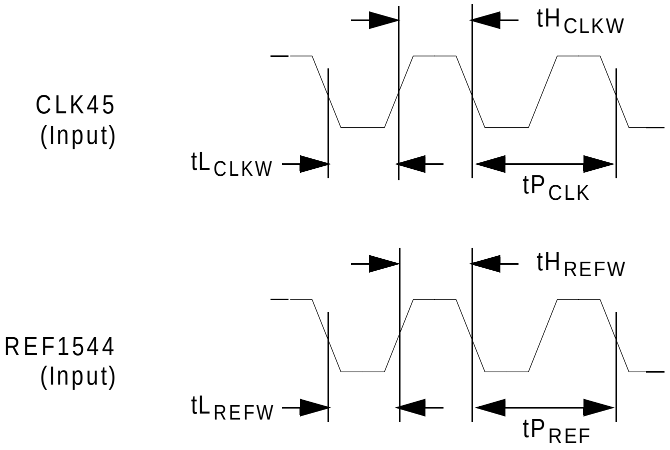


Figure 4.5 CLK45 and REF1544 Clocks Input Timing

Table 4.8 Receive Line Interface Timing

Symbol	Description	Min	Typ	Max
tP_{RXCLK}	RX Line Clock (1.544 MHz) Period		647.7	
tS_{RXCLK}	RXDATA[x]/RXPRAIL[x] or RXBPV[x]/RXNRAIL[x] to RXCLK[x] high or low Set-up Time	12		
tH_{RXCLK}	RXDATA[x]/RXPRAIL[x] or RXBPV[x]/RXNRAIL[x] Hold Time from RXCLK[x] high or low	0		

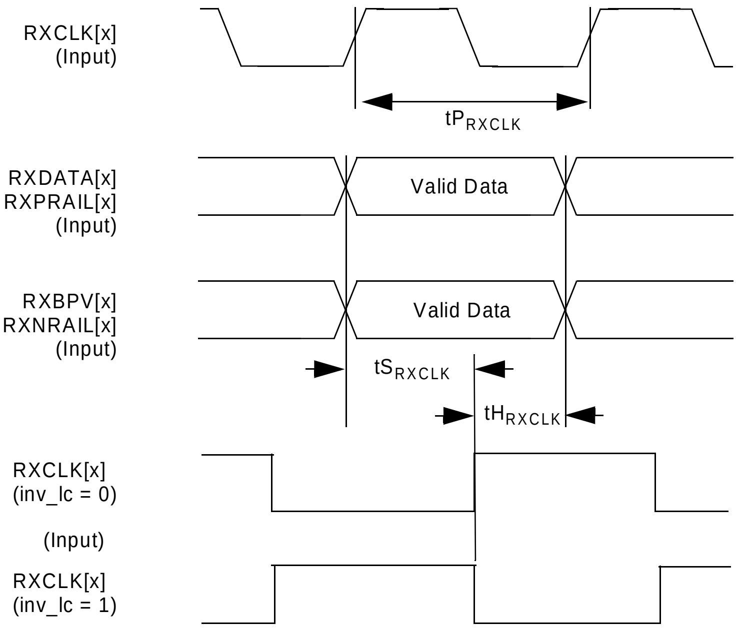


Figure 4.6 Receive Line Interface Input Timing

Table 4.9 Transmit Line Interface Timing

Symbol	Description	Min	Typ	Max
tP_{TXCLK}	TX Line Clock (1.544 MHz) Period		647.7	
tP_{TXD}	TXCLK[x] high or low to TXDATA[x]/TXPRAIL[x] or TXBPV[x]/ TXNRAIL[x] Output Valid	3		11.2

