PM5342

SPECTRA-155

SPECTRA-155 PROGRAMMER'S REFERENCE

APPLICATION NOTE

ISSUE 1: MARCH 1999

REVISION HISTORY

Issue No.	Issue Date	Details of Change
1	Mar 1999	Document created.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 REFERENCES

- Bell Communications Research, Generic Requirements GR-253-CORE – *SONET Transport Systems: Common Generic Criteria*, Issue 2, December 1995.
- ITU, Recommendation G.707 – *Network Node Interface For the Synchronous Digital Hierarchy*, 03/1996.
- ITU, Recommendation G.783 – *Characteristics of Synchronous Digital Hierarchy (SDH) Equipment Functional Blocks*, 04/1997.
- PMC-Sierra PMC-970133 – *PM5342 SPECTRA-155 SONET/SDH Payload Extractor Aligner Data Sheet*, Issue 5, August 1998.

2 BACKGROUND

PMC-Sierra's PM5342 SONET/SDH Payload Extractor/Aligner (SPECTRA-155) terminates the transport and path overhead of STS-1 (STM-0/AU-3) and STS-3/3c (STM-1/AU-3/AU-4) streams.

This document provides a convenient programmers reference for the SPECTRA-155. Although most of the information in this document can also be found in the SPECTRA datasheet, the information is distributed between pin, functional, and register descriptions. This document presents the same information in a reference form that spans functional blocks to quickly answer many software and hardware questions.

Although every effort has been taken to ensure that this document is consistent with the datasheet, some errors may occur. Where there are discrepancies with this document, the datasheet and datasheet errata (if any) will take precedence.

In this document, the term, "SPECTRA," is used interchangeably with the full product name, "SPECTRA-155."

3 ALARM, OVERHEAD, AND PAYLOAD INSERTION

The SPECTRA-155 is capable of modifying the outgoing transmitted and dropped datastreams in many different ways. Alarms, overhead, and payload are affected by register configuration, pin states, and by detected conditions such as AIS. The following sections summarize the many ways that transmitted and dropped datastreams may be affected by the SPECTRA.

SPECTRA is also capable of monitoring and reporting the status of the incoming received and added datastreams. The monitoring and reporting functions are *not* described in the following sections.

Descriptions of alarm, overhead, and payload insertion adhere to several conventions:

- The word "**Default**" in a table row indicates that the row corresponds to the power-up register configuration of the SPECTRA.

- Register bit names are prefixed with "r", and pin names are prefixed with "p". For example, "rFTPTR" refers to the FTPTR bit in register 131, 171 or 1B1h.

  Due to space limitations, register addresses cannot be listed in the table and must be determined from the datasheet. It is intended that this application note will be used in conjunction with a soft copy of the SPECTRA documentation (available from http://www.pmc-sierra.com) for performing text searches on register bit names.

  Note that the SPECTRA datasheet contains multiple instances of many bit names. In most cases, the duplication is due to replicated functional blocks. For example, there are six instances of the H4BYP bit: three in the RTAL blocks (register 128h,…) and three in the TTAL blocks (register 140h,…). The appropriate instance must be determined from context.

- A "n" replaces numbers in replicated register bits. For example, "rTPAIS_EN#n" could refer to the TPAIS_EN#1, TPAIS_EN#2, or TPAIS_EN#3, depending on the STS-1 (AU-3) of interest.

- Logical operations are indicated by the characters "!", "&", and "+", which represent logical NOT, AND, and OR, respectively.

- Braces indicate a detected condition. For example, {RX LOP-P} indicates the detection of loss-of-pointer in the receive datastream, while {RCP SENDLAIS} indicates that the SENDLAIS bit on the ring control port is active.

Abbreviations describe the source of the detected condition: ADD (add bus), RCP (ring control port), and RX (receive line interface).

3.1 Alarm Insertion

The following tables describe how the SPECTRA-155 can be configured to insert alarms in the transmit or drop datastreams (Table 1 and Table 2, respectively). Each alarm may be caused by several different conditions. Within the SPECTRA, all of the possible causes are OR'ed together—satisfying any one condition is sufficient to raise the alarm.

Table 3 lists all alarms and describes their effect on the datastream. In general, alarms have precedence over overhead insertion from any other source.

Table 1 - Transmit Alarm Insertion

Alarm	Condition	Comments
LOS	rDLOS	
AIS-L	rLAIS	
	!rRCP & pTLAIS	
	{RCP SENDLAIS} & rRCP & rRINGEN	
AIS-P	pDTPAIS & rTPAIS_EN#n	
	{ADD AIS-P} & rPAISPAIS	
	{ADD LOP-P} & rLOPPAIS	
	{ADD AISC-P} & rPAISCONPAIS	
	{ADD LOPC-P} & rLOPCONPAIS	
	rPAIS	
	{TTAL elastic store error} & rESAIS	AIS Asserted for 3 Frames
	{ADD ISF} & rTTCTE	Affected by rFISF
AIS-V	{ADD LOM} & rLOMTUAIS	
	rITUAIS	
AIS-V (DS-3)	pSMODE[2:0]=110 & (pDS3TAISn + rDS3AISGEN)	

Table 2 - Drop Alarm Insertion

Alarm	Condition	Comments
AIS-P (via RLOP)	!rRCP & pRLAIS	
	{RX AIS-L} & rALLONES	

Alarm	Condition	Comments
AIS-P (cont'd)	{RX TIU-S} & rALLONES & rRTIUINS & rAUTOSTIU1	Mode 1 TIU-S
	{RX TIU-S} & rALLONES & rRTIUINS & rAUTOSTIU2	Mode 2 TIU-S
	{RX TIM-S} & rALLONES & rRTIMINS	
	{RX LOS} & rALLONES & rLOSINS	
	{RX LOF} & rALLONES & rLOFINS	
	{RX SF} & rALLONES & rSFINS	
	{RX SD} & rALLONES & rSDINS	
AIS-P	pDTPAISn & !rTPAIS_EN	
	{RX AIS-P}	
	{RX LOP-P} & rLOPPAIS	**Default**
	{RX AISC-P} & rPAISCONPAIS	
	{RX LOPC-P} & rLOPCONPAIS	
	{RX TIM-P} & rTIMPAIS	**Default**
	{RX TIU-P} & rTIUPAIS & rAUTOPTIU1#n	Mode 1 TIU-P
	{RX TIU-P} & rTIUPAIS & rAUTOPTIU2#n	Mode 2 TIU-P
	{RX SLM-P} & rPSLMPAIS	**Default**
	{RX SLU-P} & rPSLUAIS	**Default**
	rIPAIS	
	{RTAL elastic store error} & rESAIS	AIS Asserted for 3 Frames
	{RX_ISF} & rTTCTE	Affected by rFISF
AIS-TC	rOTCTE & rISF	
AIS-V	{RX LOM} & rLOMTUAIS	**Default**
	rITUAIS	
AIS-V (DS3)	pDS3RAIS	
	rDS3AISGEN	
	{RX TIM-P} & rTIMDS3AIS	
	{RX TIU-P} & rTIUDS3AIS & rAUTOPTIU1#n	Mode 1 TIU-P
	{RX TIU-P} & rTIUDS3AIS & rAUTOPTIU2#n	Mode 2 TIU-P
	{RX LOP-P} & rLOPDS3AIS	
	{RX SLM-P} & rPSLMDS3AIS	
	{RX SLU-P} & rPSLUDS3AIS	
	({RX LOS} + {RX LOF} + {RX AIS-L}) & rALMDS3AIS	

Table 3 - Alarm Descriptions

Alarm	Description
AIS-L	Line AIS (MS-AIS). All ones in line overhead, H1 H2 pointer, path overhead, and payload.
AIS-P	Path AIS (HP-AIS). All ones in H1 H2 pointer, path overhead, and payload.
AIS-TC	Tandem connection AIS. All ones in payload. IEC field in Z5 byte indicates ISF (1111b). Tandem connection data link remains active. Error-free path BIP inserted in the B3 byte.
AIS-V	Tributary AIS. All ones in payload (but not path overhead). Effectively, this appears as AIS for all tributaries.
AIS-V (DS-3)	Tributary AIS. All ones in payload for the affected DS-3. This type of AIS is only applicable when the SPECTRA is operating in DS-3 mode.
LOS	Loss-of-signal. Transmit all zeros.

