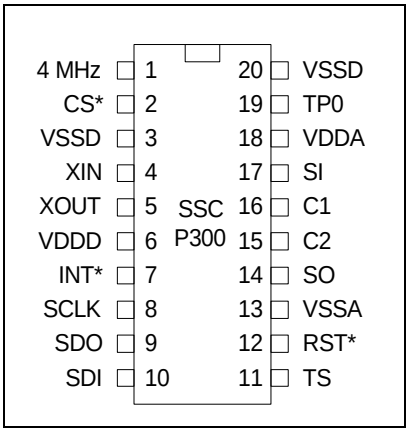
Technical Data Sheet **SSC P300 PL Network Interface Controller**

Features

- Enables Low-cost CEBus® compatible products
- EIA-600 (CEBus) Data Link Layer services
- EIA-600 Physical Layer transceiver
- Spread Spectrum Carrier™ Power Line technology
- SPI Host Processor interface
- Data Link, Controller, and Monitor modes
- Single +5 Volt power supply requirement
- 20 pin SOIC package

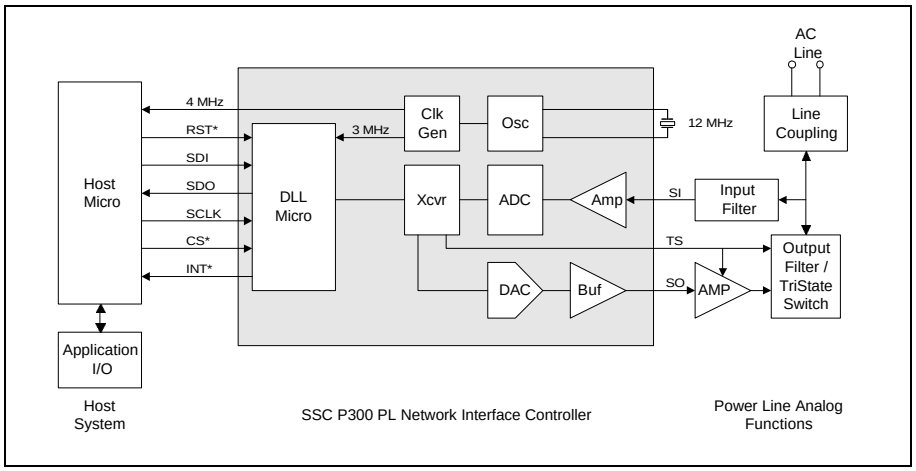


Introduction

The Intellon SSC P300 PL Network Interface Controller is a highly integrated power line transceiver and channel access interface for implementing CEBus® compatible products. The SSC P300 provides the Data Link Layer (DLL) control logic for EIA-600 channel access and communication services, a Spread Spectrum Carrier™ (SSC) power line transceiver, signal conditioning circuitry, and a serial peripheral interface (SPI) compatible host interface. A minimum of external circuitry is required to connect the SSC P300 to the power line. Superior performance is achieved using the SSC P300 in conjunction with the SSC P111 Media Interface IC. The SSC P300 is used with a host microcontroller to construct CEBus compatible products, and serves as the basic communications element in a variety of low-cost power line networking applications.

The inherent reliability of SSC signaling technology and incorporation of basic Data Link functionality combine to provide substantial improvement in network and communication performance over other power line communication methods. The SSC P300 also makes an excellent low cost network interface for twisted pair and DC power systems. A typical CEBus power line node using the SSC P300 is illustrated below.

SSC P300 Node Block Diagram



Absolute Maximum Ratings

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage	-0.3 to 7.0	V
V_{IN}	Input Voltage at any Pin	V_{SS}-0.3 to V_{DD}+0.3	V
T_{STG}	Storage Temperature	-65 to +150	°C
T_L	Lead Temperature (Soldering, 10 seconds)	300	°C

Note:
Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages.

Recommended Operating Conditions

Symbol	Parameter	Min	Typical	Max	Unit
V_{DD}	DC Supply Voltage	4.5	5.0	5.5	V
F_{OSC}	Oscillator Frequency		12 +/- 0.01%		MHz
T_A	Operating Temperature	-40	+25	+85	°C
	Humidity		non-condensing		

Electrical Characteristics

Conditions: V_{DD} = 4.5 to 5.5 V T= -40 to +85°C

Symbol	Parameter	Min	Typical	Max	Units
V_{OH}	Minimum High-level Output Voltage	2.4			V
V_{OL}	Maximum Low-level Output Voltage (1)			0.4	V
V_{IH}	Minimum High-level Input Voltage	2.0			V
V_{IL}	Maximum Low-level Input Voltage			0.8	V
Hys	Minimum Input hysteresis	350			mV
I_{IL}	Maximum Input Leakage Current			+/-10	µA
v_{SO}	SSC Signal Output Voltage (2)		4		V_{P-P}
I_{DD}	Total Power Supply Current		25		mA
	Latchup (3)	150			mA

Notes:
1. I_{OL} = 2 mA
2. Z_L = 2K Ω || 10 pF
3. JEDEC JC -40.2

SSC P300 Pin Assignments

Pin	Mnemonic	Name	Description
1	4MHZ	4 MHz clock out	4 MHz clock output available for host microcontroller.
2	CS*	Chip select	Digital input, active low. Enables serial peripheral interface.
3	VSS_D	Digital ground	Digital ground reference.
4	XIN	Crystal input	Connected to external crystal to excite the IC's internal oscillator and digital clock.
5	XOUT	Crystal output	Connected to external crystal to excite the IC's internal oscillator and digital clock.
6	VDD_D	Digital supply	5.0 VDC +/- 10% digital supply voltage with respect to VSS_D.
7	INT*	Interrupt	Digital output, active low. Attention request to host microcontroller.
8	SCLK	SPI data clock	Serial peripheral interface clock input from host microcontroller.
9	SDO	SPI data out	Data output to host microcontroller serial peripheral interface. SDO is` tristate when CS* is false.
10	SDI	SPI data in	Data input from host microcontroller serial peripheral interface.
11	TS	Tristate	Active low digital output signal driven from the same internal signal that enables the output amplifier.
12	RST*	Reset	Active low digital input.
13	VSS_A	Analog ground	Analog ground reference.
14	SO	Signal output	Analog signal output. Tristate enabled with internal TS signal.
15	C2	Capacitor 2	Connection for 680pF capacitor to ground.
16	C1	Capacitor 1	Connection for 680pF capacitor to ground.
17	SI	Signal input	Analog signal input.
18	VDD_A	Analog supply	5.0 VDC +/- 10% analog supply voltage with respect to VSS_A.
19	TP0	Test point 0	Reserved pin for testing.
20	VSS_D	Digital ground	Digital ground reference.

SSC P300 Node Overview

The SSC P300 is designed to meet the needs of products requiring EIA-600 compatibility. As the SSC P300 uses fewer interface signals than the SSC P400 does, a lower cost host (microcontroller) may be used. Coupling the lower cost host with the low cost of the SSC P300, an EIA-600 compliance node can be added to cost sensitive products. The SSC P300 can transmit and receive all four Data Link services defined in the EIA-600 standard, which allows the designer to select the best Data Link service for the job.

As seen in the block diagram on page 1, a typical node consists of three sections: The first section is the host microprocessor or microcontroller, which is responsible for communicating with the SSC P300 and performing application specific tasks. The second section is the SSC P300 itself. It is responsible for resource intensive Data Link functions and Physical layer services of the protocol. Specific DLL services include transmission and reception of packets, byte-to-symbol conversion for transmitted packets, symbol-to-byte conversion for received packets, transmit channel access (based on packet priority and EIA-600 access rules), and CRC generation and checking. The last section is the power line analog functions. These functions include: coupling the signal frequencies onto the medium, amplification of the transmitted signal to drive the impedance of the medium, and input filtering of the incoming signal.

The host communicates with the SSC P300 by issuing commands. These commands provide for the initialization and verification of the node's operating mode and addresses, for the transmission and reception of packets, and for the return of status information. In general, the host must provide the following functions in order to utilize the SSC P300:

1. Initialization routine.
2. Routine to write commands out to the chip.
3. Routine to read data from the chip.
4. Interrupt service routine.

The SSC P300 can be placed into one of three operating modes: Data Link Layer (DLL) mode, Controller (CON) mode, and Monitor (MON) mode. In the DLL mode, the P300 will manage all address matching, and timer resources. In the CON mode, the P300 converts the incoming signal into bytes. It becomes the responsibility of the host to manage address matching and timer resources. The MON mode monitors the medium. Any packet detected on the medium is passed up to the host regardless of the packet's address or type.

DLL Mode

The DLL mode supports the standard CEBus Data Link Layer functionality at the Logical Link Sublayer. (See Table 1). Most DLL mode functions directly map to the primitives specified in the CEBus Logical Link Sublayer specification. Detailed control and status information is also provided. The P300 supports all four Data Link services: Unacknowledged, Acknowledged, Addressed Unacknowledged, and Addressed Acknowledged services. The DLL mode is selected by writing a value of 00 to the *Service_Level* field of *Mode_Control* byte in the *Layer_Config_Info* structure. See the section "P300 Data Structures" later in this document.

Table 1. EIA-600 Device Functionality

EIA-600 Data Link Layer Function Description	P300 Implementation
Transmit and receive all of the Data Link layer packet types: ACK_DATA, UNACK_DATA, ADRACK_DATA and ADRUNACK_DATA.	One receive and one transmit address sequence number association is maintained.
Maintain and supply system and node addresses and layer system management parameters as requested from a higher layer.	Address and layer management parameters are volatile. Restoration following power loss is the responsibility of the attached host processor.
Recognize own system and node addresses and the broadcast address (destination system and node addresses = 0x0000, or destination system address = own system address and destination node address = 0x0000).	Fully implemented.
Recognize group addresses.	One group address is supported.

Receive Packet Procedure in DLL Mode

The following assumes a valid correctly addressed packet has been received. The various packet receive conditions are given below. References to Rc_Dest_Address and Rc_Source_Address refer to the respective four bytes comprising the Rc_Dest_Device_Code and Rc_Dest_House_Code, and the Rc_Source_Device_Code and Rc_Source_House_Code as found in the *Receive_Header* data structure.

Host Interaction for Packet Reception in DLL Mode

Packet acceptance is posted for the host processor by using the *Rc_Attn* and *Rc_Except* flags in the *Interface_Flags* byte. The host uses the *Interface_Flags* to determine what action to take to service the SSC P300. When *Rc_Attn* flag is TRUE, the SSC P300 generates a host processor interface attention signal to asynchronously alert the host that service is required for a received packet. *Rc_Attn* is set to TRUE for all received packets accepted and not discarded due to refusal, overrun or duplicate receipt. The *Rc_Except* flag is also set if an ACK_DATA or ADRACK_DATA packet was refused, an UNACK_DATA or ADRUNACK_DATA packet overrun occurred, or if the current receive packet has a different source and/or destination address than the previous receive packet. *DLL_Rc_Link_Status* data structure may be accessed to determine the cause of the *Rc_Except* flag assertion.

