DETECTION OF CONCATENATION INDICATOR LOP AND AIS ALARMS

Issue 1: January 8, 1996

TABLE OF CONTENTS

OVERVIEW

The present implementation of the S/UNI-622 does not interpret the concatenation bytes of the STS-12c or STM-Mc frame structure. This application note addresses this condition (errata item number 1.6.5 of the 'S/UNI-622 Errata Notice' document PMC-950217, issue 4), and describes the external logic required to implement the detection of concatenation loss of pointer (LOP) and concatenation indicator alarm indication signal (AIS) as specified in ITU Recommendation G.783.

The frame structure of an STS-12c and an STM-4c is shown in figure 1 and figure2. These diagrams identify the locations of all concatenation indicator bytes for the SONET and SDH frame structures.

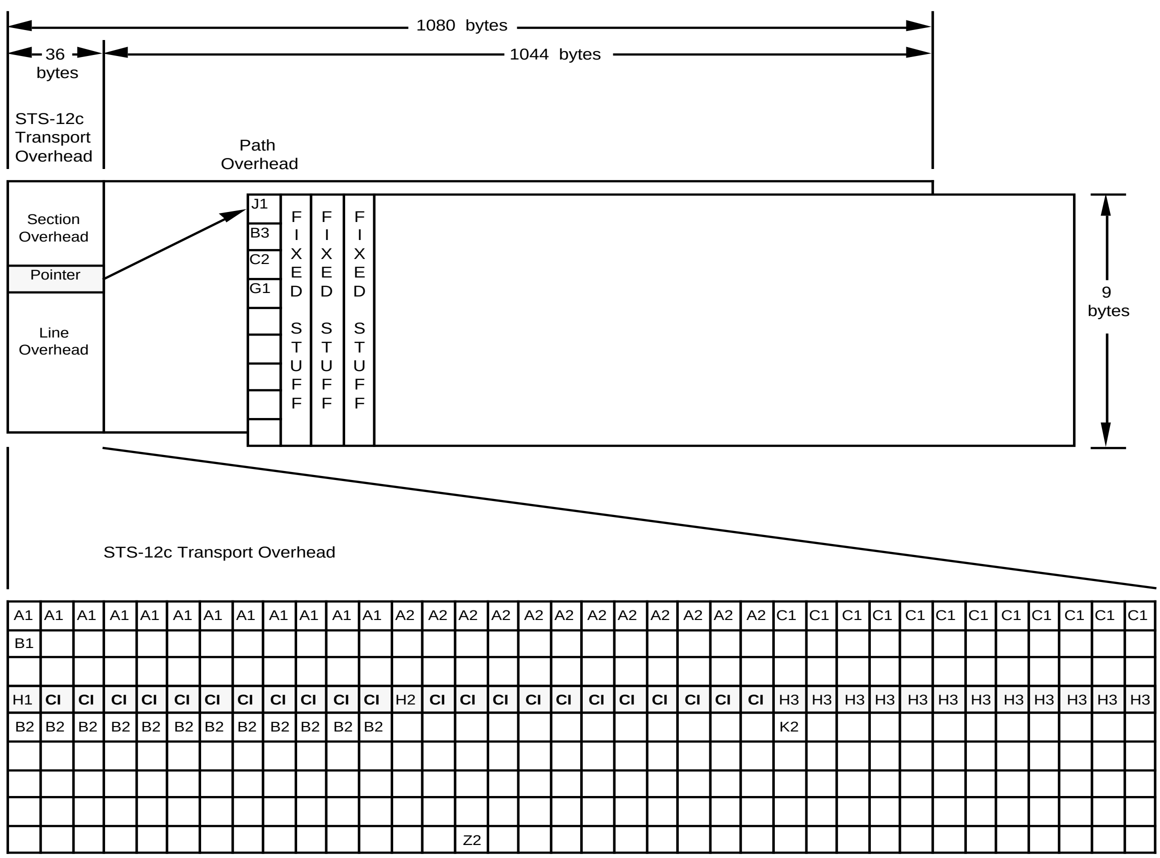
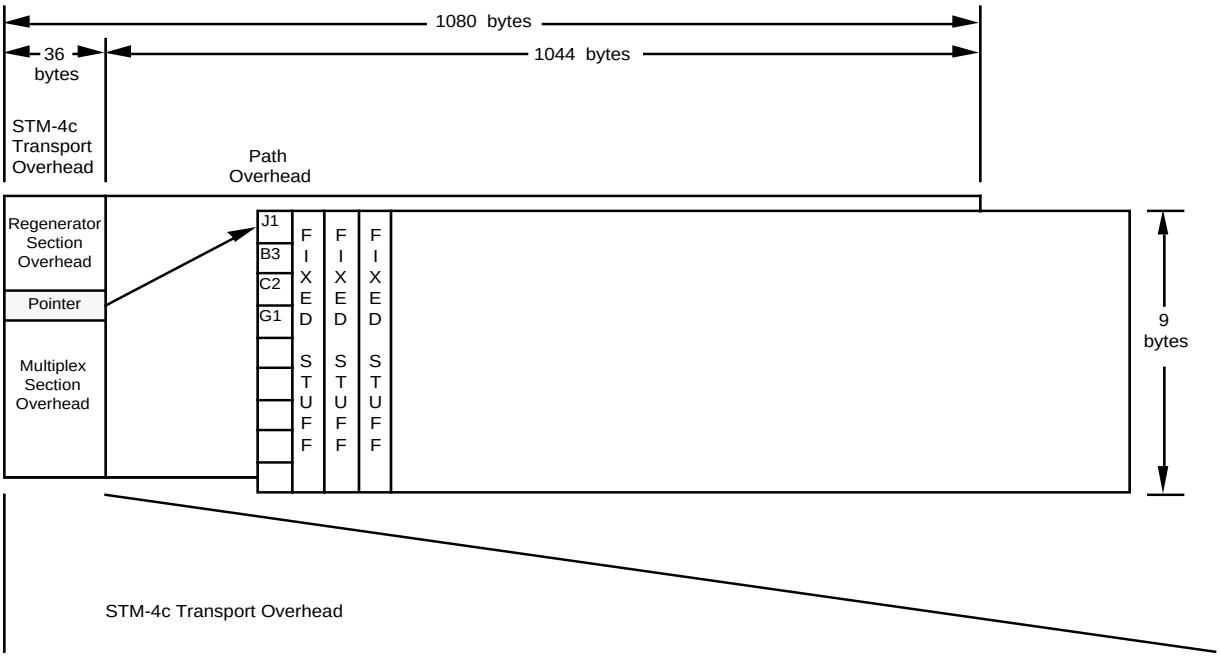