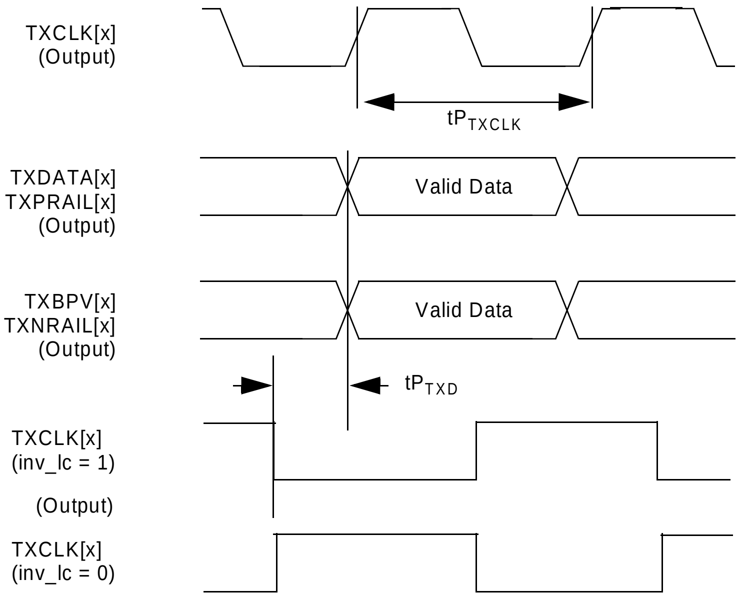


Figure 4.7 Transmit Line Interface Output Timing

Table 4.10 System Receive Interface Timing - Slip Buffer Mode

Symbol	Description	Min	Typ	Max
tP_{CCLK1}	RX System Common Clock (1.544 MHz) Period, ssp[1:0] = 00		647.7	
tP_{CCLK2}	RX System Common Clock (2.048 MHz) Period, ssp[1:0] = 01		488	
tP_{CCLK4}	RX System Common Clock (4.096 MHz) Period, ssp[1:0] = 10		244	
tP_{CCLK8}	RX System Common Clock (8.192 MHz) Period, ssp[1:0] = 11		122	
tP_{RXD}	SYSRXCCLK high or low to SYSRXDATA[x]/SYSRXSIG[x] Output Valid	5		21.7
tS_{CCLK}	SYSRXCSYNC to SYSRXCCLK high or low Set-up Time	5		
tH_{CCLK}	SYSRXCSYNC Hold Time from SYSRXCCLK high or low	-1.1		
tV_{DENB}	SYSRXDATA_ENBN low to SYSRXDATA valid			11.9
tZ_{DENB}	SYSRXDATA_ENBN high to SYSRXDATA hi-Z	1.3		3.3