A packet that is accepted and not discarded as a duplicate will cause the buffered *Receive_Header_Info* structure (Control field, Destination address and Source address) to be updated from the corresponding fields of the current *Receive_Header_Info* structure. If no packet is pending, that is the host has read all packets, then the *Receive_NPDU_Field* structure is updated to complete the reception of the packet.

Whether or not discarded as a duplicate, an accepted ADRACK_DATA/ACK_DATA packet results in the SSC P300 sending an ADRIACK/IACK acknowledgment.

Packet Acceptance or Refusal

The following describes the conditions under which a valid correctly addressed packet is accepted or refused/rejected.

Packet Refusal and Overrun

The following describes the conditions under which a valid correctly addressed packet is refused. If the previously received packet has not yet been serviced by the host, any incoming ACK_DATA or ADRACK_DATA packet will be refused and *Rc_Refused* will be posted in *DLL_Rc_Link_Status* structure. Any incoming UNACK_DATA or ADRUNACK_DATA packet will cause an *Rc_Overrun* to be posted in *DLL_Rc_Link_Status* structure. A packet which produces a refuse or overrun condition does not affect the current receive sequence timer. An ADRACK_DATA/ACK_DATA packet that is refused results in an ADRIACK/FAILURE (busy) packet being returned to the source. The source then knows that delivery was unsuccessful and, if able, can retry the transmission.

Receive Sequence Timer Expired

If the receive sequence timer has expired, the packet will be accepted. If the received packet was ACK_DATA or UNACK_DATA, the receive sequence timer is left clear (zero). If the received packet was ADRACK_DATA or ADRUNACK_DATA, the receive sequence timer is reset to approximately 937 ms (1.25 times the Maximum Receive Time-out value of 750 ms).

Receiver Sequence Timer Not Expired

Same Source and Destination Address - If the receive sequence timer is not expired, but the received packet is ADRACK_DATA or ADRUNACK_DATA with a source and destination address that matches the previously received packet, the packet is accepted. The packet is then checked for a duplicate by using the received packet's sequence number. If the received packet sequence number matches the previously received sequence number, the received packet is a

duplicate and is discarded. If discarded, the receive sequence timer is not modified. If the packet is not a duplicate and is accepted, the receive sequence timer is reset to 937 ms.

Different Source or Destination Address - If the received packet was ACK_DATA or UNACK_DATA, and the receive sequence timer is expired, the packet will be refused (ACK_DATA) or overrun will occur (UNACK_DATA). The appropriate flag (*Rc_Refused* or *Rc_Overrun*) will be posted. If the received packet is ADRACK_DATA or ADRUNACK_DATA and, if the received source address and/or the destination address does not match the respective previously received packet addresses, the packet will also be refused (ADRACK_DATA) or overrun will occur (ADRUNACK_DATA). The appropriate flag (*Rc_Refused* or *Rc_Overrun*) will be posted.

Transmit Packet Procedure in DLL Mode

The host processor is responsible for determining that no outstanding transmission is in progress before requesting a new packet transfer and transmission (see section, Posting Packet Transmission Status below). The SSC P300 supports only a single buffered transmit packet so that the previous transmission must be complete before a new packet can be transferred from the host.

Packet transmission can be accomplished by writing a complete packet with the PT command, or by using a combination of the WTH and WTI commands that transfer the *Transmit_Header_Info* structure (LPDU header) and *Transmit_NPDU_Field* structure (NPDU packet body) respectively.

The P300 uses a transmit destination association sequence timer to insure that new packets are not sent to a device whose receive time-out may not yet have expired from a previous transfer from this node. Thus packets sent to a new or different address may be delayed by the DLL waiting for the expiration of the transmit sequence timer. Successive packets to the same destination address, however, will be sent without incurring possible transmit sequence time-out delays. The P300 determines implicitly from the particular transmit commands used by the host processor if a packet with a (potentially) new destination address is to be transmitted. A command that writes the packet header containing the destination address (i.e., WTH or PT) is assumed to contain a change in the packet destination address. If, however, only the packet body (NPDU) is written by a command (i.e., WTI), then the packet must necessarily be to the same destination address using the same packet header as the last packet transmitted.

The use of WTI commands following an initial WTH or PT commands are recommended, especially for ADRACK_DATA or ADRUNACK_DATA type packets. This will allow the maximum transfer rate when sending multiple packets to the same device by using multiple WTI commands following the initial WTH or PT command. ACK_DATA or UNACK_DATA type packets do not set the transmit sequence timers, but the previous transmit sequence timer value must be zero before the P300 initiates the ACK_DATA or UNACK_DATA transmission.

The SSC P300 can process two packets: one that is currently transmitting and one in the buffer. A third packet cannot be sent to the SSC P300 until the transmit association timer for the first packet has expired.

Writing a Transmit Header

The host, using the WTH command, writes packet Transmit_Header_Info. Both the system and device source address values transferred in the *Transmit_Header_Info* should normally be 0xFFFF, resulting in the P300 substituting its current system and device address as the header packet source address. The P300 assumes that writing a header with the WTH command changes the destination address, therefore, actions are taken to preserve transmit time-outs. When the WTH command is executed, the following steps occur:

1. If the transmit sequence timer is not expired, and the transmit service is addressed-type, the failure REMOTE_BUSY_XMIT_LIST status is returned, and the WTH command is ignored.

2. If the transmit sequence timer is expired, or the transmit service is non-addressed-type, the data is transferred to the P300; source addresses are substituted as required and control is returned to the host.

3. The host may now transfer the *Transmit_NPDU_Field* and initiate packet transmission via the WTI command (see Transmitting a Packet with an Existing Header below).

Transmitting a Packet with an Existing Header

To transmit a packet with a valid header previously transferred by a WTH or PT command, the *Transmit_NPDU_Field* is written by the host with the WTI command. The existing *Transmit_Header_Info* contents are used for the header of the packet to be transmitted. Subsequent packets addressed to the same device (i.e., using the same *Transmit_Header_Info*) only require a WTI command to send the new packet body (NPDU) portion of the subsequent packet. When the WTI command is executed the following steps occur:

1. If the transmit sequence timer is not expired, and the transmit service is ADR-type, the failure REMOTE_BUSY_XMIT_LIST status is returned, and the WTI command is ignored. If the transmit sequence timer is expired, or if the transmit service is non-ADR-type, the packet body data are transferred to the P300 and control is returned to the host. If the packet is ADRACK_DATA or ADRUNACK_DATA, the *Tr_Seq_#* is incremented, and inserted in the transmit header. The *Tr_Seq_#* is a Data Link parameter and completely maintained by the P300. If the P300 automatically generates duplicate packets based on *CH_ACCESS_NUM* (for ADRUNACK_DATA packets), the same *Tr_Seq_#* value will be used for all subsequent copies of the same packet. Thus, if more than one packet is received at the destination, the duplicates can be eliminated.

2. The *Tr_Except* flag is set to indicate transmission in process.

3. When the packet transmission starts (i.e. the channel access protocol has been satisfied) and if the packet is an ADRACK_DATA or ADRUNACK_DATA type, the transmit sequence timer is loaded with 1125 ms. On any subsequent retransmission of an ADR-type packet, due to ADRIACK (remote busy) or the CH_ACCESS_NUM parameter, the transmit sequence timer is reloaded with 1125 ms.

4. Upon transmission completion, *Tr_Attn* is asserted. *Tr_Except* will reflect the success or failure of the transmit process.

Transmit Complete Packet

To transmit a complete packet with a single command, the *Transmit_Header_Info* and *Transmit_NPDU_Field* are written consecutively by the host with the PT command.[1] When the PT command is executed the following steps occur:

1. The transmit header and packet body data is transferred to the P300, source address substitution is made as required and control is returned to the host. If the packet is an ADRACK_DATA or ADRUNACK_DATA, the *Tr_Seq_#* is incremented, and inserted in the transmit header. The *Tr_Seq_#* is a Data Link parameter and completely maintained by the P300. If the P300 automatically generates duplicate packets based on *CH_ACCESS_NUM* (for ADRUNACK_DATA packets), the same *Tr_Seq_#* value will be used for all subsequent copies of the same packet. Thus if more than one packet is received at the destination, the duplicates can be eliminated.
2. The *Tr_Except* flag is set to indicate transmission in process.
3. The P300 then waits for the transmit sequence timer to reach zero (the transmit sequence timer may have a residual value due to the previous transmission of an ADRACK_DATA or ADRUNACK_DATA packet), if it has not already done so. When the transmit sequence timer has expired, the transmission process begins.
4. When the packet transmission starts (i.e. the channel access protocol has been satisfied) and if the packet is an ADRACK_DATA or ADRUNACK_DATA type, the transmit sequence timer is loaded with 1125 ms. (1.5 times the Maximum Receive Time-out value of 750 ms.)
5. Upon transmission completion, *Tr_Attn* is asserted. *Tr_Except* will reflect the success or failure of the transmit process.

ACK_DATA or UNACK_DATA Packet Transmission

Use of UNACK_DATA or ACK_DATA packet transmission is generally discouraged. The principal use of such packet types can be for broadcast messages in which sequencing is unimportant, e.g. “hailing”. UNACK_DATA and ACK_DATA packet transmissions do not set the transmit association timer, but transmission of these types still require the expiration of any outstanding transmit association timer value.

Posting Packet Transmission Status

In-progress, completion and success of a transmission are posted in the *Interface_Flags*, using the *Tr_Attn* and *Tr_Except* flags. The host can interrogate these flags to determine what action is necessary. The *Tr_Attn* flag is set TRUE when the transmit sequence timer becomes zero after a WTH command is executed, or when a packet transmission, including multiple accesses, if specified, has completed after a PT or WTI command. The *Tr_Except* flag is always TRUE when a packet transmission is in progress following a PT or WTI command, or the transmit sequence timer has not expired subsequent to a WTH command. *Tr_Except* is also set whenever *Tr_Attn* is set TRUE and a transmit exception condition has occurred.

When the *Tr_Attn* flag indicates a packet transmission is complete, a new packet can immediately be sent to the same destination address by using a WTI command. However, transmitting a packet to a different (new) address may cause the P300 to wait until the current transmit sequence timer has expired. The P300 assumes that executing a WTH command changes the destination address. Consequently, execution of a WTH will set *Tr_Except* to TRUE and *Tr_Attn* to FALSE if the transmit sequence timer has not expired. When the transmit sequence timer expires, *Tr_Attn* will then be posted as TRUE. If the transmit sequence timer has already expired, *Tr_Attn* will be set TRUE immediately. Thus a WTI command should not be executed to send the new packet after a WTH command until *Tr_Attn* is set to TRUE.

[1] A null length packet transfer with a PT or WTI command will retransmit the previous packet if transmission is not in process. Note that this is equivalent to transmitting a completely new packet (i.e., Data Link sequence numbers are incremented), and consequently should not be used for higher level CEBus (e.g. application) transmissions, as the new packet application level sequence information (i.e., Invoke ID number) is not changed.

If a packet write command (PT, WTI or WTH) is issued while the P300 has an outstanding transmission in progress (*Tr_Attn* is FALSE and *Tr_Except* is TRUE), the command shall be ignored and an *Interface_Error* flag will be posted in the *Interface_Flags* byte.