Figure 1: SONET STS-12c Frame Structure.

The concatenation indicator bytes in an STS-12c frame structure are shown labeled CI in figure 1. The concatenation indicator byte in column 2 of the transport overhead and the concatenation indicator byte in column 14 of the transport overhead both combine to produce the first of a set of eleven complete concatenation indicators. The second complete concatenation indicator is

produced by combining the CI byte in column 3 with the CI byte in column 15. This continues until the last complete concatenation indicator is produced by combining the CI byte in column 12 with the CI byte in column 24.



A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1
B1																																		
H1	**CI**	**CI**	**CI**	FS	FS	FS	FS	FS	FS	FS	FS	**H2**	**CI**	**CI**	**CI**	FS	FS	FS	FS	FS	FS	FS	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3
B2	B2	B2	B2	B2	B2	B2	B2	B2	B2	B2	B2																							
													Z2																						

Figure 2: SDH STM-4c Frame Structure

The concatenation indicator bytes in an STM-4c frame structure are shown labeled CI in figure 2. Bytes labeled FS are fixed stuff bytes and should be ignored for concatenation indicator interpretation. The concatenation indicator byte in column 2 of the transport overhead and the concatenation indicator byte in column 14 of the transport overhead combine to produce the first of a set of three complete concatenation indicators. The second complete concatenation indicator is produced by combining the CI byte in column 3 with the CI byte in column 15. The third complete concatenation indicator is produced by combining the CI byte in column 4 with the CI byte in column 16.

CONCATENATION INDICATOR INTERPRETATION STATE MACHINE

ITU recommendation G.783 specifies a procedure that describes the interpretation of the concatenation indicator bytes and the generation of LOPC (Loss Of Concatenated Pointer) and AISC (Concatenated Alarm Indication Signal) alarm conditions. Figure 3 in this document is identical to figure B.2 shown in the ITU recommendation G.783 (dated March 1994) and shows the three state state diagram required to interpret the concatenation indicator for the detection of concatenation Indicator LOPC and AISC alarms.

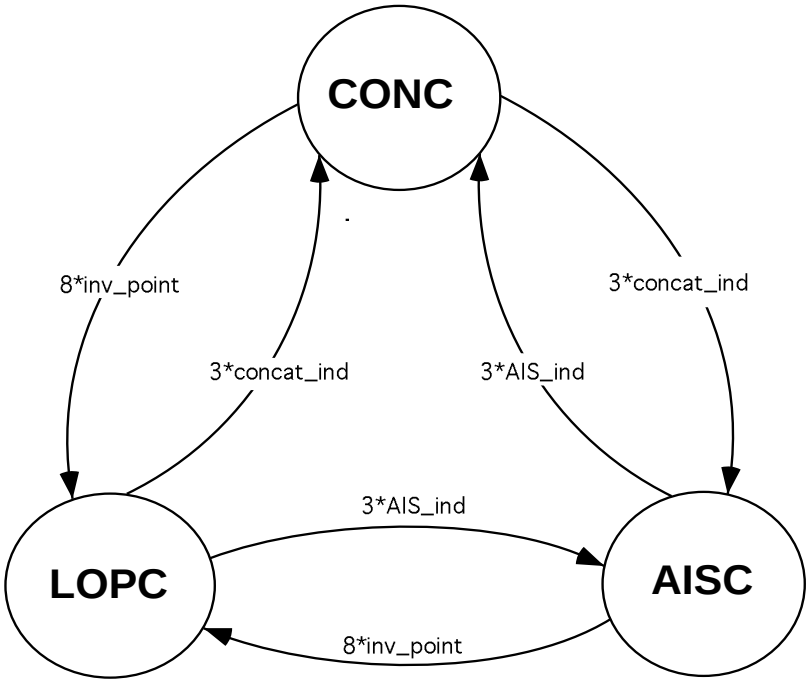


Figure 3. Concatenation Indicator Interpretation State Diagram

CONC State

This state is the normal state and is occupied when a minimum of three consecutive concatenation indicators have been received. This state is exited when 8 consecutive 'inv_point' values or 3 consecutive 'AIS_ind' concatenation indicator values are received.

LOPC State

This state is entered when 8 consecutive 'inv_point' concatenation indicator values are received when in any of the other states of the state machine. When in this

state, the LOPC alarm is generated. The LOPC alarm is negated after exiting this state

AISC State

This state is entered when 3 consecutive 'AIS_ind' concatenation indicator values are received when in any of the other states of the state machine. When in this state, the AISC alarm is generated. The AIS alarm is negated after exiting this state.