3.2 Overhead Insertion

SPECTRA is capable of modifying the overhead of the transmitted and dropped datastreams in many different ways. The following tables describe how transport and path overhead bytes are inserted into the transmitted data stream (Table 4) or the dropped data stream (Table 6). For each overhead byte location, the tables describe the inserted value and the condition or conditions under which that value will be inserted. See the Appendix for a brief overview of overhead byte functions.

In some overhead bytes, bits are allocated for specific functions. For example, bits 1-4 of the G1 path overhead byte are allocated for the REI-P function. Overhead insertion functions that apply only to allocated bits are indicated separately in the tables. If the table does not identify any allocated bits, then the overhead insertion function applies to the entire overhead byte.

Values inserted into overhead bytes may take many forms: constants (e.g. “F6h” or “110b”); register contents (e.g. “reg 1Ah”); all ones or all zeros, according to the value of a single register bit or pin (e.g. “rUNUSED_V”); a default value that has been XORed with an error mask (e.g. “mask FFh”); or streamed in from a serial overhead interface pin (e.g. “pTTOH”). Some special bytes may have other

values inserted into them; refer to the bit descriptions in the SPECTRA datasheet for more information.

Overhead insertion conditions are generally listed in order of increasing priority. The bottom-most insertion condition usually will take priority over any conditions above it. For example, in the transmit direction, if the TTOHEN pin is active then the D1-D3 overhead bytes will be inserted from the TTOH pin, independent of the state of any other register bits or pins. Some exceptions to this rule occur for the more complicated overhead bytes, such as G1. In these cases, insertion conditions are expanded to describe the full logical hierarchy of insertion values.

Some overhead insertion conditions are left blank, indicating that the value will be inserted if all other insertion conditions are false. For example, in the transmit direction, B1 will be calculated based on the previous frame's data if register bit DBIP8 and pin TTOHEN are both false.

Table 4 - Transmit Overhead Insertion

Overhead Byte	Bit(s)	Value	Condition	Comments
A1, A2		F6h, 28h	!rDFP	**Default**
		76h, 28h	rDFP	
		pTTOH	pTTOHEN	
J0		01h	!rSTEN	**Default**
		section trace buffer	rSTEN	
		pTTOH	pTTOHEN & !rSTEN	
Z0		02h, 03h	!rZ0INS	**Default**
		reg 1Bh	rZ0INS	
		pTTOH	pTTOHEN	
B1		calculate		**Default**
		mask FFh	rDBIP8	Error Mask
		mask pTTOH	pTTOHEN	Error Mask
E1		pTSOW		**Default**
		pTOH	rTOHSEL[2:0]=000	
		pTTOH	pTTOHEN	**Default**
F1		pTSUC		
		pTOH	rTOHSEL[2:0]=001	
		pTTOH	pTTOHEN	
D1-D3		rTDLVAL	rTDLSEL	

Overhead Byte	Bit(s)	Value	Condition	Comments
D1-D3		pTSLD	!rTDLSEL	**Default**
(cont'd)		pTTOH	pTTOHEN	
H1, H2	pointer	calculate		**Default**
		+ justification	rDOPJ1 & !rDOPJ0	Force justification every 4th frame.
		- justification	!rDOPJ1 & rDOPJ0	Force justification every 4th frame.
		reg 136h, 135h,…	rFTPTR	Pointer substitution without NDF.
	NDF	reg 136h,…	{TTAL discontinuous pointer change} + rNDF	
	SS	reg 136h,…		**Default**
		mask pTTOH	pTTOHEN	Error Mask
H1, H2 Concat		reg 13Fh, 13Eh	rTMODE[1:0]=11	
		mask pTTOH	pTTOHEN	Error Mask
H3		insert		**Default**
		pTTOH	pTTOHEN	
B2		calculate		**Default**
		mask FFh	rDB2	Error Mask
		mask pTTOH	pTTOHEN	Error Mask
K1, K2		ring control port	!rTAPSTAP & rRCP	filtered
		reg 06h, 07h	!rTAPSTAP & rAPSREG	
		pTOH	!rTAPSTAP & !rAPSREG & rTOHSEL[2:0]=100	**Default**
		pTTOH	!rTAPSTAP & pTTOHEN	
		pTAD	rTAPSTAP	
	RDI-L	110b	rLRDI	
		110b	pTLRDI & !rRCP	
		110b	{RX AIS-L} & !rRCP & rLAISINS	**Default**
		110b	{RX TIU-S} & !rRCP & rRTIUINS & rAUTOSTIU1	Mode 1 TIU-S
		110b	{RX TIU-S} & !rRCP & rRTIUINS & rAUTOSTIU2	Mode 2 TIU-S
		110b	{RX TIM-S} & !rRCP & rRTIMINS	

Overhead Byte	Bit(s)	Value	Condition	Comments
K1, K2 (cont'd)	RDI-L (cont'd)	110b	{RX LOS} & !rRCP & rLOSINS	**Default**
		110b	{RX LOF} & !rRCP & rLOFINS	**Default**
		110b	{RX_SF} & !rRCP & rSFINS	
		110b	{RX SD} & !rRCP & rSDINS	
		110b	RCP_SENDLRDI & rRCP & rRINGEN	
D4-D12		pTLD	!rTDLSEL	**Default**
		pTSLD	rTDLSEL	
		pTTOH	pTTOHEN	
S1		reg 1Ah		**Default**
		pTTOH	pTTOHEN	
Z1		00h	!rUNUSED_EN	**Default**
		rUNUSED_V	rUNUSED_EN	
		pTTOH	pTTOHEN	
Z2		00h	!rUNUSED_EN	**Default**
		rUNUSED_V	rUNUSED_EN	
		pTTOH	pTTOHEN	
M0,M1	REI-L	00h	!rAUTOLREI & !rRINGEN	**Default**
		RCP LREI	rRCP & rRINGEN	
		RX B2 count	rAUTOLREI & !rRINGEN & !rBLOCKBIPO	
		RX B2	rAUTOLREI & !rRINGEN & rBLOCKBIPO	
		pTTOH	pTTOHEN	
E2		pTLOW		**Default**
		pTOH	rTOHSEL[2:0]=010	
		pTTOH	pTTOHEN	
TOH Unused (not Z1, Z2)		00h	!rUNUSED_EN	**Default**
		rUNUSED_V	rUNUSED_EN	
		pTTOH	pTTOHEN	
TOH National (except Z0)		00h	!rNAT_EN	**Default**
		rNAT_V	rNAT_EN	
		pTTOH	pTTOHEN	
J1		reg 137h,…	!rTDIS & !rTPTBnEN	**Default**
		SPTB	!rTDIS & rTPTBnEN	