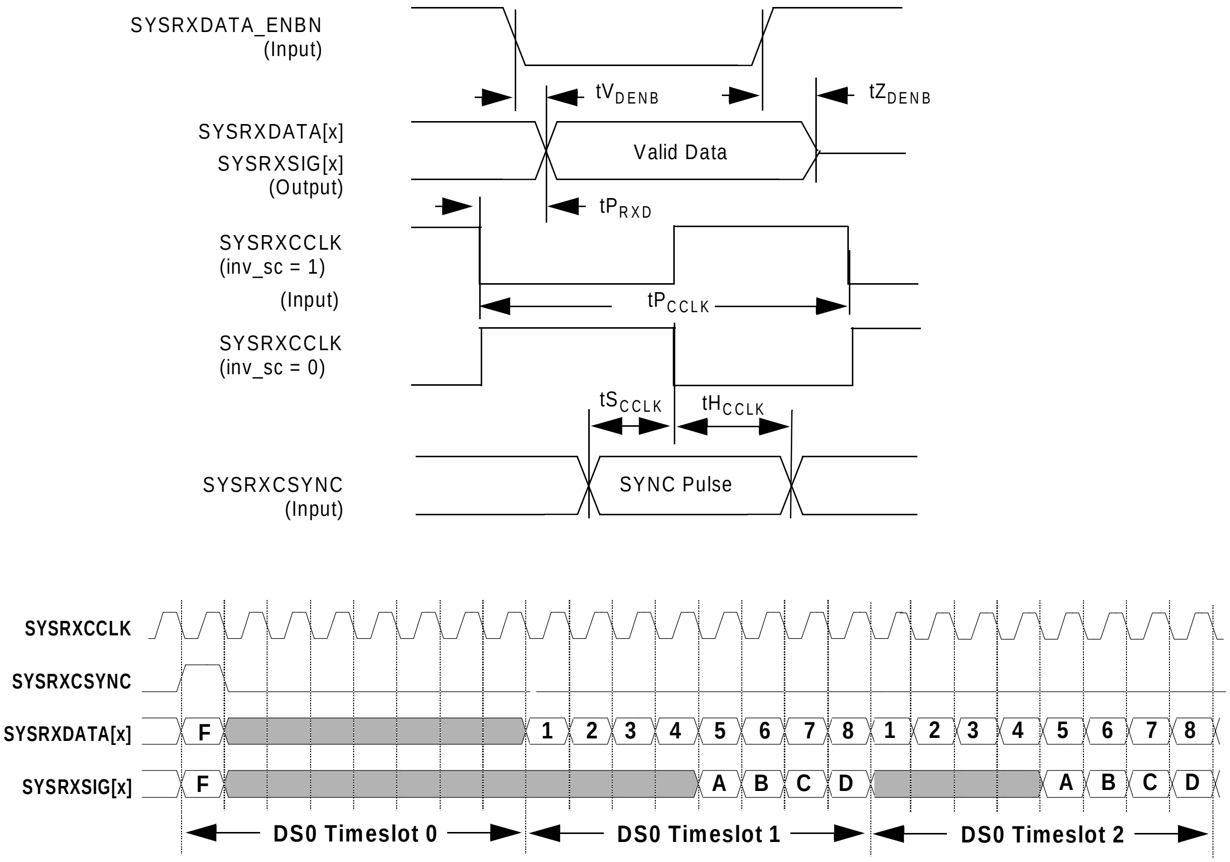


Figure 4.8 System Receive Interface and Functional Timing - Slip Buffer Mode

Table 4.11 System Receive Interface Timing - Data Termination or Low Latency Mode

Symbol	Description	Min	Typ	Max
tP_{RXCLK}	RX System Clock Output (1.544 MHz) Period		647.7	
tP_{RXD}	SYSRXCLK[x] high or low to SYSRXDATA[x] or SYSRXSYNC[x] Output Valid	1.9		12.2
tV_{DENB}	SYSRXDATA_ENBN low to SYSRXDATA valid			11.9
tZ_{DENB}	SYSRXDATA_ENBN high to SYSRXDATA hi-Z	1.3		3.3

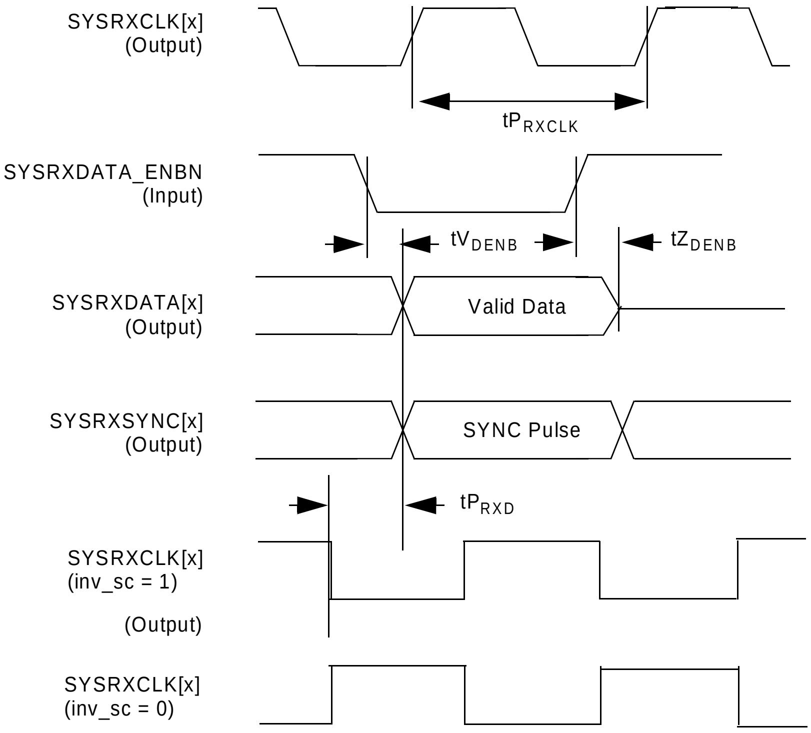


Figure 4.9 System Receive Interface Timing - Data Termination or Low Latency Mode