Concatenation Indicator Event Interpretation

In an STS-Nc or STM-Mc frame the pointer bytes are located in the first H1 and H2 bytes (as previously shown in figure 1 and figure 2). The remaining N-1 (or M-1) sets of H1 and H2 bytes contain the concatenation indicator and will contain the following two valid values, 1001dd11-11111111B' (Conc_ind) or 11111111-11111111B' (AIS_ind). Any other binary value is an invalid value and is labeled "Inv_point". In addition, this application note will implement the following interpretation/extension to the concatenation indicator interpretation; "all N-1 (M-1) sets of concatenation indicators must be processed as a group and they must all be identical to be correctly interpreted as an 'AIS_ind' or a 'Conc_ind'. Any mismatch in the N-1 (or M-1) sets of concatenation indicators will be interpreted as an 'Inv_point'". As an example, a correct "AIS_ind" in an STS-12c (which has eleven concatenation indicators) frame will be interpreted if all eleven contain the value 11111111-11111111B'. If however one of these concatenation indicators were to contain some other value, then the concatenation indicator will be interpreted as an 'inv_point'. Similarly, a correct "Concat_ind" will be interpreted if all eleven concatenation indicators contain the value 1001dd11-11111111B'.

TOP LEVEL BLOCK DIAGRAM

The block diagram in figure 4 shows the logic required to implement the detection of concatenated pointer AIS and concatenated pointer LOP. The serial transport overhead output streams of the S/UNI-622 are used by the 'H1 Pointer Extract' and 'H2 Pointer Extract' blocks to extract all H1 and H2 bytes (including the CI bytes).

The pointer/indicator extracted bytes are decoded and interfaced to the 'EVENT LOG' block on the H1_EQ_CI, H1_EQ_AIS, H2_EQ_CI and H2_EQ_AIS inputs. The 'EVENT LOG' block inspects the decoded pointer/indicator bytes in a serial fashion beginning with the first byte (indicated by the START control output from the 'CONTROL' block) and ending with the last H2 byte. The START input allows the 'EVENT LOG' block to start its byte serial inspection process afresh. After inspecting all such bytes the type of concatenated indicator event is indicated on the INV, AIS and CI outputs. These outputs are utilized at the next RTOHFP synchronization.

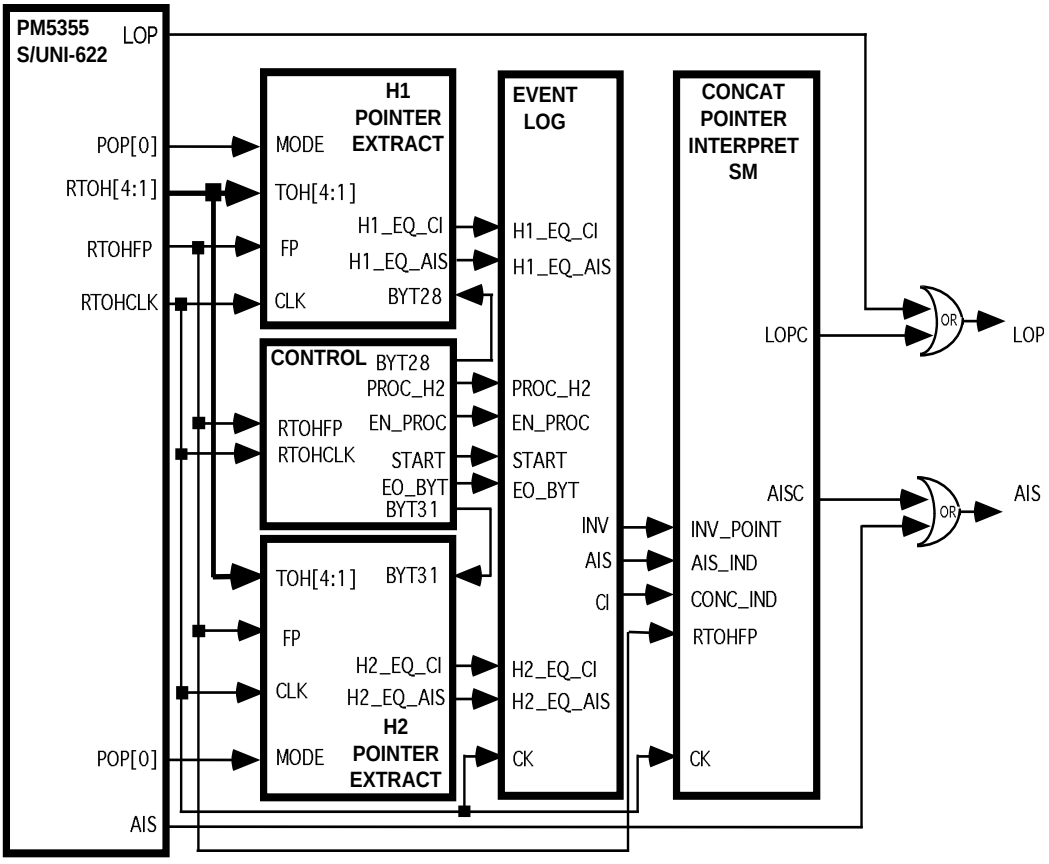


Figure 4: Top Level Block Diagram

The 'Concat Pointer Interpret SM' block implements the functionality of figure 3. The 'Inv_point', 'AIS_ind' and 'Conc_ind' inputs are input from the 'EVENT LOG ' block. The LOPC output indicates that a concatenated indicator loss of pointer has been detected. The AISC output indicates that a concatenated indicator AIS alarm is detected. These outputs are logically OR'ed with their pointer LOP and pointer AIS counterpart that is generated by the S/UNI-622.