Host Processor Busy Processing

Host Processor Busy – The host Indicates that it does not have resources available to service a new packet by setting the *Host_Busy* flag. If the *Host_Busy* flag is set TRUE by the host, the P300 will not generate an attention sequence in response to a *Rc_Attn* or *Tr_Attn* condition. All ongoing Data Link operations continue as normal, including any packet refusals or overrun conditions that may occur during the period in which the host is busy. When the host leaves the busy condition and notifies the P300 by setting the *Host_Busy* flag to FALSE, any pending receive or transmit conditions (*Rc_Attn* or *Tr_Attn*) will cause an immediate attention sequence to be generated.

Upper Layer Busy - Indicates the host cannot process receive packets at this time. No attention requests for received packets are generated (the first received packet is stored—subsequent received packets are refused/ignored). Other attention requests (including transmit complete) are generated normally. If an attention request is generated due to a non-"received packet" reason, but a packet has been received, it is the responsibility of the host to either process the received packet immediately or store the fact and process the received packet later. See the description in the Data Structures section for information on the *Interface_Flags.* See the Attention Sequence section, which pertains to processing all set flags.

Receive Association Limitation

One receive source association variable is provided by the P300. This limits the attached host to receiving packets from only one source address and one destination address[2] within any given receive time-out period. ADRACK_DATA and ACK_DATA packets received from a different source address or going to a different destination address are given a ADRIACK/FAILURE (busy) if the receive sequence timer has not expired. ADRACK_DATA packets can subsequently (automatically) be retransmitted by the source, and so are not lost. However, the source will not know to retransmit the ADRUNACK_DATA and UNACK_DATA packets when the receive sequence timer or other busy condition is over. Consequently, ADRUNACK_DATA and UNACK_DATA packets may be lost. An occurrence of a potentially “lost” packet is noted by the P300 posting the *Rc_Overrun* flag in *DLL_Rc_Link_Status* structure and by setting the *Rc_Except* flag in Interface_Flags to TRUE.

DLL Timing Constraints

In high traffic situations, the real time communication constraints of servicing the SSC P300 must be considered. A worst case condition occurs when a second packet, addressed to the node, is being received before the host has read the previous packet. In order to prevent packet overrun, the SSC P300 must be serviced as soon and as quickly as possible. A typical calculation for the service turn-around time uses the minimum quiet time between packets, a “typical” preamble, an IEOF, and null address fields. Adding up these values yields a service turn-around time of approximately 4ms. This time should be adequate for CEBus compliance cases.

Controller (CON) Mode

In this mode, a host may support communications to multiple devices simultaneously using the ADR-type transmit services. This simultaneous device support requires the maintenance of separate transmit/receive address associations for each device in the host process.

The CON Mode is selected by the value 11 in the *Service_Level* field of the *Mode_Control* byte in the *Layer_Config_Info* structure*.*

Packet Receive in CON Mode

[2] Four different destination addresses exist : 1) system address/node address, 2) system address/group address, 3) system address/broadcast node address, and 4) broadcast system address/broadcast node address.

Received packets are transferred to the host in a manner allowing the host to determine the addressing status of the packet and manage the sequence number and association timers for source and destination addresses. The notification of a received header is via *Rc_Except* in *Interface_Flags*. After receiving the header via the RRH command and processing the *Received_Header_Info* structure, and prior to the receipt of the CRC, the host makes a determination whether the packet can be accepted, rejected, failed busy or ignored. This determination is based on the state of the received service, sequence number, and receive association timer. The host must notify the P300 controller via *Rcv_Disposition* in *Node_Control* structure as to the suitable disposition of the packet such that appropriate action at the end of the packet can be performed, including sending an IACK packet. If a *Rcv_Disposition* response is not received in time, the packet will be ignored. If the packet has been accepted, the Rc_Attn is posted to notify the host of the receipt of the completed CRC (end of packet). This information can be used by the host to manage the receive association timer. The received NPDU is conveyed via the RRI (preferred) or PR command. The host is responsible for logging the received NPDU event for association timer purposes.

Packet Transmit in CON Mode

The host sets up the address information for transmission, including the state of the sequence number, and manages the transmit association timers. Information regarding actual channel access (beginning of packet) will be passed to the host to assist in transmit association timer management via *Tx_Except*. Re-transmission (immediate and multiple ADR-type accesses) are handled by the P300 controller, based on the transmitting packet's source address fields. Transmission completion is handled as described for the DLL mode.

WTH and WTI are not recommended commands for transmitting in this mode, because WTI will retransmit without changing the sequence number.

CON Timing Constraints

When a data packet is received, the host is responsible for determining the acceptance or rejection of the packet before the NPDU is completely received. The worst case occurs when the ID packet is received. In this case, the NPDU contains only one byte. This yields 1.2 ms for the host to acknowledge the received packet after the attention sequence.

Monitor Mode

The MON Mode is selected by writing a value 10 in the *Service_Level* field of *Mode_Control* byte in the *Layer_Config_Info* structure.

Packet Receive in MON Mode

In this mode, the SSC P300 will receive and forward to the host all packets on the channel. A packet must have a legitimate start (IEOF), valid header, valid end (EOP) and valid CRC. No address or type checking is done and packets addressed to this device will not be acknowledged or sequence number checked. Further, all packets have the end of packet time (the time of the last UST of the CRC) logged, which is readable by the host. This time is posted in internal clock cycles (1.5625 µs per count) and is in a free-running 32-bit format.

Packet Transmit in MON Mode

Any packets transmitted in the MON mode will not be checked for type and will be transmitted as if they were UNACK_DATA packets. The time of the last UST of the transmitted packet is also logged so that all channel packet traffic timing is available to the host. If a packet type is transmitted that generates an acknowledge from the receiver, the acknowledgment will be logged and forwarded to the host just like any other packet received in the MON Mode. NOTE: Transmitting in MON mode is not recommended.

MON Timing Constraints

The host must service the received monitor packets before the next packet on the channel is received in order to avoid an overrun. In the worst case scenario, this means that in the monitor mode, the host must respond to an attention sequence within about 1 ms and must be able to support the packet transfer protocol at or near the maximum rate (see the section Signal Timing). If the response time requirements cannot be met or the maximum transfer rate cannot be supported, received packets may be lost and/or overwritten.

SSC P300 Power Line Interface

Analog data is transferred between the AC power line and the SSC P300 over the Signal In (SI) and Signal Out (SO) pins. When transmitting, SSC “chirps” from the SSC P300 SO pin are routed to the output amplifier, which is enabled by the SSC P300 Tristate (TS) signal. Once amplified, the output signal passes through a low-pass output filter, which removes harmonic energy (distortion) from the transmit signal. The signal then passes to the Tristate switch. This switch is enabled by the SSC P300 TS signal and serves to isolate the amplifier and filter from the power line coupling circuit during receive operation. When the Tristate switch is enabled, the power line communication signal is routed to the 60 Hz power line through the power line coupling circuit.

When receiving, the power line analog signal passes through the external power-line-coupling network to the input filter. This bandpass filter passes the chirp frequency band (100 to 400 kHz) to the SSC P300 SI input.

The output amplifier provides the power necessary to drive the impedance of the power line. The output amplifier is easily implemented using Intellon’s P111 PL Media Interface IC, as illustrated in Figure 1. The P111 provides greater signal output power than previous discrete designs and replaces as many as 30 discrete components. This increases reliability and requires smaller board size than commonly used discrete implementations.

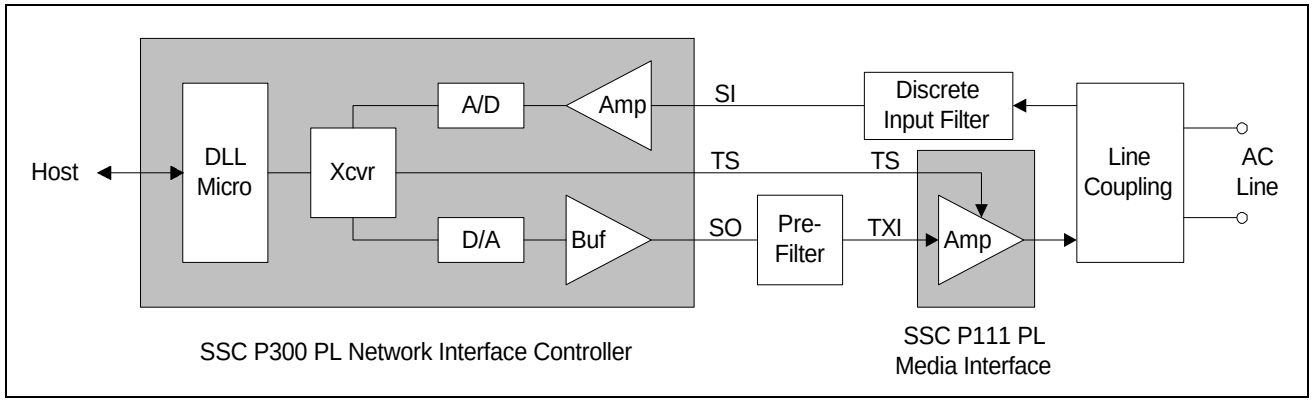


Figure 1. SSC P300 with P111 PL Media Interface IC

SSC P300 Host Interface

Host Signal Description

SSC P300 interface to the Host system is supported through a Serial Peripheral Interface (SPI) using five I/O lines. A hardware, active-low, reset signal is also supplied by the host system. These signals are INT*, CS*, SCLK, SDI, SDO, and RST*. Their description is given below:

INT* - The P300 pulls this line low whenever service from the host is required. The three most common conditions that generate an interrupt are packet reception, packet transmission, and ready for next SPI byte transfer.

CS* - The host pulls the chip select line low to start communication with the P300. This line remains low during the read or write command operation, which is called a transaction. The host sets this line high when the transaction is completed.

SCLK - The host generates a clock signal to synchronize the data transfer. When the host is writing data, the P300 reads the data on the rising edge of the clock. When the host is reading data, the P300 places data on serial data output (SDO) pin on the falling edge of the clock and the host reads the data on the rising edge of the clock. See section 6.0, Transaction Timing.

SDI - The data generated by the host is placed on this line and clocked into the P300. The data must be valid on the rising edge of the clock.

SDO - The data generated by the P300 is placed on this line on the falling edge of the serial clock and is valid on the rising edge of the clock. The setup and hold times for the P300 must be met if the data is to be valid on the rising edge of the clock.

RST* - This signal line is pulled low to provide a hardware reset.

Table 2. Summary of SPI Pin Descriptions

Mnemonic	Name	Direction	Use
INT*	Interrupt	Output	Active low. Used to indicate an attention request (packet received or transmit packet completed) or "ready for SPI byte transfer."
CS*	Chip Select	Input	Active low. Must be asserted during a read or write command operation ("transaction").
SCLK	Shift Clock	Input	Used to synchronize data transfer.
SDI	Serial Data In	Input	Serial data input (normally connected to host's SDO signal). Data is shifted MSB first.
SDO	Serial Data Out	Output	Serial data output (normally connected to host's SDI signal). Data is shifted MSB first.
RST*	Reset	Input	Reset signal line. The host may assert this signal low (open collector drive) to provide a hardware reset.

A simple protocol is used to transfer commands and data between the host and SSC P300. These commands and data include packets to be transmitted, received packets, status and configuration information.