Table 4.12 System Receive Interface Timing - Pin Efficient Muxed Data Mode

Symbol	Description	Min	Typ	Max
tP_{CLK}	System Clock Input (44.736 MHz) Period		22.35	
tP_{MRXD}	CLK45 high to MUXRXDATA Output Valid	5		16.35
tP_{MSYNC}	CLK45 high to MUXRXSYNC or MUXRXSYNC45 Output Valid	5		16.7
tP_{MRXENA}	CLK45 high to MUXRXENA Output Valid	5		16.85
tV_{DENB}	SYSRXDATA_ENBN low to SYSRXDATA valid			11.9
tZ_{DENB}	SYSRXDATA_ENBN high to SYSRXDATA hi-Z	1.3		3.3

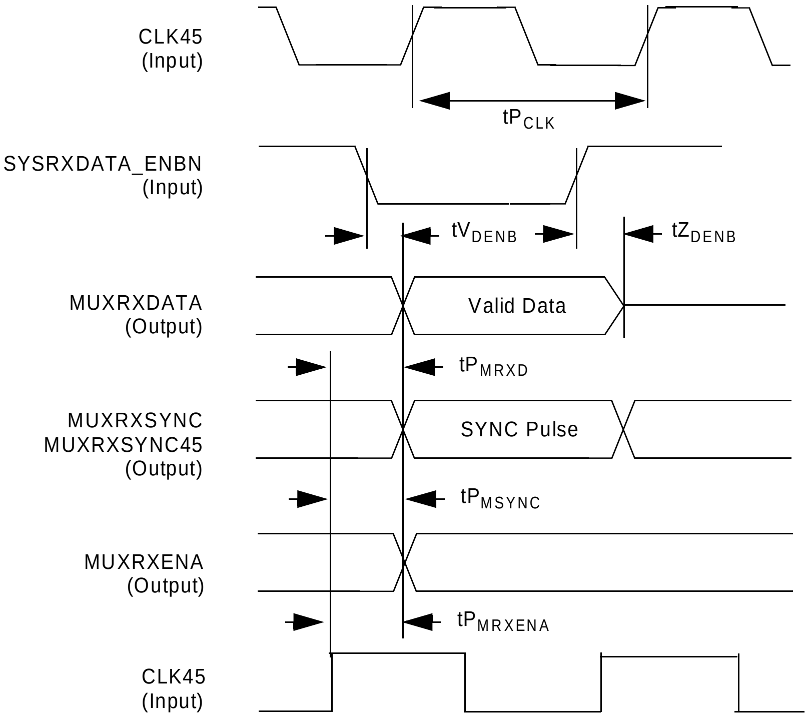


Figure 4.10 System Receive Interface Timing - Pin Efficient Muxed Data Mode

Table 4.13 System Transmit Interface Timing - Slip Buffer Mode

Symbol	Description	Min	Typ	Max
tP_{CCLK1}	TX System Common Clock (1.544 MHz) Period, ssp[1:0] = 00		647.7	
tP_{CCLK2}	TX System Common Clock (2.048 MHz) Period, ssp[1:0] = 01		488	
tP_{CCLK4}	TX System Common Clock (4.096 MHz) Period, ssp[1:0] = 10		244	
tP_{CCLK8}	TX System Common Clock (8.192 MHz) Period, ssp[1:0] = 11		122	
tS_{TXD}	SYSTXDATA[x] or SYSTXSIG[x] to SYSTXCCLK high or low Set-up Time	4.6		
tH_{TXD}	SYSTXDATA[x] or SYSTXSIG[x] Hold Time from SYSTXCCLK high or low	-0.1		
tS_{CCLK}	SYSTXCSYNC to SYSTXCCLK high or low Set-up Time	4.6		
tH_{CCLK}	SYSTXCSYNC Hold Time from SYSTXCCLK high or low	-1.1		