H1 Pointer Extract

This block extracts the H1 bytes (H1 pointer byte plus 11 concatenation indicators) and decodes the value of the pointer. The 8 bit shift registers hold the incoming serial data from each of the four RTOH[4:1] streams. The serial data is shifted in on the DIN input of the shift register on the rising edge of RTOHCLK. The parallel outputs D[7:0] are the outputs of each DFF in the shift register chain, where D[7] represents the last stage of the shift register and D[0] represents the first stage of the shift register. The parallel outputs from the shift register are decoded by the 'CI COMP' and 'AIS COMP' blocks. The 'CI COMP' block compares the register value with 1001xx11B'. The 'AIS COMP' block compares the register value with 11111111B'. If either of these blocks find a match, the appropriate output is set active, high; H1_EQ_CI is activated when the H1 bytes compare with 1001xx11B' and the H1_EQ_AIS is activated if the H1 bytes compare with 11111111B'.

When BYT28 is active the RTOH[1] serial to parallel conversion is ignored by the AIS COMP and CI COMP blocks since the first H2 byte is not a concatenation indicator.

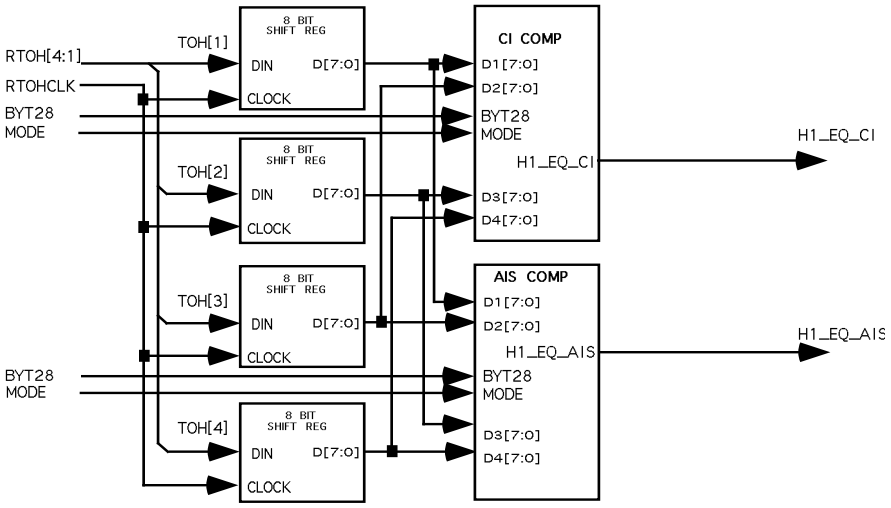


Figure 5: H1 Pointer Extract

When MODE is active (selecting a second mode of operation for SDH instead of SONET) all bytes other than the byte identified by BYT28 are ignored by the AIS COMP and CI COMP blocks since the H1 bytes not identified as BYT28 are all fixed stuff bytes. In addition the bytes on stream RTOH[1] are ignored at all times

since the bytes on this stream are either real pointer bytes (as opposed to concatenated pointer bytes) or fixed stuff bytes.

CI COMP

This block detects the concatenation value of 1001xx11B' on all valid H1 bytes. When MODE is logic 1 (SDH mode) only BYT28 is valid. When MODE is logic 0 all H1 bytes are valid except byte 28 (decimal) on stream RTOH[1]. Figure 6 shows the required logic.

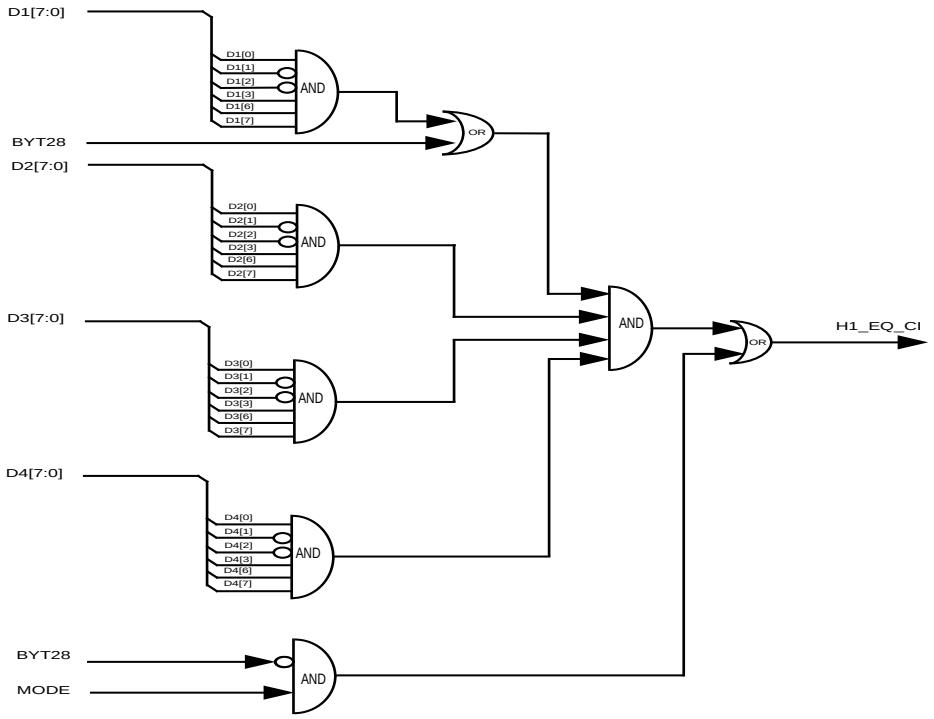


Figure 6: CI COMP

AIS COMP

This block detects the concatenation value of 11111111B' on all valid H1 bytes. When MODE is logic 1 (SDH mode) only BYT28 is valid. When MODE is logic 0 all H1 bytes are valid except byte 28 (decimal) on stream RTOH[1]. Figure 7 shows the required logic.