Command Format

The host processor and the P300 have a master/slave relationship. That is, the P300 cannot send data to the host until commanded to do so. The P300 can only request service from the host, via the interrupt signal. The host responds to the interrupt with a command that allows the SSC P300 to return its data. Commands from the host allow the host to read or write the internal registers of the P300. These registers are grouped together into data structures, which are used for configuring the chip, setting parameters of the Data Link, returning status information, receiving packet information, and transmitting packets. See the Commands section for more detailed information about the commands and their use. The command formats are classified as either a write transaction or a read transaction. A write transaction is composed of a command byte, a length byte, and the data bytes to be written into the data structure. A read transaction causes the P300 to transfer data to the host from a data structure. When reading, the command byte is sent from the host to the SSC P300. The SSC P300 will return a length byte followed by the data bytes from the data structure. All SSC P300 system information is loaded by the host processor following the procedure illustrated in Figure 2. All downloaded information is volatile and must be reloaded in case of a power failure. Data is transferred from the SSC P300 using the read command format shown in Figure 3. For timing information, see the section on Transaction Timing.

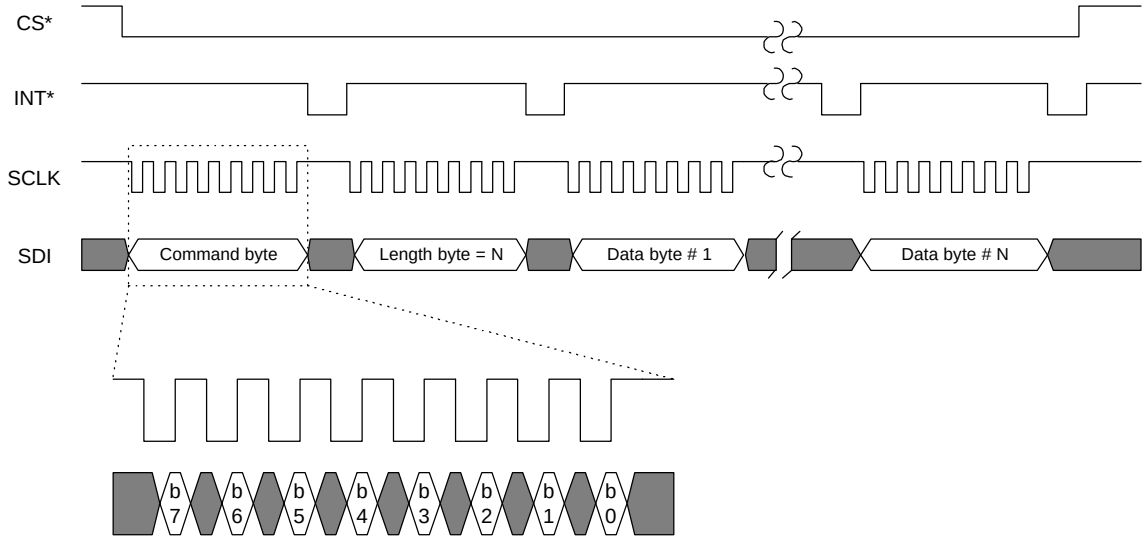


Figure 2. Write Command Format

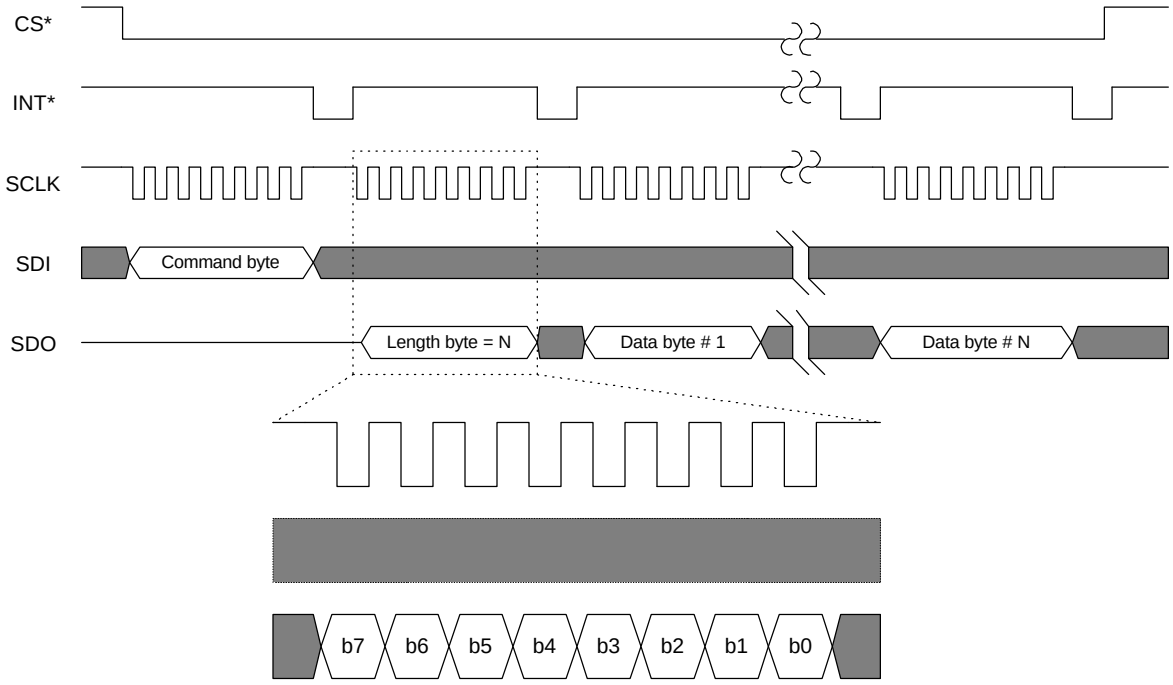


Figure 3. Read Command Format

Flowcharts for write and read transactions are given in Figures 4 and 5.