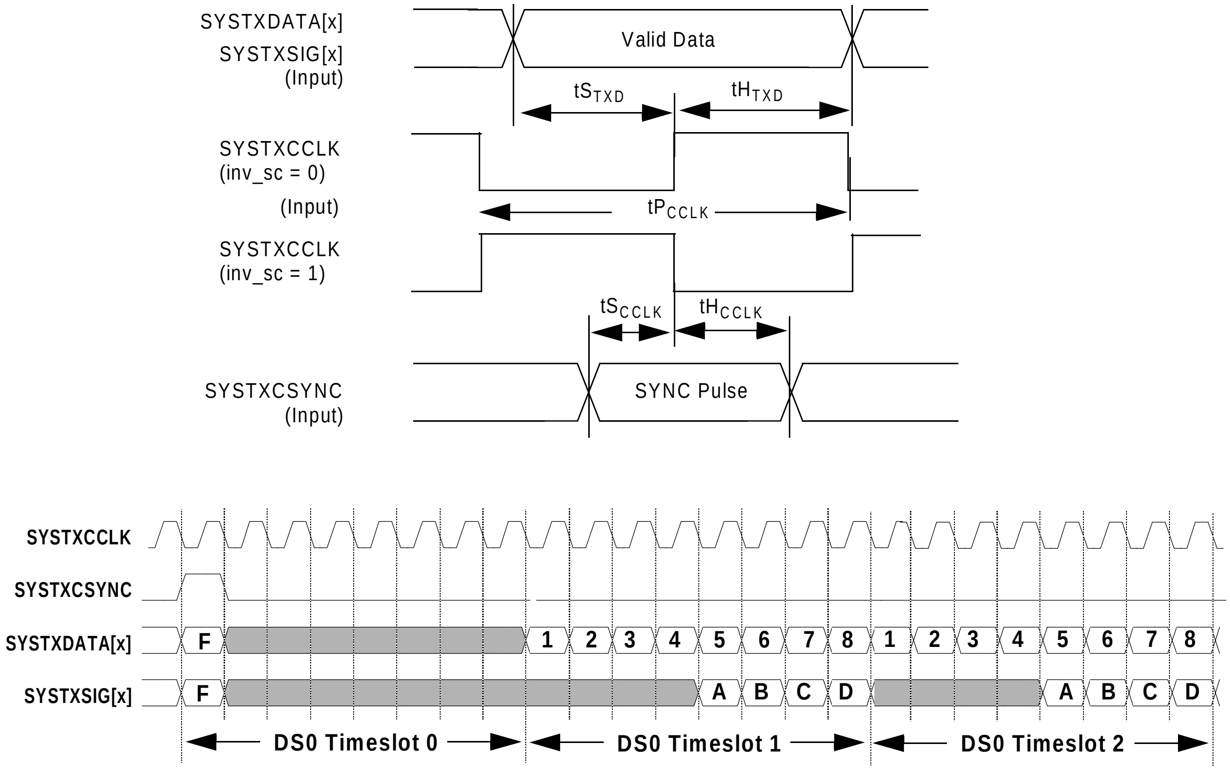


Figure 4.11 System Transmit Interface and Functional Timing - Slip Buffer Mode

Table 4.14 System Transmit Interface Timing - Data Termination Mode

Symbol	Description	Min	Typ	Max
tP_{SYSTXK}	TX System Clock (1.544 MHz) Period		647.7	
tP_{SYNC}	SYSTXCLK[x] high or low to SYSTXSYNC[x] Output Valid	2.7		12.7
tS_{TXD}	SYSTXDATA[x] to SYSTXCLK[x] high or low Set-up Time	21		
tH_{TXD}	SYSTXDATA[x] Hold Time from SYSTXCLK[x] high or low	-4.4		

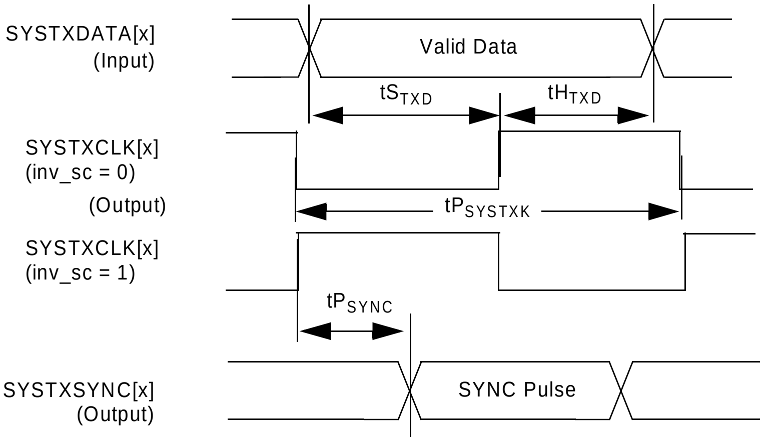


Figure 4.12 System Transmit Interface Timing - Data Termination Mode

Table 4.15 System Transmit Interface Timing - Low Latency Mode

Symbol	Description	Min	Typ	Max
tP_{SYSTXK}	TX System Clock (1.544 MHz) Period		647.7	
tS_{TXD}	SYSTXDATA[x] to SYSTXCLK[x] high or low Set-up Time	8.1		
tH_{TXD}	SYSTXDATA[x] Hold Time from SYSTXCLK[x] high or low	0		
tS_{SYNC}	SYSTXSYNC[x] to SYSTXCLK[x] high or low Set-up Time	8.1		
tH_{SYNC}	SYSTXSYNC[x] Hold Time from SYSTXCLK[x] high or low	0		