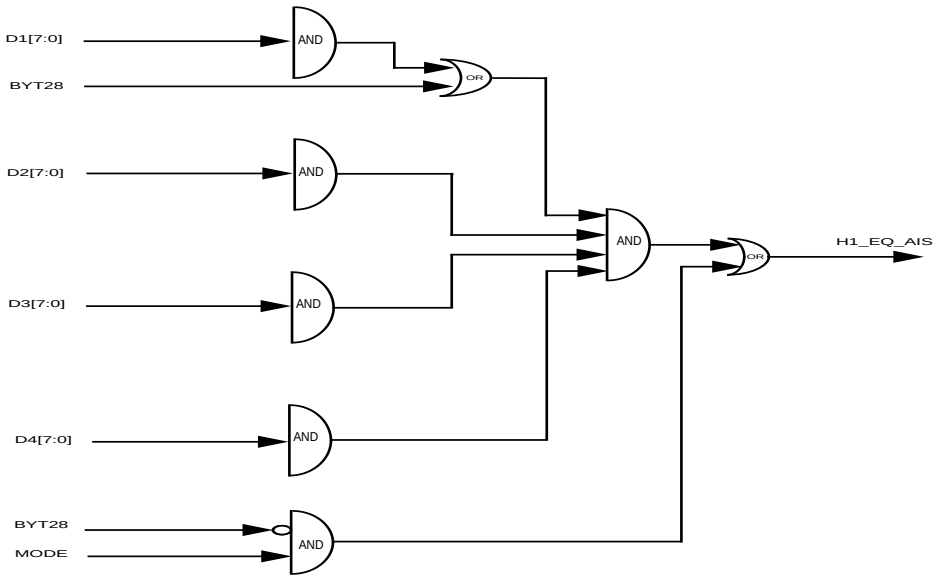


Figure 7: AIS COMP

H2 Pointer Extract

This block extracts the H2 bytes (H2 pointer byte plus 11 concatenation indicators) and decodes the value of the pointer. The 8 bit shift registers hold the incoming serial data from each of the four RTOH[4:1] streams. The parallel output from the shift register is decoded by the 'AIS/CI COMP' block. The 'AIS/CI COMP' block compares the register value with 11111111B'. If this blocks finds a match, the output goes active high; both H2_EQ_CI and H2_EQ_AIS is activated when the H2 bytes compare with 11111111B'.

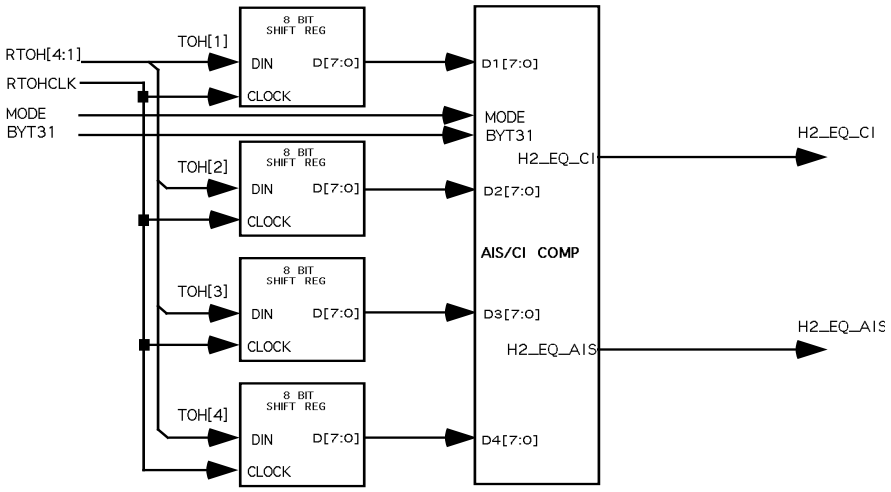


Figure 8: H2 Pointer Extract

When BYT31 is active the RTOH[1] serial to parallel conversion is ignored by the AIS/CI COMP block since the first H2 byte is not a concatenation indicator.

When MODE is active (selecting a second mode of operation for SDH operation instead of SONET operation) all bytes other than the byte identified by BYT31 are ignored by the AIS/CI COMP block since the H2 bytes not identified as BYT31 are all fixed stuff bytes. In addition the bytes on stream RTOH[1] are ignored at all times since the bytes on this stream are either real pointer bytes (as oposed to concatenated pointer bytes) or fixed stuff bytes.

AIS/CI COMP

This block detects the concatenation value of 11111111B' on all valid H2 bytes. When MODE is logic 1 (SDH mode) only BYT31 is valid. When MODE is logic 0 all H2 bytes are valid except byte 31 (decimal) on stream RTOH[1]. Figure 9 shows the required logic.