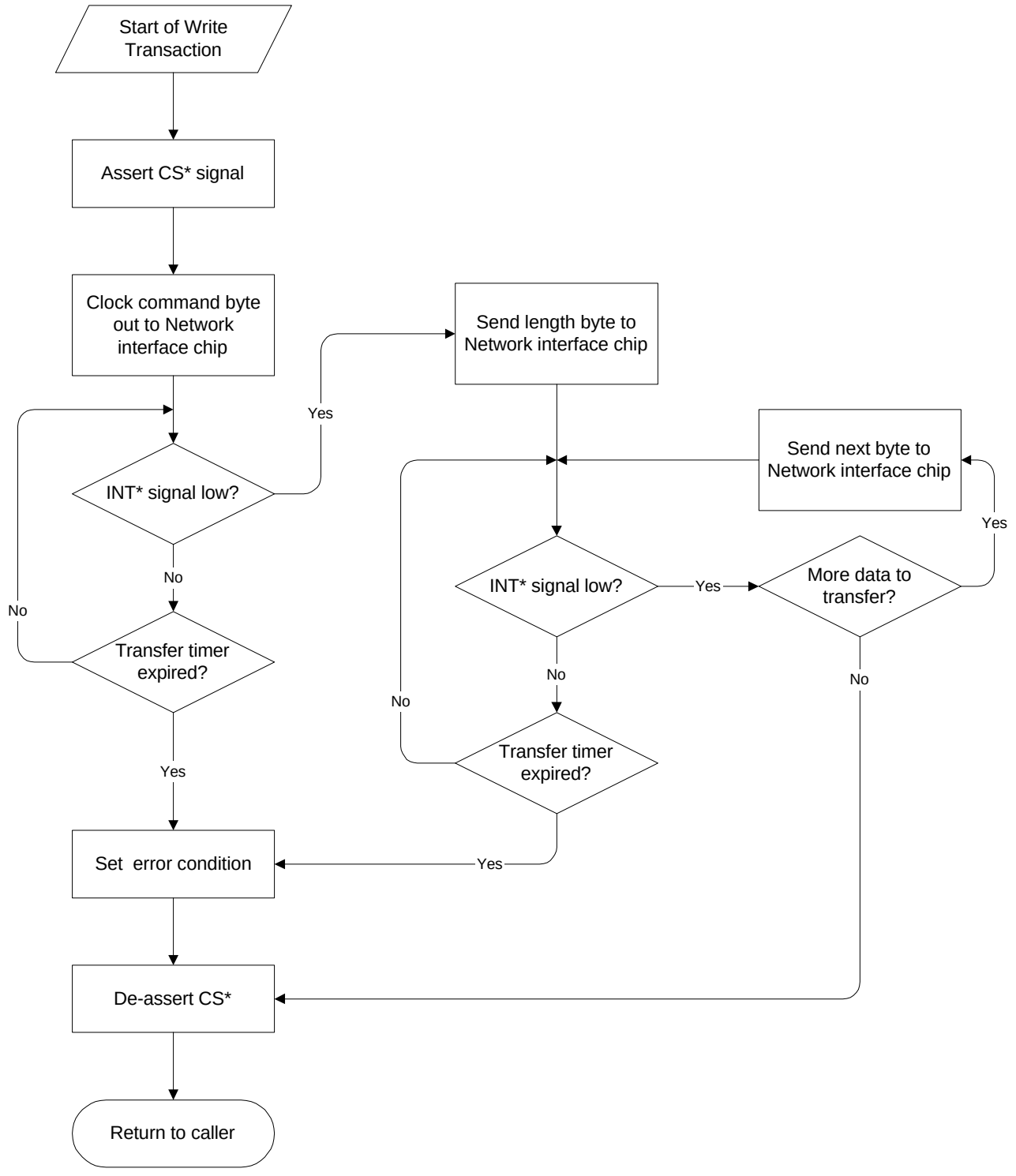


Figure 4. Write Transaction Flow Chart

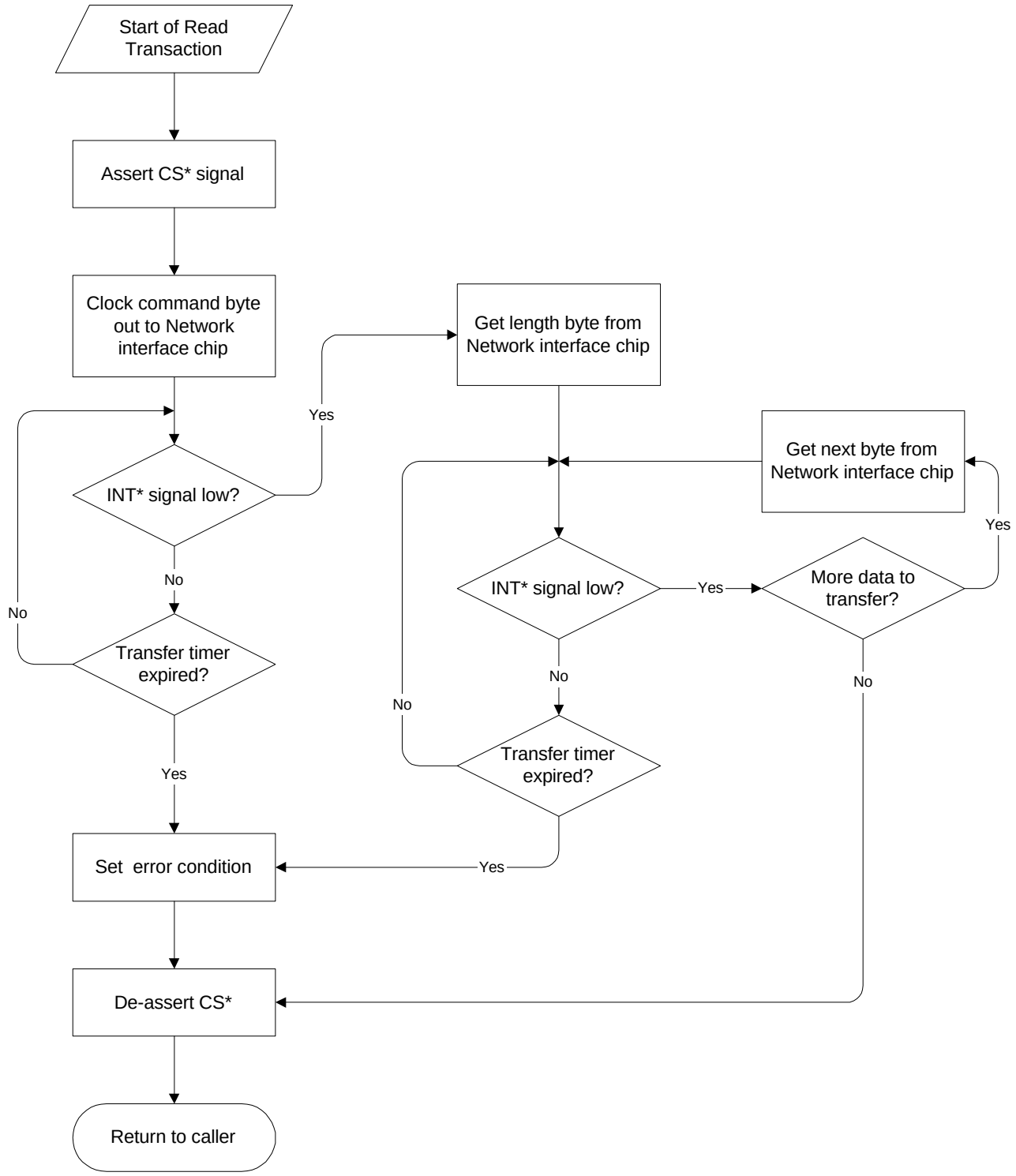


Figure 5. Read Transaction Flow Chart

SSC P300 Commands

Command Overview

A summary of information needed to use each command is given in Tables 3A and 3B. The information in these tables defines the command's name, mnemonic and hex value. The command is then further classified as belonging to a write transaction or a read transaction. The length defines how many bytes of information are transferred. The affected data structure column relates the command to the data structure affected by that command.

Table 3A. Summary of Commands

Name	Mnemonic	Hex Value	Transaction Type	Length	Affected Data Structure
Force_Interface_Error	FIE	0x00	Read	0	None
Reset	RST	0x01	Write	0	None
Layer_Management_Read	LR	0x02	Read	7	*Layer_Config_Info*
Layer_Management_Write	LW	0x03	Write	7	*Layer_Config_Info*
Interface_Read	IR	0x04	Read	1	*Interface_Flags*
Control_Write	CW	0x05	Write	2	*Node_Control*
Status_Read	SR	0x06	Read	6	*Status_Info*
Packet_Receive	PR	0x08	Read	Variable	*Receive_Header & Receive_NPDU*
Packet_Transmit	PT	0x09	Write	Variable	*Transmit_Header & Transmit_NPDU*
Read_Receive_Information	RRI	0x0A	Read	Variable	*Receive_NPDU*
Write_Transmit_Invoke	WTI	0x0B	Write	Variable	*Transmit_NPDU*
Read_Receive_Header	RRH	0x0C	Read	9	Unbuffered *Receive_Header*
Write_Transmit_Header	WTH	0x0D	Write	9	*Transmit_Header*
Write_Register_46	WRS-46	0x46	Write	1	*DLL_Access_Control*
Read_Register_4	RRS-4	0x84	Read	1	*DLL_Rc_Link_Status*

Table 3B. Command Description

Mnemonic	Description
FIE	This command sets the interface error flag in the *Interface_Flags data* structure. This command is typically used for testing and debugging. The host performs a read transaction with the command byte set to 0x00. The SSC P300 returns a length byte of zero.
RST	This command disables all functions and I/O except the host interface. The host performs a write transaction with the command byte set to 0x01 and the length byte set to zero. This command can be used after an error condition to reset the chip to its initial state. After this command is issued, the chip must be re-initialized as described in the Initialization section.
LR	This command returns the information stored in the *Layer_Config_Info* data structure. The host performs a read transaction with the command byte set to 0x02. The P300 will return a length byte set to 7 followed by the 7 data bytes of the *Layer_Config_Info* data structure. This command is typically used to verify the node configuration parameters. See section on Initialization.

Mnemonic	Description
LW	This command sends 7 bytes of data to be stored in the *Layer_Config_Info* data structure. The host performs a write transaction with the command byte set to 0x03 and a length byte set to 7 followed by the information to fill the *Layer_Config_Info* data structure starting with byte 0. It is necessary to set up the *Layer_Config_Info* structure before the P300 can transmit and receive properly. See section on Initialization.
IR	This command returns the information stored in the *Interface_Flags* data structure. The host performs a read transaction with the command byte set to 0x04. The P300 returns a length byte set to 1 followed by the one byte from the *Interface_Flags* data structure. This command is typically issued after an attention sequence to determine the cause of the attention sequence. See the Attention Sequence section. Reading the *Interface_Flags* data structure clears all flags; therefore, the host must process all of the returned flags.
CW	This command sends 2 bytes of data to be stored in the *Node_Control* data structure. The host performs a write transaction with the command byte set to 0x05 and a length byte set to 2 followed by the information to fill the *Node_Control* data structure starting with byte 0. The most common use of this command is issuing a receive disposition when in CON mode.
SR	This command returns the information stored in the *Status_Info* data structure. The host performs a read transaction with the command byte set to 0x06. The P300 will return a length byte set to 6 followed by the 6 data bytes of the *Status_Info* data structure. This command is typically used to determine the success or failure of a transmitted packet.
PR	This command returns the information stored in the buffered *Receive_Header* data structure and the *Receive_NPDU* data structure, which together comprise the received packet. The host performs a read transaction with the command byte set to 0x08. The P300 will return a length byte set to the number of bytes in the received packet. The maximum length is 41 bytes. Packet_Receive command can always be used in response to a received packet. However, host interface traffic can be reduced if, after finding the Rc_Addr_Diff flag FALSE in the *DLL_Rc_Link_Status* data structure, the RRI command is used to retrieve the data from the *Receive_NPDU* data structure.
PT	This command places data into the *Transmit_Header* and *Transmit_NPDU* data structures. The host performs a write transaction with the command byte set to 0x09 and the length byte set to the number of bytes in the packet with a maximum length of 41 bytes. The length byte is followed by the data bytes to be placed in the *Transmit_Header* and *Transmit_NPDU* data structures. If the transmitted packet is ADRACK or ADRUNACK, the transmit association timer is set or reset to 1.125 seconds. Each time this command is issued, the P300 assumes the transmission to be a different remote device and the transmit association timer must be expired to accept the command. Issuing the PT command when a previous transmission is in progress will cause the P300 to ignore the command and the P300 will assert the Interface_Error flag in the *Interface_Flags* data structure.
RRI	This command returns the information stored in the *Receive_NPDU* data structure. The host performs a read transaction with the command byte set to 0x0A. The P300 will return a length byte set to the number of bytes in the *Receive_NPDU*. The maximum length is 32 bytes. By using this command host interface traffic can be reduced. Before issuing this command, check the Rc_Addr_Diff flag in the *DLL_Rc_Link_Status* data structure is FALSE. The RRI command can then be used to retrieve the data from the *Receive_NPDU* data structure.

Mnemonic	Description
WTI	This command places data into the *Transmit_NPDU* data structure. The host performs a write transaction with the command byte set to 0x0B and the length byte set to the number of bytes to transfer to the *Transmit_NPDU* data structure with a maximum length of 32 bytes. The length byte is followed by the data bytes to be placed in the *Transmit_NPDU* data structure. Before using this command, the information in the *Transmit_Header* data structure must be set up by issuing a Write_Transmit_Header (WTH) command or a Packet_Transmit (PT) command. Once the *Transmit_Header* is setup, the WTI command changes the information in the *Transmit_NPDU* data structure and initiates transmission. By using the WTI command the number of byte transferred to the P300 is reduces, which reduces the host processing time. Issuing the WTI command when a previous transmission is in progress will cause the P300 to ignore the command and the P300 will assert the Interface_Error flag in the *Interface_Flags* data structure.
RRH	This command returns the information stored in the unbuffered *Receive_Header* data structure. The host performs a read transaction with the command byte set to 0x0C. The P300 will return a length byte set to 9 followed by the 9 data bytes of the unbuffered *Receive_Header* data structure. This command is primarily used in CON mode for address matching.
WTH	This command places data into the *Transmit_Header* data structure. The host performs a write transaction with the command byte set to 0x0D and the length byte set to 9 followed by the data bytes to be placed in the *Transmit_Header* data structure. Issuing the WTH command when a previous transmission is in progress will cause the P300 to ignore the command and the P300 will assert the Interface_Error flag in the *Interface_Flags* data structure.
WRS-46	This command sends 1 byte of data to be stored in the *DLL_Access_Control* data structure. The host performs a write transaction with the command byte set to 0x46 and a length byte set to 1 followed by the information to fill the *DLL_Access_Control* data structure. This command sets up parameters that determine how many times and how often the channel is accessed.
RRS-4	This command returns the information stored in the *DLL_Rc_Link_Status* data structure. The host performs a read transaction with the command byte set to 0x84. The P300 will return a length byte set to 1 followed by the information from the *DLL_Rc_Link_Status* data structure. This command is typically used to determine if an overrun or refused condition has occurred.

SSC P300 Data Structures

Table 4 presents a map of the data structures accessible to the host. The following subsections discuss the use of each data structure.