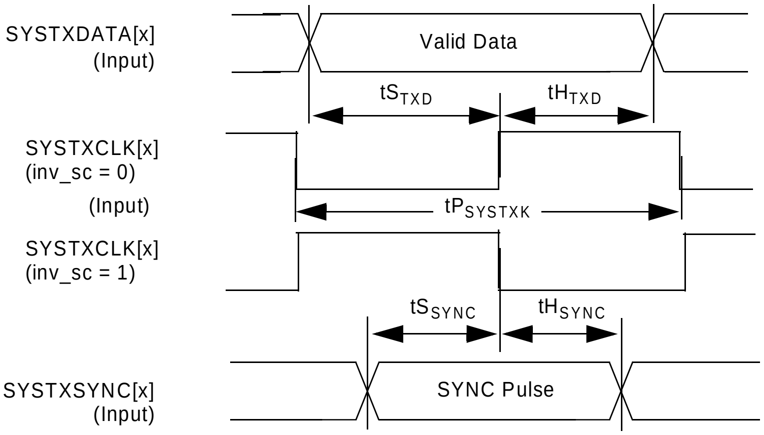


Figure 4.13 System Transmit Interface Timing - Low Latency Mode

Table 4.16 System Transmit Interface Timing - Pin Efficient Muxed Data Mode

Symbol	Description	Min	Typ	Max
tP_{CLK}	System Clock Input (44.736 MHz) Period		22.35	
tS_{MTXD}	MUXTXDATA to CLK45 high Set-up Time	5.8		
tH_{MTXD}	MUXTXDATA Hold Time from CLK45 high	-1.1		
tP_{MSYNC}	CLK45 high to MUXTXSYNC or MUXTXSYNC45 Output Valid	5.2		16.2
tP_{MTXENA}	CLK45 high to MUXTXENA Output Valid	5.2		16.1

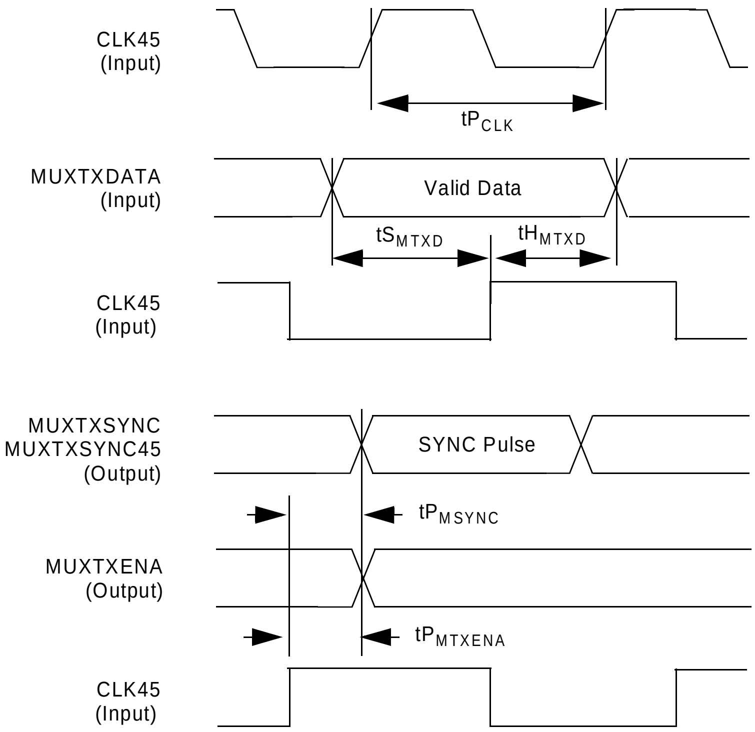


Figure 4.14 System Transmit Interface Timing - Pin Efficient Muxed Data Mode

Notes on Line and System TX/RX Interfaces Timing:

1. All input or output measurement points are at 1.4 volt.
2. Maximum output propagation delays are measured with a 25 pF load on all outputs.
3. All minimum output propagation delays are measured with a 5 pF load.