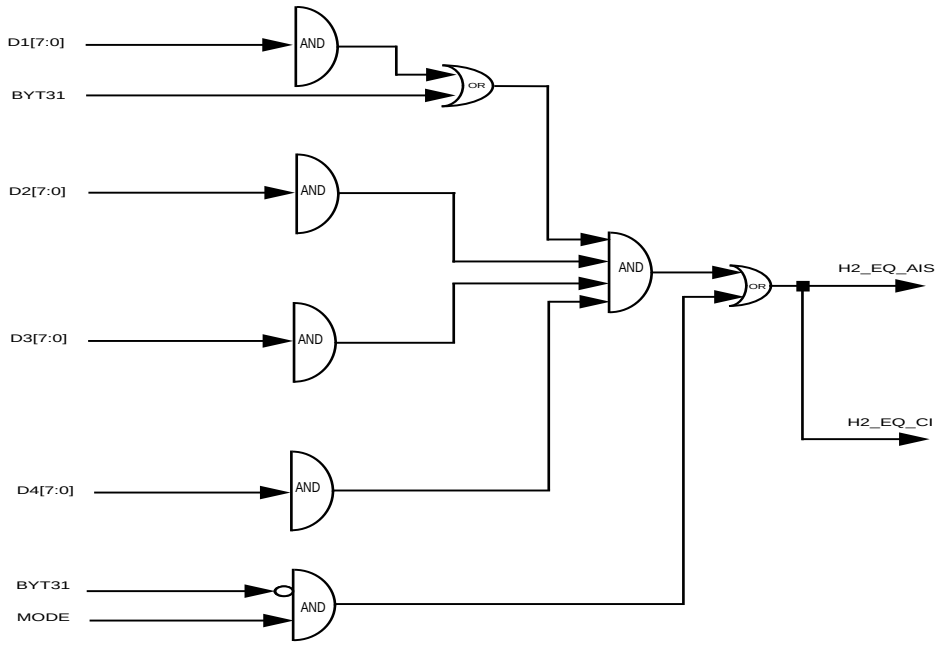


Figure 9: AIS/CI COMP

CONTROL

This block controls and times the functionality of the other blocks, such that the signals from these blocks are used only when valid and ignored otherwise. The control logic consists of a 9 bit counter and a decoder. The nine bit count value is sufficient to identify the H1 and H2 bytes within the RTOH[4:1] serial streams. When the count value of the upper 6 bits is 28 (decimal) the first H1 byte is identified. When the count value of the upper 6 bits is 33 the last H2 byte is identified. The counter counts past the count required for the identification of the

H1 and H2 bytes and halts until the next frame pulse is detected. The arrival of the frame pulse resets the counter back to zero and allows the identification of the H1 and H2 bytes again during the following frame. This process repeats at every frame.

The outputs of the 'CONTROL' block indicate the start of the concatenation byte sequence (START), the time when the H2 byte processing is valid (PROC_H2) and the time when processing of either H1 or H2 is valid (EN_PROC). It also identifies byte 28 (decimal) and byte 31 (decimal). Bytes 28 and 31 need to be identified in order to ignore their value during concatenated pointer interpretation.

The BYTEDEC block decodes the time when the counter reads decimal 33 (BYT33), 28 (BYT28) and 31 (BYT31). The BIT_EQ7 output is a decode of count bits CNT[2:0] being 111B'. The GT30 output is a decode of a count value being greater than decimal 30. This is logically derived by ORing the output of CNT[0]**&**CNT[1]**&**CNT[2]**&**CNT[3]**&**CNT[4] with the output of the logical OR of CNT[5]**OR**CNT[6]**OR**CNT[7]**OR**CNT[8].

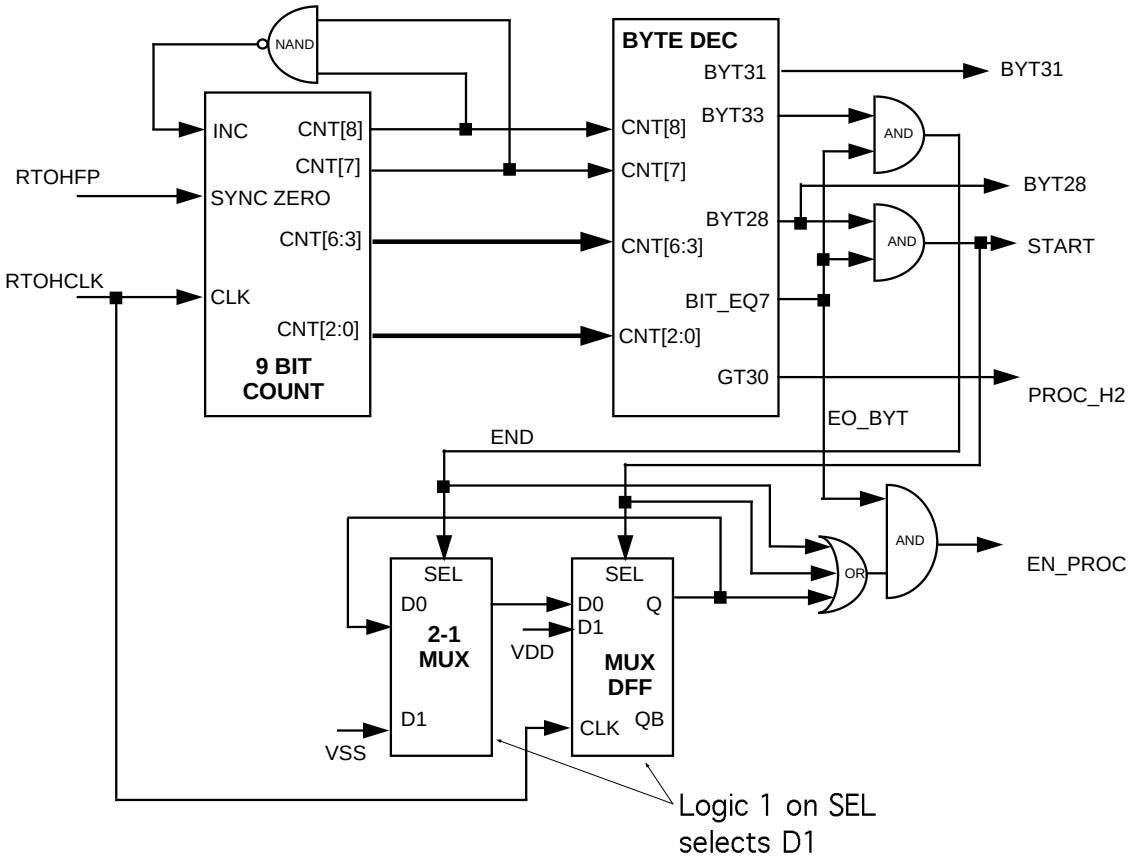


Figure 10: Control Block.