Table 4. Data Structures Map

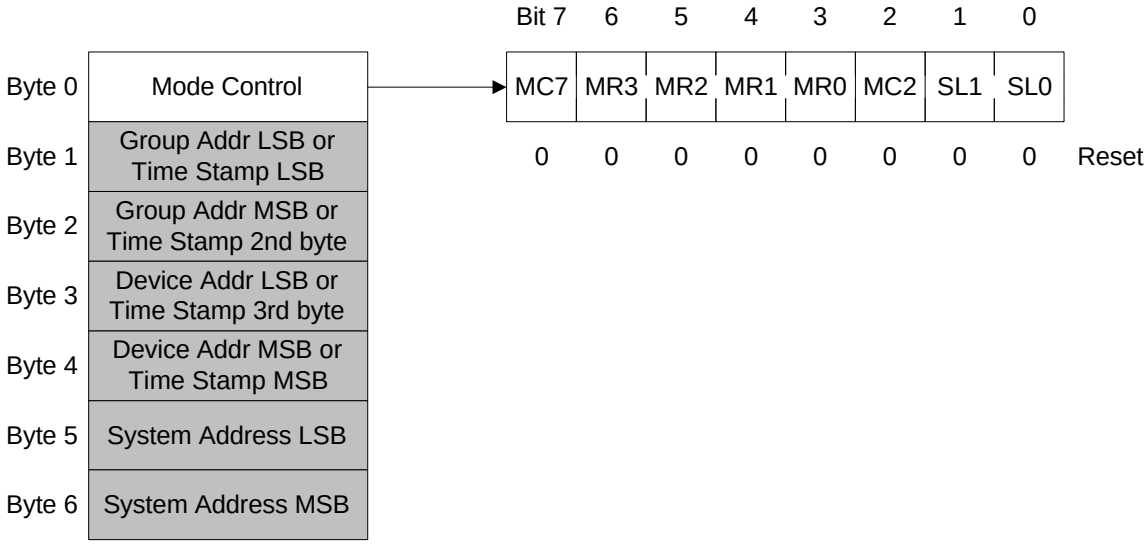
Structure Name (full)	Structure Name (abbr.)	Bytes
Layer Configuration Information	*Layer_Config_Info*	7
Interface Flags	*Interface_Flags*	1
Node Control Flags	*Node_Control*	2
Status Information	*Status_Info*	6
DLL Access Control	*DLL_Access_Control*	1
Data Link Receive Link Status	*DLL_Rc_Link_Status*	1
CEBus Receive Packet Format	*Receive_Packet*	41
CEBus Transmit Packet Format	*Transmit_Packet*	41

Layer Configuration Information Data Structure

The *Layer_Config_Info* structure determines:

- The mode of operation.
- The device's group, device, and system addresses.

The functions of bytes 1, 2, 3, and 4 change depending upon the operating mode. When in the DLL mode, bytes 1 and 2 define the one Group_Address supported by the device. Bytes 3 and 4 define the Device_Address. When in the MON mode, bytes 1 through 4 contains the value of a 32 bit counter. This is the time log of the last transmitted or received packet counted in 1.5625 μs/count. The figures below give the value of the register after reset and is shown by the value next to the word Reset.



Byte 0: ***Mode Control*** determines the maximum number of channel accesses attempted before the transmission is reported as a failure. It also determines the operating mode. The use of each bit field is given below.

MC7: ***Mode Control bit 7*** reserved for future use and must be set to 0.

MR3 - MR0: ***Max_Restarts*** specifies the maximum number of channel accesses attempted before the transmission is reported as a failure to the host (a value of 120 is recommended for moderately to heavily active networks).

MR3	MR2	MR1	MR0	*Max_Restarts*
0	0	0	0	0
0	0	0	1	8
0	0	1	0	16
0	0	1	1	24
0	1	0	0	32
0	1	0	1	40
0	1	1	0	48
0	1	1	1	56
1	0	0	0	64
1	0	0	1	72
1	0	1	0	80
1	0	1	1	88
1	1	0	0	96
1	1	0	1	104
1	1	1	0	112
1	1	1	1	120

MC2: ***Mode Control bit 2*** reserved for future use and must be set to 0.

SL1 - SL0: ***Service_Level*** determines mode of operation.

SL1	**SL0**	***Service_Level***
0	0	Data Link (*DLL*) mode Most applications will use this mode. The DLL mode supports the standard CEBus Data Link Layer functionality at the Logical Link Sublayer. Most *DLL* mode functions map directly to the primitives specified in the CEBus Logical Link Sublayer specification. Detailed control and status information is also provided. The P300 *DLL* mode supports all four Data Link services (UNACK, ACK, ADRACK, ADRUNACK).
0	1	Reserved
1	0	Monitor (*MON*) mode The P300 will receive and forward to the host all packets on the channel. The only requirement is that a packet must have a legitimate start (Preamble EOF), valid header, valid end (EOP) and valid CRC. No address or type checking is done. Further, all packets have the end of packet time (the time of the last UST of the CRC) logged, which is readable by the host. This time is posted in internal clock cycles (1.5625 usec per count) and is in a free-running 32-bit format. Since the packets received in the *MON* mode are not checked for type, packets addressed to this device will not be acknowledged or checked for correct sequence number. NOTE: Any packets transmitted in the *MON* mode will not be checked for type and will be transmitted as if they were UNACK packets. The time of the last UST of the transmitted packet is also logged so that all channel packet traffic timing is available to the host. If a packet type is transmitted that generates an acknowledge from the receiver, the acknowledgment will be logged and forwarded to the host just like any other packet received in the *MON* Mode. Transmitting in *MON* mode is not recommended.

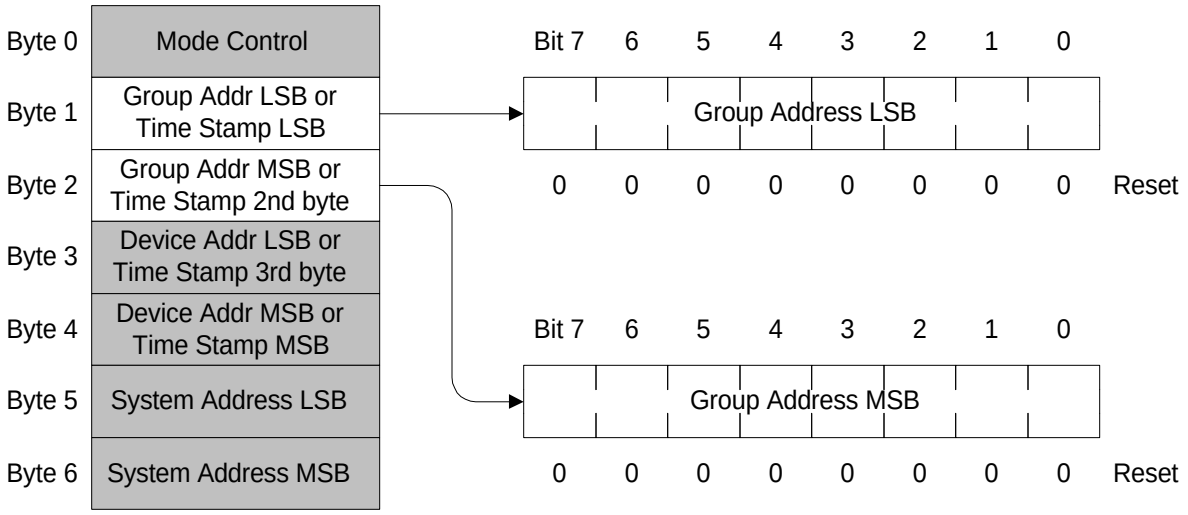
SL1	SL0	***Service_Level***
1	1	Controller (*CON*) mode

In CON mode, the P300 acts as a Medium Access device utilizing the EIA CEBus channel access protocol, transmit service and packet format. The host is responsible for address recognition and may redefine the contents of the address fields within the header and the NPDU (see section 4.6 *Transmit_Packet* Data Structure).

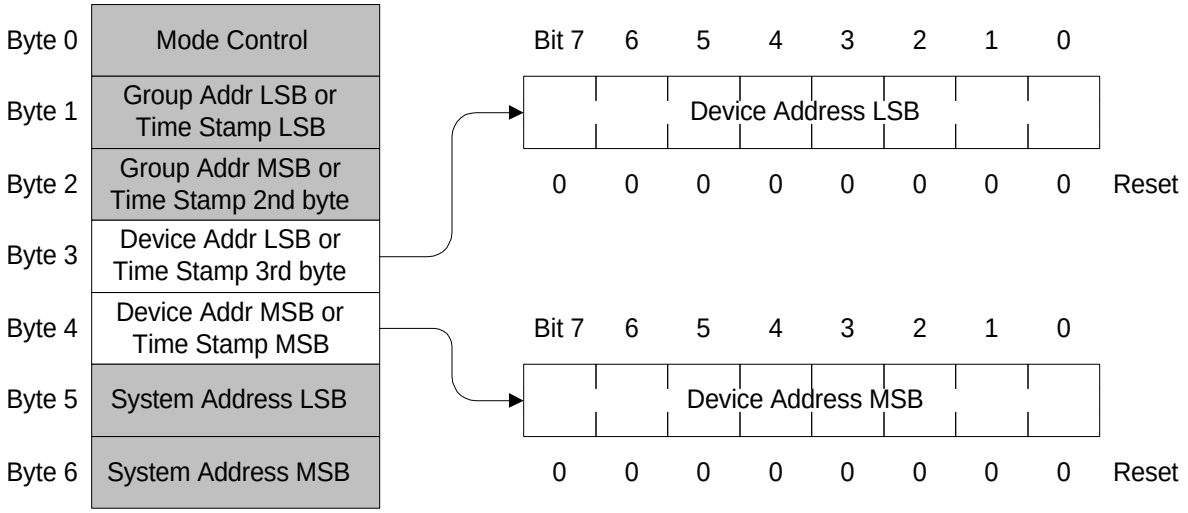
Packet Receive Received packets are transferred to the host without address detection, which requires the host to determine the address status of the packet and manage the sequence number and association timers for source and destination addresses. The receipt of a new *Receive_Header* is posted via the *Rc_Except* flag in the *Interface_Flags* data structure. After retrieving the header via the *Read_Receive_Header* command the host must determine if the packet is to be accepted, rejected or ignored based on the destination addresses, state of the received service, sequence number and receive association timer(s). The host must issue a receive disposition (See *Rcv_Disposition* in *Node_Control* data structure) prior to the receipt of the packet's CRC such that appropriate action at the end of the packet can be performed, including sending an IACK packet. If a receive disposition is not issued in time, the packet will be ignored. If the packet has been accepted, *Rc_Attn* flag will be posted in the *Interface_Flags* data structure and the host can use this event in receive association timer management. The *Receive_NPDU* can be retrieved via the *Read_Receive_Information* (preferred) or *Packet_Receive* command. The host is responsible for logging the received NPDU event for association timer purposes.

Packet Transmit The host sets up the addressing information for transmission, including the state of the sequence number and management of the transmit association timer(s). Information regarding actual channel access (beginning of packet) will be passed to the host to assist in transmit association timer management via the *Tx_Except* flag in the *Interface_Flags* data structrue.