Table 4.17 JTAG and SCAN Interface Timing

Symbol	Description	Min	Max
t_{TCK}	TCK Frequency		5
DC_{TCK}	TCK Duty Cycle	40%	60%
tS_{TMS}	TMS Set-up Time to TCK high	50	
tH_{TMS}	TMS Hold Time from TCK high	50	
tS_{TDI}	TDI Set-up Time to TCK high	50	
tH_{TDI}	TDI Hold Time from TCK high	50	
tS_{TST}	TEST_EN or TRST_N Set-up Time to TCK high	50	
tH_{TST}	TEST_EN or TRST_N Hold Time from TCK high	50	
tP_{TDO}	TCK Low to TDO Output Valid	10	50

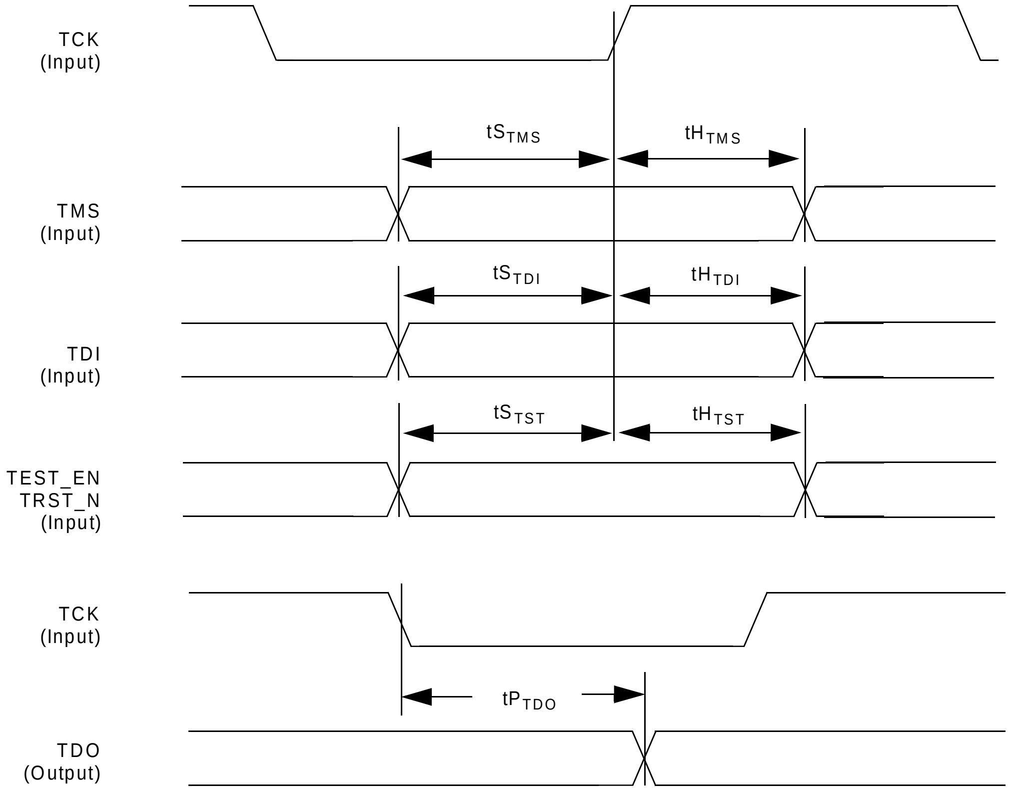
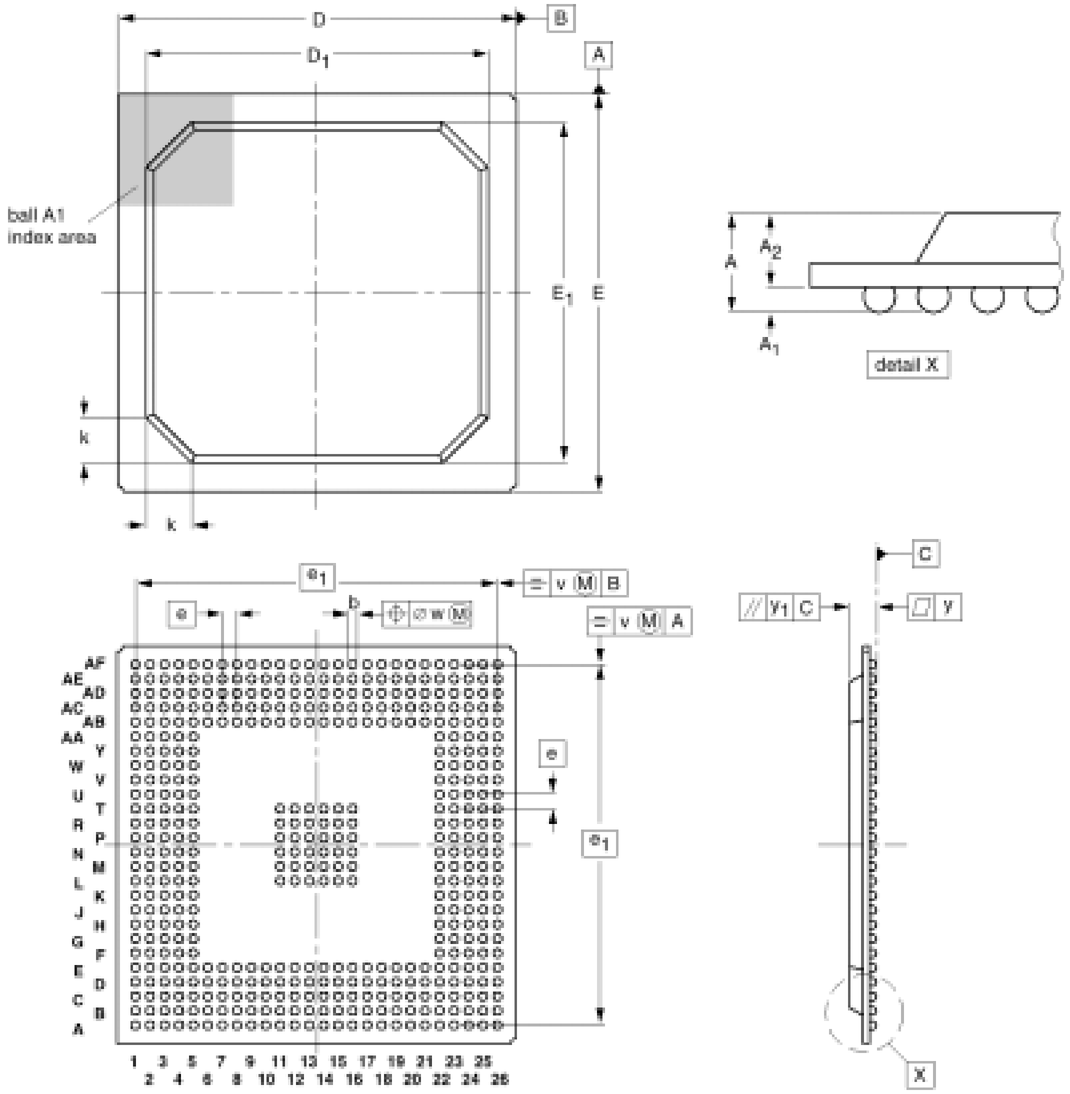


Figure 4.15 JTAG and SCAN Interface Timing Diagram

Notes on JTAG Interfaces Timing:

1. No Internal pull-up or pull-down on JTAG port or scan's TEST_EN pad cells.
2. A single 10k ohm pull-down can be tied to all 5 test pins (TEST_EN, TRST_N,TCK, TDI, TMS), when JTAG and scan testing are not used.

5.0 Mechanical Information



UNIT	A max.	A_1	A_2	b	D	D_1	E	E_1	e	e_1	k	v	w	y	y_1
mm	2.54	0.70 0.50	1.84 1.62	0.90 0.60	35.20 34.80	30.70 29.95	35.20 34.80	30.70 29.95	1.27	31.75	4.1 3.9	0.3	0.1	0.15	0.35

Figure 5.1 VSC9670 456-Pin BGA Packaging