The 2-1 MUX and MUX-DFF ensure that the processing is enabled during the 3 H1 (equivalent to 12 H1 bytes over all 4 RTOH[n] streams) and 3 H2 (equivalent to 12 H2 bytes over all 4 RTOH[n] streams) byte interval. The output EN_PROC is active during the last bit interval of every byte.

EVENT LOG

This block serially processes the H1 and H2 events as directed by the 'CONTROL' block. The first stage of two 2-1 muxes selects the appropriate byte for processing. If PROC_H2 is active (logic 1) then the multiplexers select H2_EQ_AIS and H2_EQ_CI for processing. If PROC_H2 is inactive (logic 0) then the multiplexers select H1_EQ_AIS and H1_EQ_CI for processing.

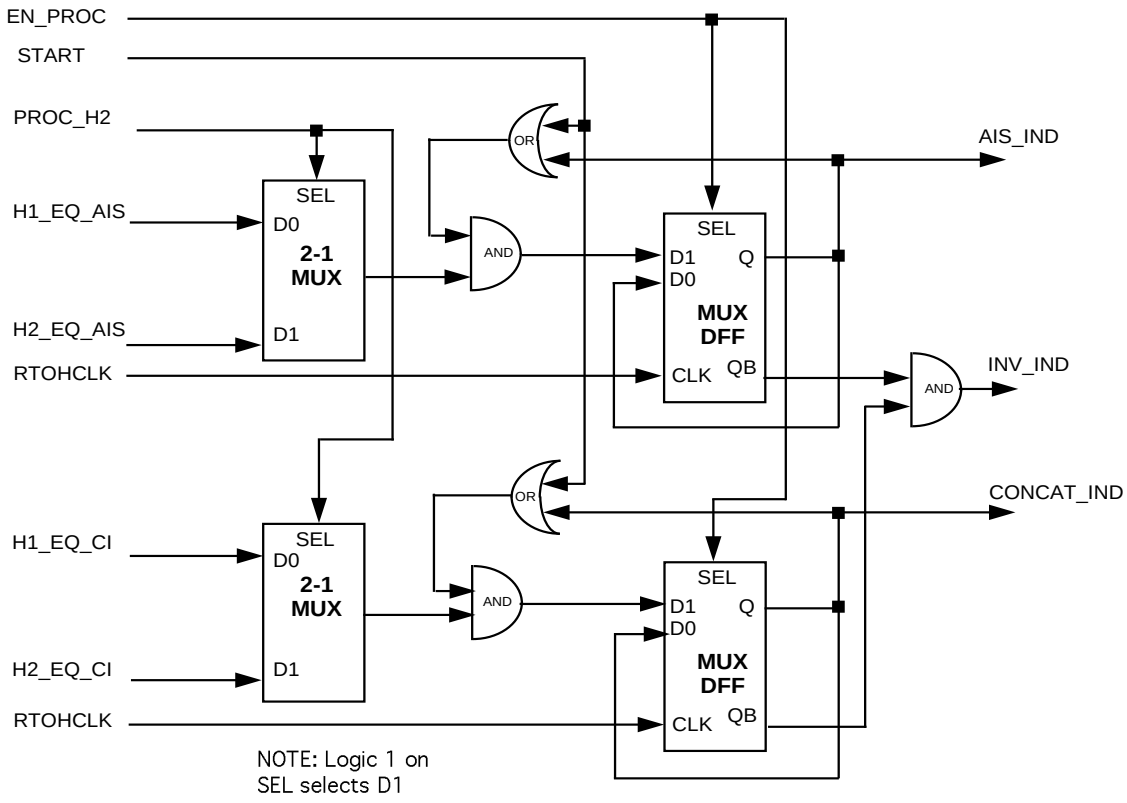


Figure 11: EVENT LOG Block.

The two mux DFF's store a successive indication of the H1 and H2 byte events as directed by the EN_PROC input. When START is active, the fed back output of these DFF's are ignored in determining the next value of the MUX-DFF's Q output; during this time only the 2-1 MUX output determines the data written to the MUX-DFF. After the first initial start byte, the data written to the MUX-DFF is dependent on the 2-1 MUX output and the previous data contained in the MUX-DFF; i.e. processing proceeds such that if a zero ever gets written into the MUX-DFF, then it will remain a zero until the reinitialization at the next START event. This means

that if a H1 or H2 byte was received that did not match an AIS indication then a zero will be written into the AIS_IND output and will remain logic zero (regardless of the remaining received concatenation indication bytes) until the next frame of concatenation indicators. A similar process applies to the CONCAT_IND output MUX-DFF.

All memory elements are reset on power up.

CONCAT POINTER INTERPRET SM.

This block implements the process shown in figure 3. The circuitry of this block is enabled once every frame during the logic high level of the RTOHFP input. The INV COUNT block counts 8 consecutive invalid indications, the CI COUNT block counts 3 consecutive concatenation indications and the AIS COUNT block counts 3 consecutive AIS concatenation indications. The outputs of these three counters indicate the three possible conditions that can cause the transition to an LOPC state or a AISC state or to a CONC state (as shown in figure 3).

The INV COUNT counter is enabled only when the EN input is logic 1; during all rising edges encompassed by the RTOHFP frame synchronization. An increment occurs at a RTOHCLK rising edge during a logic 1 on the RTOHFP input and a logic one on the INV_IND input. The counter is synchronously reset to zero at the next RTOHCLK rising edge when INV_IND input is logic 0 and the RTOHFP input is logic 1. This ensures that the counter will count contiguous INV_IND events only.

The CI COUNT counter is enabled only when the EN input is logic 1; during all rising edges encompassed by the RTOHFP frame synchronization. An increment occurs at a RTOHCLK rising edge during a logic 1 on the RTOHFP input and a logic one on the CONC_IND input. The counter is synchronously reset to zero at the next RTOHCLK rising edge when CONC_IND input is logic 0 and the RTOHFP input is logic 1. This ensures that the counter will count contiguous CONC_IND events only.