Overhead Byte	Bit(s)	Value	Condition	Comments
J1 (cont'd)		ADD passthrough	rTDISn & !rADDUEn	
		rADDUEV	rTDISn & rADDUEn	
		pTPOH	!rTPTBEN & pTPOHEN	
B3		calculated	!rTDISn	**Default.** Affected by rEXCFS.
		calculated w.r.t. Z5	!rTDISn & rTTCTE	Affected by rEXCFS.
		mask FFh	!rTDISn & rDB3	Error Mask
		ADD passthrough	rTDISn & !rADDUEn	
		rADDUEV	rTDISn & rADDUEn	
		pTPOH	rTDIS & pTPOHEN	Corrupts B3 unpredictably.
		mask pTPOH	!rTDIS & pTPOHEN	Error Mask
C2		reg 138h,…	!rTDISn	**Default**
		ADD passthrough	rTDISn & !rADDUEn	
		rADDUEV	rTDISn & rADDUEn	
		pTPOH	pTPOHEN	
G1		ADD passthrough	rTDISn & !rADDUEn	
		rADDUEV	rTDISn & rADDUEn	
		pTPOH	pTPOHEN	
	REI-P	0000b		**Default**
		rPREI[3:0]	!rTDISn & !pTPOHEN	Inserts REI count in next available G1.
		RX B3 count	!rTDISn & rRXSEL[1:0]=00 & rAUTOPREI & !rBLKBIPO	
		RX B3	!rTDISn & rRXSEL[1:0]=00 & rAUTOPREI & rBLKBIPO	
		pTAD	!rTDISn & rRXSEL[1:0]=01	
		ADD passthrough	!rTDISn & rRXSEL[1:0]=10	
		0000b	!rTDISn & rRXSEL[1:0]=11	
	RDI-P	RPOP†	rEPRDIEN & rEPRDISRC & rRXSEL[1:0]=00	
		pTAD	rEPRDIEN & rEPRDISRC & rRXSEL[1:0]=01	
		ADD passthrough	rEPRDIEN & rRXSEL[1:0]=10	

Overhead Byte	Bit(s)	Value	Condition	Comments
G1 (cont'd)	RDI-P (cont'd)	000b	rEPRDIEN & rEPRDISRC & rRXSEL[1:0]=11	
		pTPOH	pTPOHEN & !rEPRDIEN	
	RDI-P (bit 5)	rPRDI	(!pTPOHEN & !rEPRDIEN & rRXSEL[1:0]=11) + (rEPRDIEN & !rEPRDISRC & !(rRXSEL[1:0]=10))	
		rPRDI + RPOP†	!pTPOHEN & !rEPRDIEN & rRXSEL[1:0]=00	**Default**
		rPRDI + pTAD	!pTPOHEN & !rEPRDIEN & rRXSEL[1:0]=01	
		rPRDI + ADD passthrough	!pTPOHEN & !rEPRDIEN & rRXSEL[1:0]=10	
	RDI-P (bit 6)	rEPRDI6	(!pTPOHEN & !rEPRDIEN) + (rEPRDIEN & !rEPRDISRC & !(rRXSEL[1:0]=10))	
	RDI-P (bit 7)	rEPRDI7	(!pTPOHEN & !rEPRDIEN) + (rEPRDIEN & !rEPRDISRC & !(rRXSEL[1:0]=10))	
	bit 8	rG1[0]	!pTPOHEN	**Default**
F2		reg 13Ah,…	!rTDISn	**Default**
		ADD passthrough	rTDISn & !rADDUEn	
		rADDUEV	rTDISn & rADDUEn	
		pTPOH	pTPOHEN	
H4		TTAL	!rTDISn & !rH4BYP	**Default**
		mask FFh	!rTDISn & rDH4	Error Mask
		ADD passthrough	rTDISn & rH4BYP & !rADDUEn	
		rADDUEV	rTDISn & rH4BYP & rADDUEn	
		mask pTPOH	pTPOHEN	Error Mask
F3/Z3		reg 13Bh,…	!rTDISn	**Default**
		ADD passthrough	rTDISn & !rADDUEn	
		rADDUEV	rTDISn & rADDUEn	
		pTPOH	pTPOHEN	
K3/Z4		reg 13Ch,…	!rTDISn	**Default**
		ADD passthrough	rTDISn & !rADDUEn	
		rADDUEV	rTDISn & rADDUEn	
		pTPOH	pTPOHEN	

Overhead Byte	Bit(s)	Value	Condition	Comments
N1/Z5		reg 13Dh,…	!rTDISn	**Default**
		ADD passthrough	rTDISn & !rADDUEn	
		rADDUEV	rTDISn & rADDUEn	
	IEC	0000b	rTTCTE	
	TCDL	0000b	rTTCTE	
		pTPOH	pTPOHEN	

†The RPOP will source RDI-P values according to the consequent actions that are configured according to Table 5.

Table 5 - Consequent RDI-P Insertion

RDI-P Value (G1 bits 5-7)	Condition	Comment
1xxb	{RX TIM-P} & rTIMPRDI	**Default**
1xxb	{RX TIU-P} & rAUTOPTIU1#n rTIUPRDI	Mode 1 TIU-P
1xxb	{RX TIU-P} & rAUTOPTIU2#n & rTIUPRDI	Mode 2 TIU-P
1xxb	{RX LOP-P} & rLOPPRDI	**Default**
1xxb	{RX LOPC-P} & rLOPCONPRDI	**Default**
1xxb	{RX SLM-P} & rPSLMPRDI	**Default**
1xxb	{RX AIS-P} & rPAISPRDI	**Default**
1xxb	{RX AISC-P} & rPAISCONPRDI	**Default**
1xxb	{RX UNEQ-P} & rUNEQPRDI	**Default**
1xxb	({RX LOS} + {RX LOF} + {RX AIS-L}) & rALMPRDI	
x10b	{RX TIM-P} & rTIMEPRDI & rEPRDI_EN	
x10b	{RX TIU-P} & rAUTOPTIU1#n & rTIUEPRDI & rEPRDI_EN	Mode 1 TIU-P
x10b	{RX TIU-P} & rAUTOPTIU2#n & rTIUEPRDI & rEPRDI_EN	Mode 2 TIU-P
x01b	{RX LOP-P} & rNOLOPEPRDI & rEPRDI_EN	
x01b	{RX LOPC-P} & rNOLOPCONEPRDI & rEPRDI_EN	
x10b	{RX SLM-P} & rPSLMEPRDI & rEPRDI_EN	
x01b	{RX AIS-P} & rNOPAISEPRDI & rEPRDI_EN	
x01b	{RX AISC-P} & rNOPAISCONEPRDI & rEPRDI_EN	
x10b	{RX UNEQ-P} & rUNEQEPRDI & rEPRDI_EN	
x01b	({RX LOS} + {RX LOF} + {RX AIS-L}) & rNOALMEPRDI &	

RDI-P Value (G1 bits 5-7)	Condition	Comment
	rEPRDI_EN	

Notes:

- When no error conditions are present, the RPOP will generate a non-enhanced RDI-P code of 0xxb and an enhanced RDI-P code of x01b.
- The assertion time for all of the above RDI-P consequent actions is affected by the PERSIST register bit (reg 130, 170, 1B0h).
- Hierarchy of enhanced RDI-P not shown. When multiple enhanced RDI-P conditions occur simultaneously, the "x01" indication has priority over "x10".