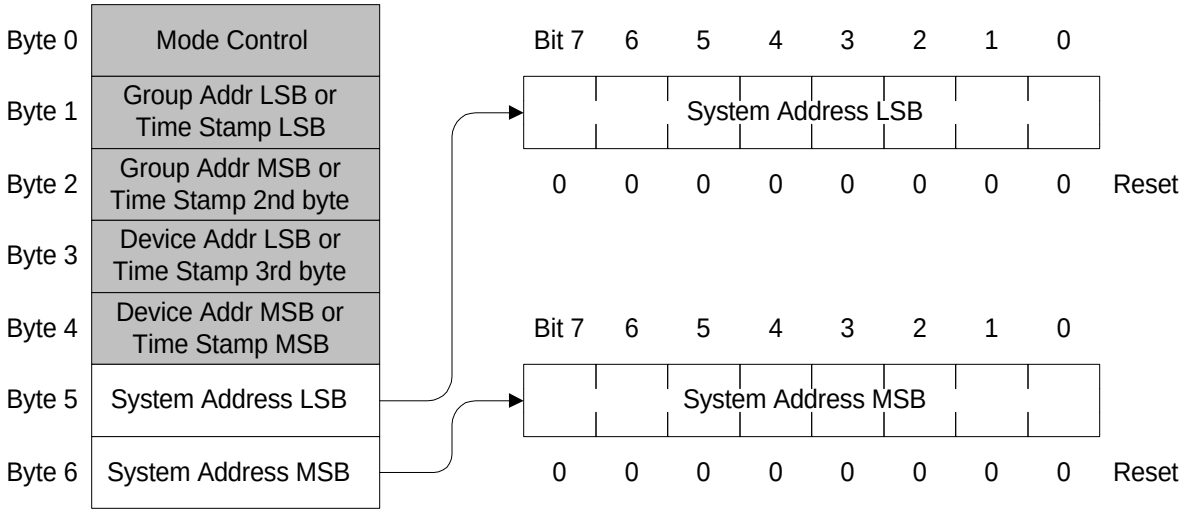
Re-transmission (immediate and multiple ADR-type accesses) are handled by the *CON* mode based on the transmit packet source address fields. NOTE: The use of the WTI command is not recommended in this mode because the sequence number is not updated properly.



Bytes 1 & 2: ***Group Address*** when in DLL mode, bytes 1 and 2 of the *Layer_Config_Info* data structure define the single group address to which the node may belong. When in MON mode, these bytes are the LSW of the 32-bit counter. This counter provides the time stamp associated with each packet returned in the monitor mode.



Bytes 3 & 4: ***Device Address*** when in DLL mode, bytes 3 and 4 of the *Layer_Config_Info* data structure define the unique device address. When in MON mode, these bytes are the MSW of the 32-bit counter. This counter provides the time stamp associated with each packet returned in the MON mode.



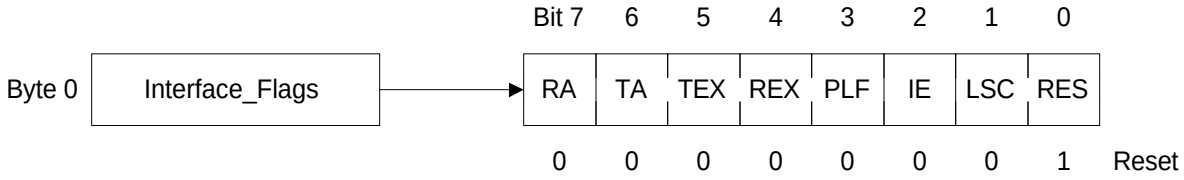
Bytes 5 & 6: ***System Address*** defines the system address used by the node.

Interface_Flags Data Structure

The read-only *Interface_Flags* register contains information relative to the state of the P300 including:

- The fact that a packet has been received and whether a receive exception exists.
- Whether a transmission has completed and whether a transmit exception exists.
- The presence of a Physical Layer failure or host interface error.
- Whether certain status counters have overflowed.
- The reset state of the device.

NOTE: All flags are cleared upon executing the *Interface_Read* command; therefore, it is the responsibility of the host to service all flags in *Interface_Flags* when read.



Byte 0: ***Interface Flags*** provides information on the current state of the P300.

RA:*Rc_Attn* indicates that a valid packet has been received.

0 No packet received.

1 Packet received (Attention sequence generated).

TA:*Tr_Attn* indicates that a pending packet transmission is complete.

0 No packet transmitted.

1 Packet transmission completed (Attention sequence generated).

TEX: *Tr_Except* In DLL and MON modes indicates a transmit condition other than a successful transmission of the last packet has occurred if the *Tr_Attn* flag is asserted along with *Tr_Except* flag. *Tx_Status_Code* in the *Status_Info* data structure may be read to determine the non-successful transmit condition. If the *Tr_Attn* flag is not asserted, then *Tr_Except* flag indicates that a packet transmission is pending or in progress.

0 No exception exists or no transmission pending or in progress.

1 Exception occurred or transmission is pending or in progress.

In *CON* mode, ONLY *Tr_Except* indicates that the Preamble EOF of a transmission attempt has been successfully transmitted to the medium.

0 Preamble EOF not yet transmitted.

1 Preamble EOF transmitted (Attention sequence generated).

REX: *Rc_Except* in DLL mode ONLY indicates that a receive condition other than the successful reception of a valid packet has occurred if *Rc_Attn* flag is asserted along with *Rc_Except* flag. *Rcv_Exceptions* may be read from the *DLL_Rc_Link_Status* data structure to determine the receive exception.

0 No receive exception occurred.

1 Receive exception occurred.

In CON mode ONLY *Rc_Except* flag indicates that a complete header has been received.

0 No header received.

1 Header received (Attention sequence generated).

PLF: *Physical_Layer_Failure* indicates an error was detected in the physical layer. *Phy_Layer_Fail_Status* may be read from the *DLL_Rc_Link_Status* data structure to determine cause of the failure.

0 No physical layer exists.

1 Physical failure occurred (Attention sequence generated).

IE: *Interface_Error* indicates an SPI protocol error occurred on the last attempted host command sequence.

0 No SPI protocol error occurred.

1 SPI protocol error detected.

LSC: *Link_Status_Condition* indicates overflow condition in the statistics counter(s) in the *Status_Info* data structure.

0 No overflow occurred.

1 Overflow occurred (Attention sequence generated by this flag if the *Link_Status_Enable* flag is set in the *DLL_Rc_Link_Status* data structure).

RES: *IC_Reset* indicates the P300 is in a reset condition. A *Layer_Mgmt_Write* command is required to initialize and activate.

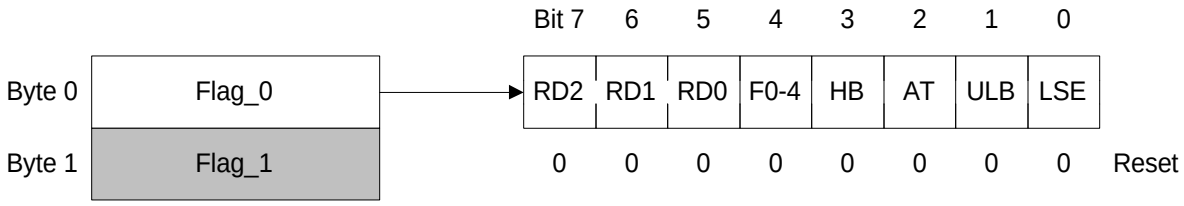
0 P300 not reset (running).

1 P300 reset (waiting for *Layer_Mgmt_Write* command).

Node_Control Data Structure

The write-only *Node_Control* structure contains flags which control node operation including:

- Enabling an attention sequence upon statistic counter overflow.
- Specifying general and specific host busy situations.
- Aborting the transmission of a packet.
- Controlling the receipt of packets in the *CON* mode.
- Controlling the receipt of extended service packets.
- Determining whether the sequence number on received ADRIACK packets should be compared or ignored.



Byte 0: ***Flag_0*** allows the host to handle the disposition of received packets in the CON mode, notify the P300 that the host is busy, etc. Information on each bit field is given below.

RD2 - RD0: ***Rcv_Disposition*** in *CON* mode ONLY determines the response to a received header and therefore the subsequent disposition of the received packet.

RD2	RD1	RD0	*Rcv_Disposition*
0	0	0	Ignore the packet (no IACKs generated).
0	0	1	Post *Rc_Attn* flag in *Interface_Flags* data structure at packet end and send IACK (if appropriate).
0	1	0	Do not post *Rc_Attn* flag in *Interface_Flags* data structure at packet end and send "default remote busy" IACK (if appropriate).
0	1	1	Do not post *Rc_Attn* flag in *Interface_Flags* data structure at packet end and send "default remote reject" IACK (if appropriate).
1	0	0	Do not post *Rc_Attn* flag in *Interface_Flags* data structure at packet end and send "remote reject extended" IACK (if appropriate).
1	0	1	Do not post *Rc_Attn* flag in *Interface_Flags* data structure at packet end and send "remote busy receive list" IACK (if appropriate).
1	1	0	Reserved
1	1	1	Reserved

F0-4: ***Flag_0 bit 4*** reserved for future use and must be set to 0.

HB: ***Host_Busy*** indicates host cannot process any attention requests at this time.

0 All attention requests generated

1	No attention requests generated

AT: *Abort_Transmission* aborts current packet transmission (if not already completed). Abort occurs prior to channel access or after transmission completion (i.e. fragments will not be produced).

0 No abort

1 Abort the current transmission

ULB: *Upper_Layer_Busy* indicates host cannot process received packets at this time. No attention requests for received packets are generated (the first received packet is stored—subsequent received packets are refused/ignored). Other attention requests (including transmit complete) are generated normally. If an attention request is generated due to a non-"received packet" reason, but a packet has been received, it is the responsibility of the host to either process the received packet immediately or store the fact and process the received packet later. See description of *Interface_Flags* pertaining to processing all set flags.

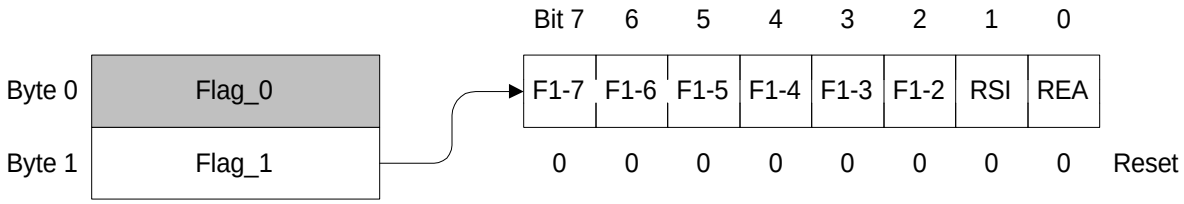
0 All attention requests generated

1 No attention requests generated for received packets

LSE: *Link_Status_Enable* enables the generation of an attention sequence when any statistics counter overflows.

0 No attention sequence generated

1 Attention sequence generated on statistic counter overflow



Byte 1: *Flag_1* information on each bit field is given below.

F1-7 - F1-2: *Flag_1 bits 7 to 2* reserved for future use and must be set to 0

RSI: *Rcv_SeqNo_Ignore* in *DLL* and *CON* modes determines whether the sequence number in the received ADRIACK is compared with the transmitted packet's sequence number or ignored.

0 Compare the received ADRIACK's and transmitted packet's sequence numbers.

1 Ignore the received ADRIACK's sequence number.

REA: *Rcv_ExtSvc_Accept* in *DLL* mode ONLY determines whether packets with EXTENDED service class are accepted or ignored.