The AIS COUNT counter is enabled only when the EN input is logic 1; during all rising edges encompassed by the RTOHFP frame synchronization. An increment occurs at a RTOHCLK rising edge during a logic 1 on the RTOHFP input and a logic one on the AIS_IND input. The counter is synchronously reset to zero at the next RTOHCLK rising edge when AIS_IND input is logic 0 and the RTOHFP input is logic 1. This ensures that the counter will count contiguous AIS_IND events only.

In order to understand the rest of the logic of figure 9 it should be noted that more than one of the three AND gates connected to the outputs of the three counters cannot be active at any one time.

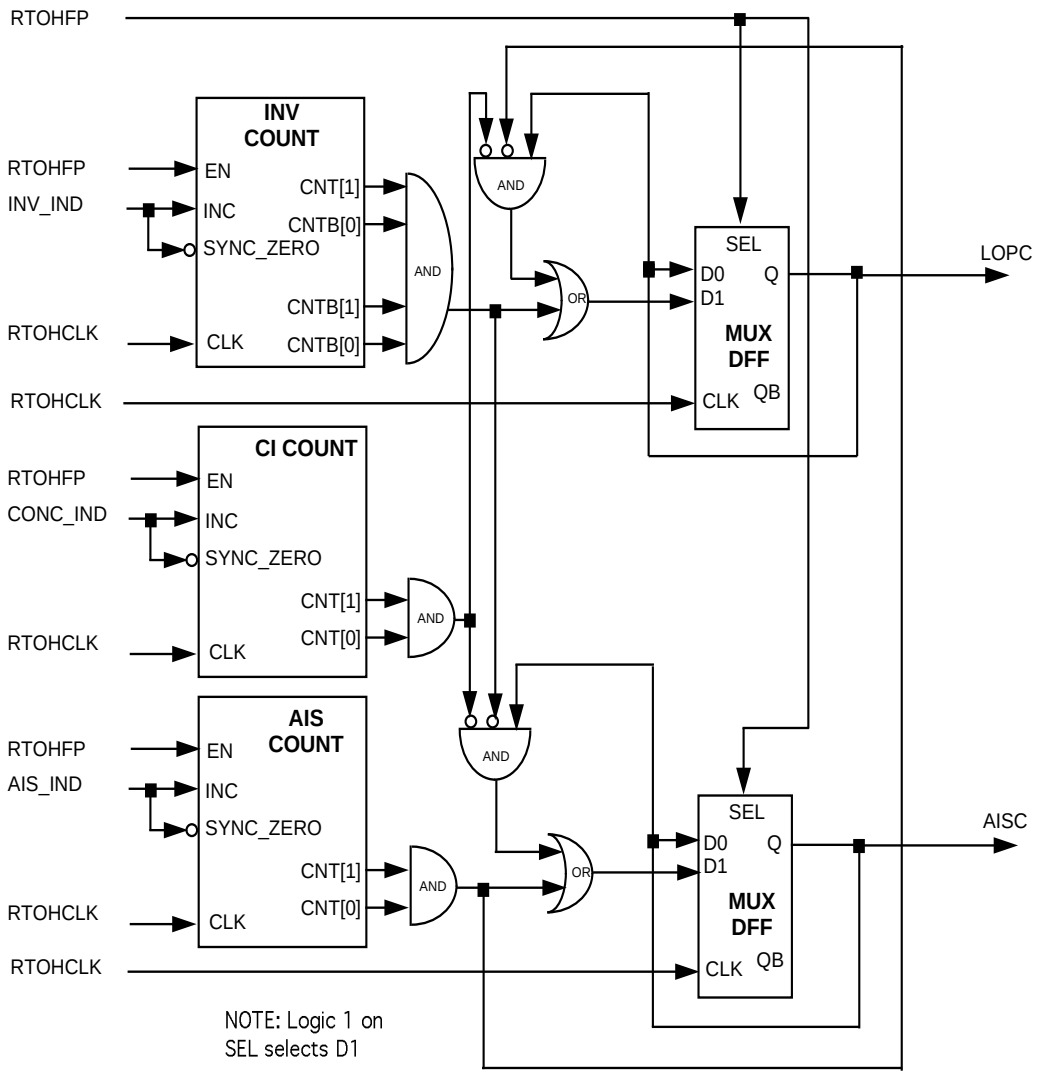


Figure 12: Concat Pointer Interpret SM.

When 8 invalid indications are received (indicated by the output of the 4 input AND gate connected to the outputs of INV COUNT), the LOPC output is set to logic one at the next RTOHCLK rising edge during the reception of RTOHFP. The LOPC output is not reset to logic 0 unless the CI COUNT **OR** the AIS COUNT counters reach a count of three.

Similarly, when 3 AIS indications are received (indicated by the output of the 2 input AND gate connected to the outputs of AIS COUNT), the AISC output is set to logic one at the next RTOHCLK rising edge during the reception of RTOHFP. The AISC output is not reset to logic 0 unless the CI COUNT **OR** the INV COUNT counters reach a count of three or eight respectively.

DISCLAIMER

The circuit presented in this application note has not been built. This circuit is therefore preliminary.

REFERENCES

[1] American National Standards for Telecommunications, ANSI T1.105.03 - 1991

[2] PMC_Sierra Document PMC-941027, S/UNI-622 (PM5355) Data Book, Issue "Advance" October 12, 1994.

[3] International Telecommunications Union, ITU-T Recommendation G.783 - March 1994. "Annex B, Algorithm for Pointer Detection "

NOTES

PMC-960156

Issue date: January 8, 1996.

Printed in Canada