Table 6 - Drop Overhead Insertion

Overhead Byte	Bit(s)	Value	Condition	Comments
A1, A2		00h		**Default**
J0		00h		**Default**
Z0		00h		**Default**
B1		00h		**Default**
E1		00h		**Default**
F1		00h		**Default**
D1-D3		00h		**Default**
H1, H2	pointer	calculate		**Default**
		+ justification	rDOPJ1 & !rDOPJ0	Force justification every 4th frame.
		- justification	!rDOPJ1 & rDOPJ0	Force justification every 4th frame.
		hold last value	rOTCTE & ({RX LOP} + {RX AIS})	
	NDF	1001b	rDLOP	
	SS	00b	!rSSS	
		10b	rSSS	**Default**
H3		insert		**Default**
B2		00h		**Default**

Overhead Byte	Bit(s)	Value	Condition	Comments
K1		00h		**Default**
K2		00h		**Default**
D4-D12		00h		**Default**
S1		00h		**Default**
Z1		00h		**Default**
Z2		00h		**Default**
M0		00h		**Default**
M1		00h		**Default**
E2		00h		**Default**
TOH Unused		00h		**Default**
TOH National		00h		**Default**
J1		RX passthrough		**Default**
B3		RX passthrough	!rTTCTE	**Default**
		calculated	rTTCTE	Reflects changes made to Z5.
C2		RX passthrough		**Default**
G1		RX passthrough		**Default**
F2		RX passthrough		**Default**
H4		RX passthrough	rH4BYP & pSMODE[2:0]=[000, 001, 010]	
		00h	rH4BYP & pSMODE[2:0]=[011, 100, 101, 110]	
		FC, FD, FE, FFh	!rH4BYP	**Default**
		invert	rDH4	Force inverted H4.
		FFh	rH4AISB & {RX AIS-P}	
F3/Z3		RX passthrough		**Default**
K3/Z4		RX passthrough		**Default**
N1/Z5	IEC	RX passthrough	!rOTCTE & !rTTCTE	**Default**
		0000b	!rOTCTE & rTTCTE	
		RX B3 count	rOTCTE & !rTTCTE	
		1111b	rOTCTE & !rTTCTE & (rISF + {RX LOP-P} + {RX AIS-P})	
		????	rOTCTE & rTTCTE	Illegal

Overhead Byte	Bit(s)	Value	Condition	Comments
	TCDL	RX passthrough	!rOTCTE & !rTTCTE	**Default**
		0000b	!rOTCTE & rTTCTE	
		1111b	rOTCTE & !rTTCTE & !rTCDLT	
		RX passthrough	rOTCTE & !rTTCTE & rTCDLT	
N1/Z5	TCDL	????	rOTCTE & rTTCTE	Illegal
(cont'd)	(cont'd)	pRTCOHn	pRTCENn	

3.3 Telecom Payload Processing

In telecom bus mode, the SPECTRA-155 has limited capability to affect the payload data. Transmit and drop payload processing are described in Table 7 and Table 8, respectively.

Table 7 - Transmit Payload Processing

Payload Location	Value	Condition	Comment
Fixed Stuff	00h	rCLRFS	
	ADD passthrough	!rCLRFS	**Default**
	00h	(pSMODE[2:0]=[011, 100] & !rTDM_FSEN) + (pSMODE[2:0]=101 & rGAPFS)	
	data	pSMODE[2:0]=[011, 100] & rTDM_FSEN	
	DS-3 data	pSMODE[2:0]=101 & !rGAPFS	
STS-1/AU-3 Payload	ADD passthrough	!rADDUEn	
	rADDUEV	rADDUEn	

Table 8 - Drop Payload Processing

Payload Location	Value	Condition	Comment
Fixed Stuff	00h	rCLRFS	
	RX passthrough	!rCLRFS	**Default**
	00h	(pSMODE[2:0]=[011, 100, 101] & !rRDMFSEN) + (pSMODE[2:0]=110 & rGAPFS))	
	data	pSMODE[2:0]=[011, 100, 101] & rRDMFSEN	

Payload Location	Value	Condition	Comment
Fixed Stuff (cont'd)	DS-3 data	pSMODE[2:0]=110 & !rGAPFS	

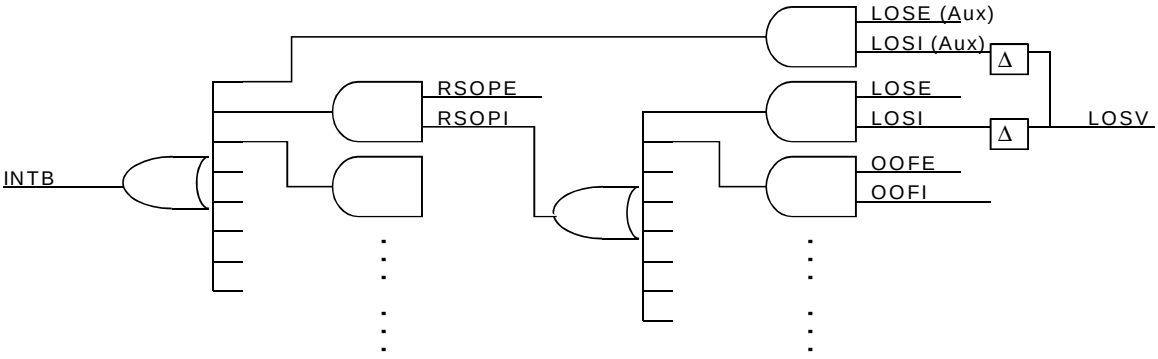
4 INTERRUPTS

The SPECTRA-155 has a single interrupt pin (INTB) which can reflect the status of over 250 independent interrupt sources. This section describes the interrupt hierarchy and then provides a quick cross-reference of all SPECTRA interrupt sources.

4.1 Interrupt Overview

Figure 1 shows a portion of the interrupt generation logic. This diagram shows the contribution of the loss-of-signal (LOS) alarm to interrupt generation.

Figure 1 - Interrupt Generation Logic



At the root of each interrupt is an interrupt source. Often, the status of the interrupt source will be reflected with a value bit, or "V" bit. For example, the LOSV bit (reg 011h) reflects the current status of the loss-of-signal detector. When LOS is detected, the LOSV bit will be 1; otherwise, LOSV will be 0. The "V" bits are "real time" and are not latched. Many interrupt sources do not have "V" bits but are reported in other ways.

4.1.1 Interrupts and Interrupt Enables

Every interrupt source has an associated interrupt status bit, or "I" bit. The "I" bit is set and latched when the interrupt event occurs, and will be cleared only after processor intervention. Depending on the "I" bit, the interrupt will be cleared either after a CPU read from the register, or after a CPU write of "1" to the interrupt bit. In Figure 1, the LOSI bit (reg 011h) is the interrupt status bit associated with the LOS detector. When the LOS detector changes state (i.e. when LOSV changes value), LOSI will be set. LOSI will be cleared only after the CPU reads from register 011h.

To filter irrelevant interrupts, every "I" bit has an associated interrupt enable bit, or "E" bit. When the "E" bit is set, the associated "I" bit will contribute to the INTB pin or to a top level interrupt status bit (more about top-level interrupts later). When the "E" bit is cleared, the associated "I" bit will have no effect on INTB or top-level interrupts. Note that the "E" bit does not affect the behaviour of the "I" bit itself. For example, the LOSE bit (reg 010h) is the interrupt enable for LOSI. If LOSE is cleared, the LOSI bit will continue to indicate changes in the LOSV bit but will never contribute to an INTB interrupt.

4.1.2 Top-Level Interrupts

Most interrupts are grouped under one of several top-level interrupt status bits. These top level "I" bits represent the logical "or" of all enabled interrupts in that group, and allow software to monitor multiple interrupts with a single CPU read. For example, the LOSI interrupt is grouped under the RSOPI top-level interrupt (reg 002h) along with all other RSOP interrupts. RSOPI will assert whenever LOSI and LOSE are both set (or OOFI and OOFE, or …). Top level interrupts cannot be cleared directly; instead, they will clear when the underlying interrupts are cleared.

Some top-level interrupts status bits have an associated enable bit. These bits are identical to the "E" bits for other interrupts. That is, the top-level interrupt will only contribute to INTB if the enable bit is set. As with other "E" bits, the interrupt enable bits for top level interrupts don't change the value of the "I" bit itself. Top-level interrupts without "E" bits always contribute to INTB.

4.1.3 Auxiliary Interrupts

A final type of interrupt is the auxiliary interrupt. Auxiliary interrupts provide a second "I" bit for many commonly used interrupts. The second "I" bit is useful when the SPECTRA-155 is used with software architectures that combine polling and interrupt service routines. With the "clear-on-read" interrupt bits, software can inadvertently clear one or more interrupts by reading from a register containing the "I" bits. The auxiliary interrupts provide a second "clear-on-write" interrupt bit that is easier to deal with in software.

As shown in Figure 1, the auxiliary interrupts have a separate enable bit from the main interrupt. Auxiliary interrupts are also cleared independently of the main interrupt. For example, clearing the LOFI interrupt (reg 011h) has no effect on the auxiliary LOFI interrupt (reg 0E6h). All auxiliary interrupts contribute directly to INTB, without going through a top-level interrupt.

4.2 Interrupt Cross-Reference

Table 9 shows the entire SPECTRA-155 interrupt hierarchy, including interrupt status bits, auxiliary interrupts, and interrupt enables. The format of Table 9 is described in Figure 2.

Figure 2 - Interrupt Cross-Reference Legend

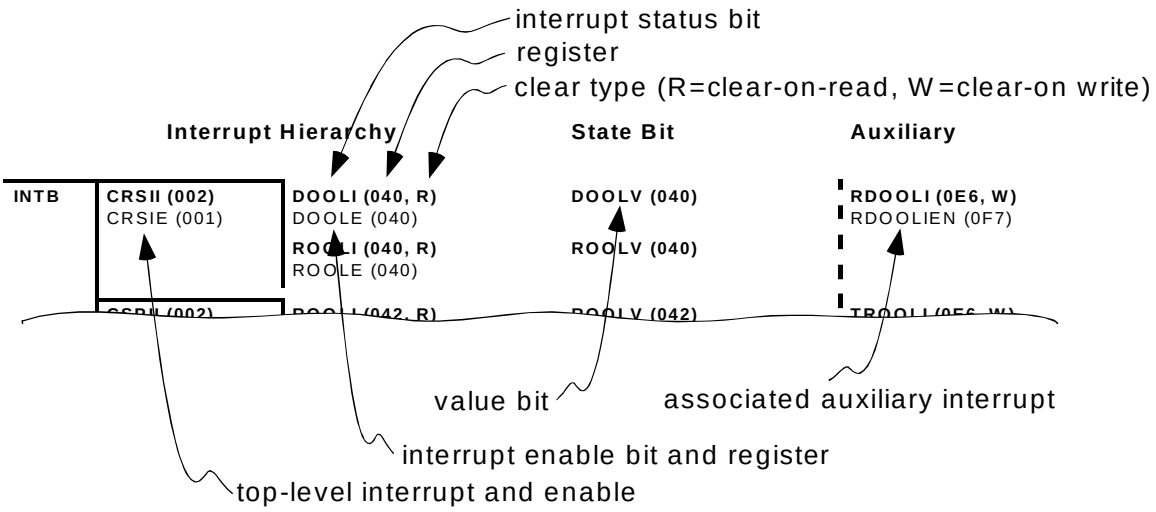


Table 9 - Interrupt Cross-Reference

	Interrupt Hierarchy		State Bit	Auxiliary
INTB	**CRSII (002)**	**DOOLI (040, R)**	**DOOLV (040)**	**RDOOLI (0E6, W)**
	CRSIE (001)	DOOLE (040)		RDOOLIEN (0F7)
		ROOLI (040, R)	**ROOLV (040)**	
		ROOLE (040)		
	CSPII (002)	**ROOLI (042, R)**	**ROOLV (042)**	**TROOLI (0E6, W)**
	CSPIE (001)	ROOLE (042)		TROOLIEN (0F7)
	RASEI (002)	**COAPSI (021, R)**		
	RASEE (001)	COAPSE (020)		
		PSBFI (021, R)	**PSBFV (021)**	
		PSBFE (020)		
		SDBERI (021, R)	**SDBERV (021)**	
		SDBERE (020)		
		SFBERI (021, R)	**SFBERV (021)**	
		SFBERE (020)		

Interrupt Hierarchy		State Bit	Auxiliary
RASEI (cont'd)	**Z1/S1I (021, R)**		
	Z1/S1E (020)		
RLOPI (002)	**BIPEI (009, R)**		
RLOPE (001)	BIPEE (009)		
	LAISI (009, R)	**LAISV (008)**	**LAISI (0E6, W)**
	LAISE (009)		LAISIEN (0F7)
	LRDII (009, R)	**LRDIV (008)**	**LRDII (0E6, W)**
	LRDIE (009)		LRDIIEN (0F7)
	LREII (009, R)		
	LREIE (009)		
RSOPI (002)	**BIPEI (011, R)**		
RSOPE (001)	BIPEE (010)		
	LOFI (011, R)	**LOFV (011)**	**LOFI (0E6, W)**
	LOFE (010)		LOFIEN (0F7)
	LOSI (011, R)	**LOSV (011)**	**LOSI (0E6, W)**
	LOSE (010)		LOSIEN (0F7)
	OOFI (011, R)	**OOFV (011)**	**OOFI (0E6, W)**
	OOFE (010)		OOFIEN (0F7)
SSTBI (002)	**RTIMI (039, R)**	**RTIMV (039)**	
SSTBE (001)	RTIMIE (038)		
	RTIUI (039, R)	**RTIUV (039)**	
	RTIUIE (038)		
	SCPIFI[0] (01F, W)	**SCPIV[0] (01F)**	
	SCPIFE[0] (015)		
	SCPIFI[1] (01F, W)	**SCPIV[1] (01F)**	
	SCPIFE[1] (015)		
	SCPIRI[0] (01F, W)	**SCPIV[0] (01F)**	
	SCPIRE[0] (015)		
	SCPIRI[1] (01F, W)	**SCPIV[1] (01F)**	
	SCPIRE[1] (015)		
	ABUS_SYNCI (07B, R)	**ABUS_SYNCV (07A)**	
	ABUS_SYNCE (07A)		
	DBUS_SYNCI (07B, R)	**DBUS_SYNCV (079)**	
	DBUS_SYNCE (079)		
	PTIU2I#1 (0EB, R)	**PTIU2V#1 (0EC)**	**TIU2I#1 (0EA, W)**
	PTIU2IE#1 (0EB)		TIU2I#1EN (0FB)
	PTIU2I#2 (0EB, R)	**PTIU2V#2 (0EC)**	**TIU2I#2 (0EA, W)**
	PTIU2IE#2 (0EB)		TIU2I#2EN (0FB)

Interrupt Hierarchy		State Bit	Auxiliary
	PTIU2I#3 (0EB, R) PTIU2IE#3 (0EB)	**PTIU2V#3 (0EC)**	**TIU2I#3 (0EA, W)** TIU2I#3EN (0FB)
	STIU2I (0EB, R) STIU2IE (0EB)	**STIU2V (0EC)**	
D3MA[3:1]I (103)	**OFLI (095, R)** OFLIEN (096)		
	UFLI (095, R) UFLIEN (096)		
	OFLI (09D, R) OFLIEN (09B)		
	UFLI (09D, R) UFLIEN (09B)		
	OFLI (0A5, R) OFLIEN (0A6)		
	UFLI (0A5, R) UFLIEN (0A6)		
D3MD[3:1]I (103)	**OFLI (091, R)** OFLIEN (092)		
	UFLI (091, R) UFLIEN (092)		
	OFLI (099, R) OFLIEN (09A)		
	UFLI (099, R) UFLIEN (09A)		
	OFLI (0A1, R) OFLIEN (0A2)		
	UFLI (0A1, R) UFLIEN (0A2)		
TPIP1I (103)	**NEWPTRI (0B0, R)** NEWPTRE (0B0)		
	BIPEI (0B1, R) BIPEE (0B3)		
	LOPCONI (0B1, R) LOPCONE (0B3)	**LOPCONV (0B0)**	
	LOPI (0B1, R) LOPE (0B3)	**LOPV (0B0)**	
	PAISCONI (0B1, R) PAISCONE (0B3)	**PAISCONV (0B0)**	
	PAISI (0B1, R) PAISE (0B3)	**PAISV (0B0)**	

Interrupt Hierarchy		State Bit	Auxiliary
TPIP1I (cont'd)	**PREII (0B1, R)**		
	PREIE (0B3)		
	CONCATI (0B2, R)		
	CONCATE (0B4)		
	DISCOPAI (0B2, R)		
	DISCOPAE (0B4)		
	ILLJREQI (0B2, R)		
	ILLJREQE (0B4)		
	INVNDFI (0B2, R)		
	INVNDFE (0B4)		
	NDFI (0B2, R)		
	NDFE (0B4)		
	NSEI (0B2, R)		
	NSEE (0B4)		
	PSEI (0B2, R)		
	PSEE (0B4)		
	COMAI (0BC, R)		
	COMAE (0BC)		
	LOMI (0BC, R)	**LOMV (0BC)**	
	LOME (0BC)		
TPIP2I (103)	**NEWPTRI (0C0, R)**		
	NEWPTRE (0C0)		
	BIPEI (0C1, R)		
	BIPEE (0C3)		
	LOPCONI (0C1, R)	**LOPCONV (0C0)**	
	LOPCONE (0C3)		
	LOPI (0C1, R)	**LOPV (0C0)**	
	LOPE (0C3)		
	PAISCONI (0C1, R)	**PAISCONV (0C0)**	
	PAISCONE (0C3)		
	PAISI (0C1, R)	**PAISV (0C0)**	
	PAISE (0C3)		
	PREII (0C1, R)		
	PREIE (0C3)		
	CONCATI (0C2, R)		
	CONCATE (0C4)		
	DISCOPAI (0C2, R)		
	DISCOPAE (0C4)		
	ILLJREQI (0C2, R)		
	ILLJREQE (0C4)		

Interrupt Hierarchy		State Bit	Auxiliary
TPIP2I (cont'd)	**INVNDFI (0C2, R)**		
	INVNDFE (0C4)		
	NDFI (0C2, R)		
	NDFE (0C4)		
	NSEI (0C2, R)		
	NSEE (0C4)		
	PSEI (0C2, R)		
	PSEE (0C4)		
	COMAI (0CC, R)		
	COMAE (0CC)		
	LOMI (0CC, R)	**LOMV (0CC)**	
	LOME (0CC)		
TPIP3I (103)	**NEWPTRI (0D0, R)**		
	NEWPTRE (0D0)		
	BIPEI (0D1, R)		
	BIPEE (0D3)		
	LOPCONI (0D1, R)	**LOPCONV (0D0)**	
	LOPCONE (0D3)		
	LOPI (0D1, R)	**LOPV (0D0)**	
	LOPE (0D3)		
	PAISCONI (0D1, R)	**PAISCONV (0D0)**	
	PAISCONE (0D3)		
	PAISI (0D1, R)	**PAISV (0D0)**	
	PAISE (0D3)		
	PREII (0D1, R)		
	PREIE (0D3)		
	CONCATI (0D2, R)		
	CONCATE (0D4)		
	DISCOPAI (0D2, R)		
	DISCOPAE (0D4)		
	ILLJREQI (0D2, R)		
	ILLJREQE (0D4)		
	INVNDFI (0D2, R)		
	INVNDFE (0D4)		
	NDFI (0D2, R)		
	NDFE (0D4)		
	NSEI (0D2, R)		
	NSEE (0D4)		
	PSEI (0D2, R)		
	PSEE (0D4)		

Interrupt Hierarchy		State Bit	Auxiliary
TPIP3I (cont'd)	**COMAI (0DC, R)**		
	COMAE (0DC)		
	LOMI (0DC, R)	**LOMV (0DC)**	
	LOME (0DC)		
RPOP1I (104)	**NEWPTRI (110, R)**		
	NEWPTRE (110)		
	BIPEI (111, R)		
	BIPEE (113)		
	ERDII (111, R)	**ERDIV[2:0] (110)**	**EPRDII#1 (0EA, W)**
	ERDIE (113)		EPRDII#1EN (0FB)
	LOPCONI (111, R)	**LOPCONV (110)**	**LOPCONI (0EA, W)**
	LOPCONE (113)		LOPCONIEN (0FB)
	LOPI (111, R)	**LOPV (110)**	**LOPI (0E7, W)**
	LOPE (113)		LOPIEN (0F8)
	PAISCONI (111, R)	**PAISCONV (110)**	**PAISCONI (0EA, W)**
	PAISCONE (113)		PAISCONIEN (0FB)
	PAISI (111, R)	**PAISV (110)**	**PAISI (0E7, W)**
	PAISE (113)		PAISEN (0F8)
	PRDII (111, R)	**PRDIV (110)**	**PRDII (0E7, W)**
	PRDIE (113)		PRDIEN (0F8)
	PREII (111, R)		
	PREIE (113)		
	CONCATI (112, R)		
	CONCATE (114)		
	DISCOPAI (112, R)		
	DISCOPAE (114)		
	ILLJREQI (112, R)		
	ILLJREQE (114)		
	INVNDFI (112, R)		
	INVNDFE (114)		
	NDFI (112, R)		
	NDFE (114)		
	NSEI (112, R)		
	NSEE (114)		
	PSEI (112, R)		
	PSEE (114)		
	COMAI (11C, R)		
	COMAE (11C)		
	LOMI (11C, R)	**LOMV (11C)**	**LOMI (0E7, W)**
	LOME (11C)		LOMIEN (0F8)

Interrupt Hierarchy		State Bit	Auxiliary
RPOP2I (104)	**NEWPTRI (150, R)**		
	NEWPTRE (150)		
	BIPEI (151, R)		
	BIPEE (153)		
	ERDII (151, R)	**ERDIV[2:0] (150)**	**EPRDII#2 (0EA, W)**
	ERDIE (153)		EPRDII#2EN (0FB)
	LOPCONI (151, R)	**LOPCONV (150)**	
	LOPCONE (153)		
	LOPI (151, R)	**LOPV (150)**	**LOPI (0E8, W)**
	LOPE (153)		LOPIEN (0F9)
	PAISCONI (151, R)	**PAISCONV (150)**	
	PAISCONE (153)		
	PAISI (151, R)	**PAISV (150)**	**PAISI (0E8, W)**
	PAISE (153)		PAISEN (0F9)
	PRDII (151, R)	**PRDIV (150)**	**PRDII (0E8, W)**
	PRDIE (153)		PRDIEN (0F9)
	PREII (151, R)		
	PREIE (153)		
	CONCATI (152, R)		
	CONCATE (154)		
	DISCOPAI (152, R)		
	DISCOPAE (154)		
	ILLJREQI (152, R)		
	ILLJREQE (154)		
	INVNDFI (152, R)		
	INVNDFE (154)		
	NDFI (152, R)		
	NDFE (154)		
	NSEI (152, R)		
	NSEE (154)		
	PSEI (152, R)		
	PSEE (154)		
	COMAI (15C, R)		
	COMAE (15C)		
	LOMI (15C, R)	**LOMV (15C)**	**LOMI (0E8, W)**
	LOME (15C)		LOMIEN (0F9)
RPOP3I (104)	**NEWPTRI (190, R)**		
	NEWPTRE (190)		
	BIPEI (191, R)		
	BIPEE (193)		

Interrupt Hierarchy		State Bit	Auxiliary
RPOP3I (cont'd)	**ERDII (191, R)** ERDIE (193)	**ERDIV[2:0] (190)**	**EPRDII#3 (0EA, W)** EPRDII#3EN (0FB)
	LOPCONI (191, R) LOPCONE (193)	**LOPCONV (190)**	
	LOPI (191, R) LOPE (193)	**LOPV (190)**	**LOPI (0E9, W)** LOPIEN (0FA)
	PAISCONI (191, R) PAISCONE (193)	**PAISCONV (190)**	
	PAISI (191, R) PAISE (193)	**PAISV (190)**	**PAISI (0E9, W)** PAISEN (0FA)
	PRDII (191, R) PRDIE (193)	**PRDIV (190)**	**PRDII (0E9, W)** PRDIEN (0FA)
	PREII (191, R) PREIE (193)		
	CONCATI (192, R) CONCATE (194)		
	DISCOPAI (192, R) DISCOPAE (194)		
	ILLJREQI (192, R) ILLJREQE (194)		
	INVNDFI (192, R) INVNDFE (194)		
	NDFI (192, R) NDFE (194)		
	NSEI (192, R) NSEE (194)		
	PSEI (192, R) PSEE (194)		
	COMAI (19C, R) COMAE (19C)		
	LOMI (19C, R) LOME (19C)	**LOMV (19C)**	**LOMI (0E9, W)** LOMIEN (0FA)
RTAL1I (104)	**ESEI (129, R)** ESEE (128)		
	NPJI (129, R) DPJEE (128)		
	PPJI (129, R) DPJEE (128)		
	ISFI (12A, R) ISFE (128)	**ISFV (12A)**	

Interrupt Hierarchy		State Bit	Auxiliary
RTAL2I (104)	**ESEI (169, R)**		
	ESEE (168)		
	NPJI (169, R)		
	DPJEE (168)		
	PPJI (169, R)		
	DPJEE (168)		
	ISFI (16A, R)	**ISFV (16A)**	
	ISFE (168)		
RTAL3I (104)	**ESEI (1A9, R)**		
	ESEE (1A8)		
	NPJI (1A9, R)		
	DPJEE (1A8)		
	PPJI (1A9, R)		
	DPJEE (1A8)		
	ISFI (1AA, R)	**ISFV (1AA)**	
	ISFE (1A8)		
SPTB1I (105)	**RTIMI (149, R)**	**RTIMV (149)**	**TIMI (0E7, W)**
	RTIMIE (149)		TIMIEN (0F8)
	RTIUI (149, R)	**RTIUV (149)**	**TIUI (0E7, W)**
	RTIUIE (149)		TIUIEN (0F8)
	UNEQI (149, R)	**UNEQV (149)**	
	UNEQIE (14D)		
	RPSLMI (14D, R)	**RPSLMV (14D)**	**PSLMI (0E7, W)**
	RPSLMIE (14D)		PSLMIEN (0F8)
	RPSLUI (14D, R)	**RPSLUV (14D)**	**PSLUI (0E7, W)**
	RPSLUIE (14D)		PSLUIEN (0F8)
SPTB2I (105)	**RTIMI (189, R)**	**RTIMV (189)**	**TIMI (0E8, W)**
	RTIMIE (189)		TIMIEN (0F9)
	RTIUI (189, R)	**RTIUV (189)**	**TIUI (0E8, W)**
	RTIUIE (189)		TIUIEN (0F9)
	UNEQI (189, R)	**UNEQV (189)**	
	UNEQIE (18D)		
	RPSLMI (18D, R)	**RPSLMV (18D)**	**PSLMI (0E8, W)**
	RPSLMIE (18D)		PSLMIEN (0F9)
	RPSLUI (18D, R)	**RPSLUV (18D)**	**PSLUI (0E8, W)**
	RPSLUIE (18D)		PSLUIEN (0F9)
SPTB3I (105)	**RTIMI (1C9, R)**	**RTIMV (1C9)**	**TIMI (0E9, W)**
	RTIMIE (1C9)		TIMIEN (0FA)

Interrupt Hierarchy		State Bit	Auxiliary
SPTB3I (cont'd)	**RTIUI (1C9, R)** RTIUIE (1C9)	**RTIUV (1C9)**	**TIUI (0E9, W)** TIUIEN (0FA)
	UNEQI (1C9, R) UNEQIE (1CD)	**UNEQV (1C9)**	
	RPSLMI (1CD, R) RPSLMIE (1CD)	**RPSLMV (1CD)**	**PSLMI (0E9, W)** PSLMIEN (0FA)
	RPSLUI (1CD, R) RPSLUIE (1CD)	**RPSLUV (1CD)**	**PSLUI (0E9, W)** PSLUIEN (0FA)
TTAL1I (105)	**ESEI (141, R)** ESEE (140)		
	NPJI (141, R) PJEE (140)		
	PPJI (141, R) PJEE (140)		
	ISFI (142, R) ISFE (140)	**ISFV (142)**	
TTAL2I (105)	**ESEI (181, R)** ESEE (180)		
	NPJI (181, R) PJEE (180)		
	PPJI (181, R) PJEE (180)		
	ISFI (182, R) ISFE (180)	**ISFV (182)**	
TTAL3I (105)	**ESEI (1C1, R)** ESEE (1C0)		
	NPJI (1C1, R) PJEE (1C0)		
	PPJI (1C1, R) PJEE (1C0)		
	ISFI (1C2, R) ISFE (1C0)	**ISFV (1C2)**	
	API (109, R) APE (109)		

5 SONET AND SDH CONFIGURATION

The following table lists the major configuration differences in the SPECTRA-155 for all SDH and SONET applications. This list is not exhaustive, and all configuration values should be checked to ensure they meet current standards requirements.

Not shown in the table are configuration registers that are application dependent, like the C2 byte. Also not shown are registers that require real-time processor intervention, such as the K1, K2 bytes which are monitored and updated according to the APS status.

Table 10 - SDH and SONET Configuration Values

Register Bit (address)	SDH Value	SONET Value	Standards References
LRDIDET (08)	0 or 1	0	G.783 (04/97) 2.2.2.7 GR-253-CORE (Issue 2) R6-184
LAISDET (08)	0 or 1	0	G.783 (04/97) 5.2.2 GR-253-CORE (Issue 2) R6-147
RASE Block	see datasheet SDH settings	see datasheet SONET settings	G.783 (04/97) 2.2.2.5 GR-253-CORE (Issue 2) 5.3.3
LEN16 (38, 148, 188, 1C8)	1	0	G.783 (04/97) 9.2.2.2
PER5 (38, 148, 188, 1C8)	?	?	not defined
PSL5 (110, 150, 190)	1	1	G.783 (04/97) 2.2.2.2 GR-253-CORE (Issue 2) R6-119
RDI10 (116, 156, 196)	0	0 or 1	G.783 (04/97) 2.2.2.15 GR-253-CORE (Issue 2) R6-197, R6-198
DISFS (BD, CD, DD, 11D, 15D, 19D)	1	0	G.707 (03/96) 9.3.1.2 GR-253-CORE (Issue 2) R3-21
ENSS (BD, CD, DD, 11D, 15D, 19D)	1	0	G.783 (04/97) Annex C GR-253-CORE (Issue 2) 3.5.1
SSS (128, 168, 1A8)	1	0	G.783 (04/97) Annex C GR-253-CORE (Issue 2) 3.5.1
PERSIST (130, 170, 1B0)	0	1	G.783 (04/97) under discussion GR-253-CORE (Issue 2) O6-193
EXCFS (130, 170, 1B0)	1	0	G.707 (03/96) 9.3.1.2 GR-253-CORE (Issue 2) R3-21

Register Bit (address)	SDH Value	SONET Value	Standards References
S[1:0] (135, 175, 1B5)	10b	xxb (usually 00b)	G.707 (03/96) 8.1.2 GR-253-CORE (Issue 2) 3.5.1
CONCAT[11:10] (13F)	xxb (usually 10b)	xxb (usually 00b)	G.707 (03/96) 8.1.2 GR-253-CORE (Issue 2) 3.5.1.4

APPENDIX: OVERHEAD BYTES

The figure below gives a map of all overhead bytes. This map shows the names given to overhead bytes in both SONET and SDH. Where the two standards give different names to the bytes, the SONET name is shown on top and the SDH name on the bottom. Unused bytes are not named.

Figure - SONET and SDH Overhead Bytes

Section RSOH	A1	A1	A1	A2	A2	A2	J0	Z0 Nat	Z0 Nat	J1
	B1			E1			F1	Nat	Nat	B3
	D1			D2			D3			C2
AU	H1	H1*	H1*	H2	H2*	H2*	H3	H3	H3	G1
Line MSOH	B2	B2	B2	K1			K2			F2
	D4			D5			D6			H4
	D7			D8			D9			Z3 F3
	D10			D11			D12			Z4 K3
	S1	Z1	Z1	Z2	Z2	M1	E2	Nat	Nat	Z5 N1

Path

Transport Overhead

A1, A2 — Framing. The framing bytes contain constant F6, 28h values which allow network equipment to determine the start of a frame in the serial datastream.

B1 — BIP-8. Bit interleaved parity for error monitoring at the Section (Regenerator Section) level.

B2 — BIP-*m*x8. Bit interleaved parity for error monitoring at the Line (Multiplex Section) level.

D1 – D12 — Data Communications Channels. Separated into two separate channels: a 192kb/s Section (Regenerator Section) DCC consisting of D1-D3, and a 578kb/s Line (Multiplex

	Section) DCC consisting of D4-D12. The DCC channels normally contain alarm, maintenance, and control information using a complex message-based protocol contained within HDLC frames.
E1, E2	Orderwire. 64kbit/s voice channels (8-bit PCM encoded) for craftsperson communications.
F1	User Channel. Reserved for use by the network provider.
H1, H2	Pointer. The pointer value bytes locate the start of the SPE (VC-4 or VC-3) within the frame.
H3	Pointer Action. The pointer action byte contains payload when a negative stuff event occurs, and is undefined at all other times.
J0	Section Trace/Multiplex Section Trace. In older systems, J0 was known as C1, and contained a 01h to identify the first STS-1 (AU-3) within a multiplexed datastream. This byte has since been redefined to contain a 64 byte (16 byte in SDH) trace identifier message.
K1, K2	Automatic Protection Switching (APS) Channel. The K1 and K2 bytes are allocated for signalling in Line (Multiplex Section) protection schemes. In addition, bits 6, 7, and 8 of the K2 byte are used to convey Line (Multiplex Section) RDI and AIS signals.
M0	STS-1 (STM-0) REI. Not defined in STS-3 (STM-1) and higher rates. Bits 5-8 provide a count of far-end B2 errors (formerly known as FEBEs). Bits 1 through 4 are currently undefined.
M1	STS-N REI (MS-REI). Provides a count of far-end B2 errors (formerly known as FEBEs).
Nat	National Use (SDH only). Reserved for national use.
S1	Synchronization Status Byte. The S1 byte provides the synchronization status byte. Bits 5 through 8 of the synchronization status byte identifies the synchronization source of the STS-3c/1 (STM-1) signal. Bits 1 through 4 are currently undefined.

Z0	Section Growth (SONET only). Reserved for future growth.
Z1, Z2	Growth (SONET only). Reserved for future growth.

Path Overhead

B3	Path BIP-8. Bit interleaved parity for monitoring at the Path level.
C2	Signal Label. A code identifying the content or status of the payload. For example, the value 02 indicates a VT (TUG) structured payload.
F2	Path User Channel. Reserved for communications between path terminating network equipment.
F3	Path User Channel (SDH only). Reserved for communications between path terminating network equipment.
G1	Path Status. Bits 1-4 provide a count of far-end B3 errors. (REI, formerly FEBE). Bit 5 provides a remote defect indication (RDI). Bits 6 and 7 optionally contribute to the RDI to form an enhanced RDI code in which the type of defect may be communcated. Bit 8 is undefined.
H4	Position Indicator. Payload-specific position indicator. For VT (TUG) structured payloads, this byte is used as a multiframe phase indicator.
J1	Path Trace. A 64 byte (16 byte in SDH) trace identifier message.
K3	Automatic Protection Switching (APS) Channel (SDH only). Allocated for signalling in Path protection schemes.
N1	Network Operator Byte (SDH only). Used for the Tandem Connection function.
Z3 - Z5	Path Growth (SONET only). Reserved for future growth.

CONTACTING PMC-SIERRA, INC.

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

Tel: (604) 415-6000

Fax: (604) 415-6200

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

PMC-961079 (R5) ref PMC-xxxxxx (Rx) Issue date: May 1998