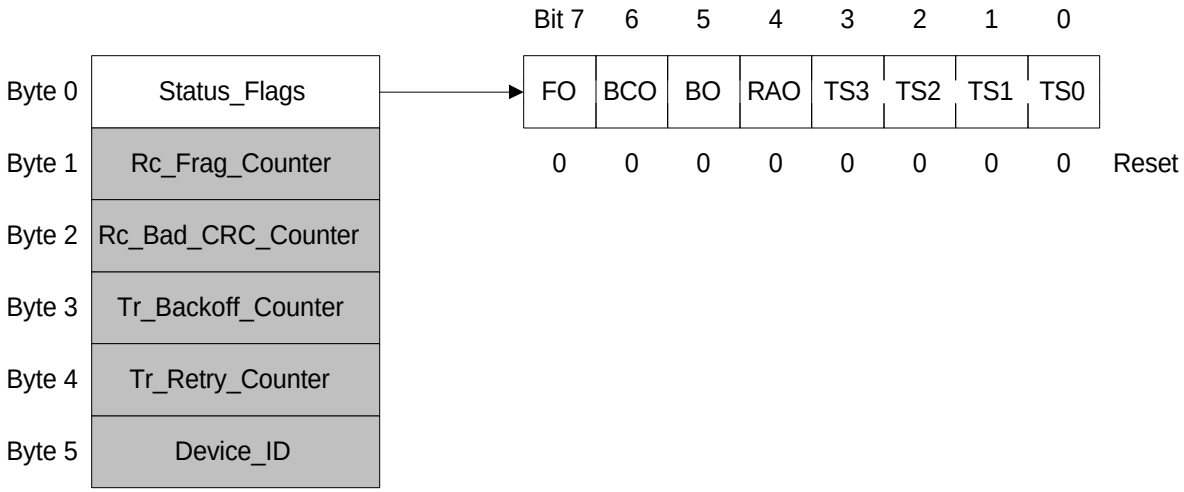
0 Ignore extended service packets.

1 Accept extended service packets.

Status_Information Data Structure

The read-only *Status_Info* structure contains flags including:

- The status of the last packet's transmission.
- Statistic counter values and overflow states.
- The P300 device type and version.



Byte 0: ***Status_Flags*** givens information on the status of statistical counters and the transmission status. A description of each bit field is given below.

FO:*Frag_Ovfl* indicates *Rc_Frag_Counter* overflow.

0 No overflow

1 Overflow

BCO: *Bad_CRC_Ovfl* indicates *Rc_Bad_CRC_Counter* overflow.

0 No overflow

1 Overflow

BO: *Backoff_Ovfl* indicates *Tr_Backoff_Counter* overflow.

0 No overflow

1 Overflow

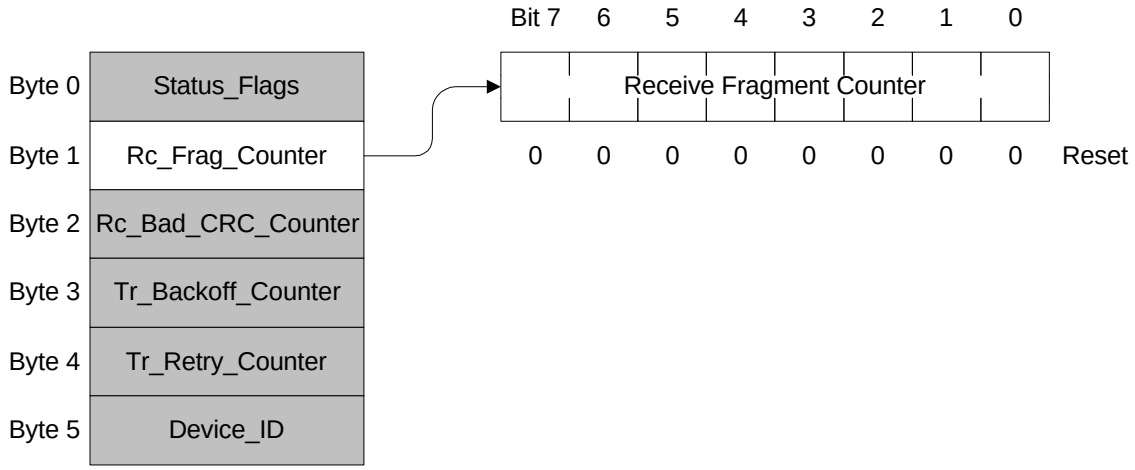
RAO: *Retry_Abort_Ovfl* indicates *Tr_Retry_Counter* overflow.

0 No overflow

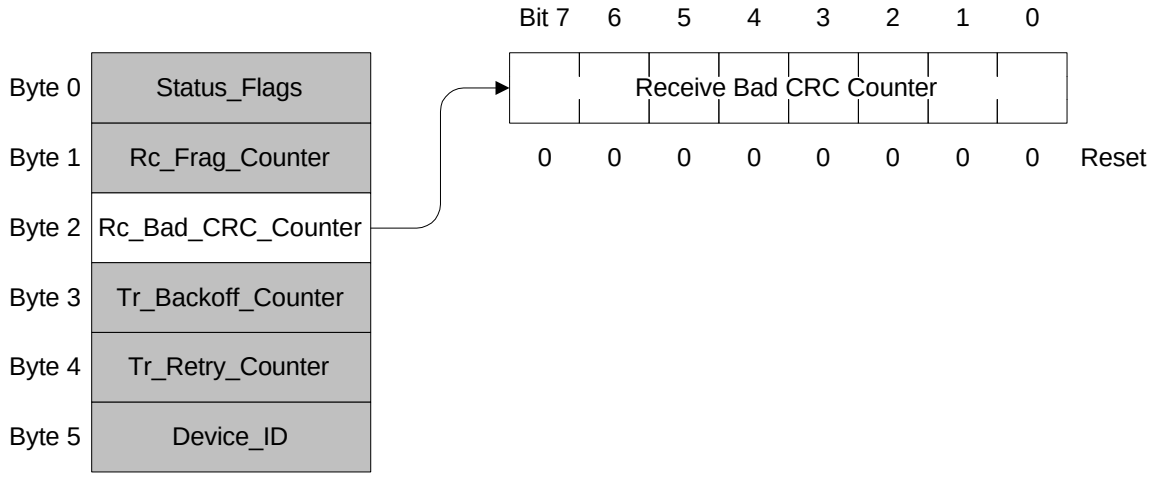
1 Overflow

TS3 - TS0: ***Tx_Status_Code*** indicates the last packet's transmission completion status.

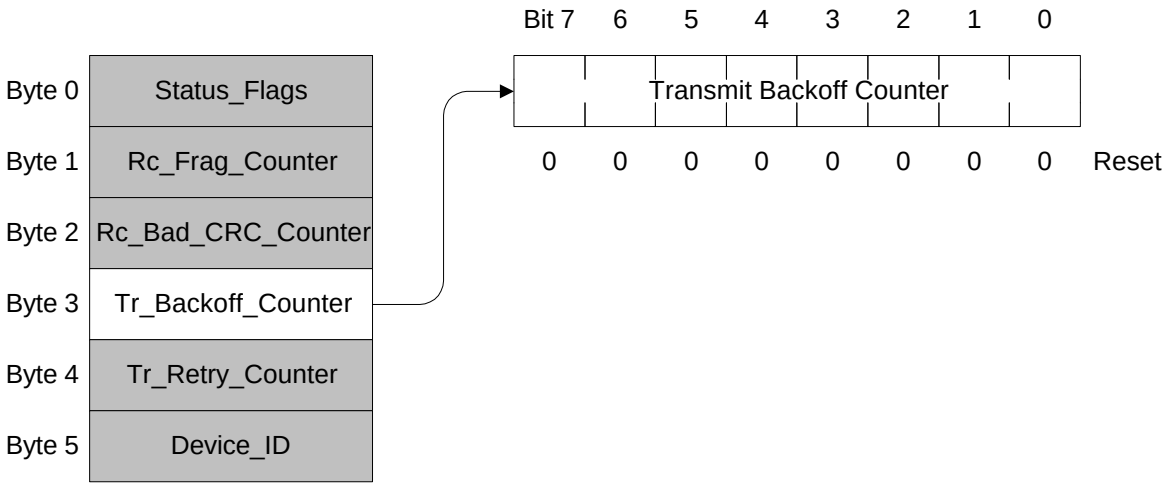
TS3	TS2	TS1	TS0	*Tx_Status_Code*
0	0	0	0	Success: packet transmission succeeded
0	0	0	1	Remote_Busy: remote device transmitted ADRIACK/FAIL "remote busy".
0	0	1	0	Remote_Reject/Other: remote device transmitted ADRIACK/FAIL "remote reject" or "other".
0	0	1	1	Excessive_Retries: transmitted ADRACK packet and did not receive ADRIACK within Max_Retrans time.
0	1	0	0	Excessive_Collisions: exceeded *Max_Restarts* attempts.
0	1	0	1	No_IACK: did not receive IACK in required time.
0	1	1	0	Bad_IACK: received fragment or packet with bad CRC received instead of IACK.
0	1	1	1	Wrong_IACK: received non-(ADR)IACK/FAIL packet or ADRIACK with wrong sequence number.
1	0	0	0	Misc_Xmit: detected unexpected packet termination.
1	0	0	1	Physical_Failure: transceiver or medium failure detected.
1	0	1	0	Transmit_Aborted: host aborted transmission via *Abort_Transmission* flag found in *Flag_0* of the *Node_Control* data structure.
1	0	1	1	Max_Retrans_Exceeded: could not complete transmission within Max_Retrans time.
1	1	0	0	Local_Busy_Xmit_List: packet transmission cannot be accepted due to un-expired transmit association timer.
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved



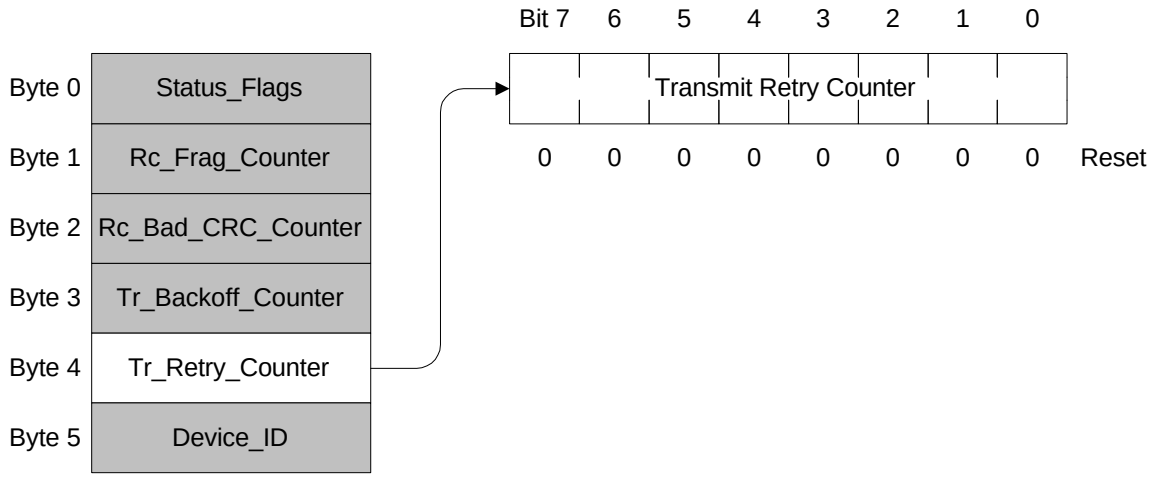
Byte 1: ***Rc_Frag_Counter*** an 8-bit free-running counter starting at zero after reset. When this counter overflows, it is posted in the Status_Flags bit FO; see description above. Counts the number of packet fragments detected. A fragment contains an initial packet Preamble EOF and no terminating EOP.



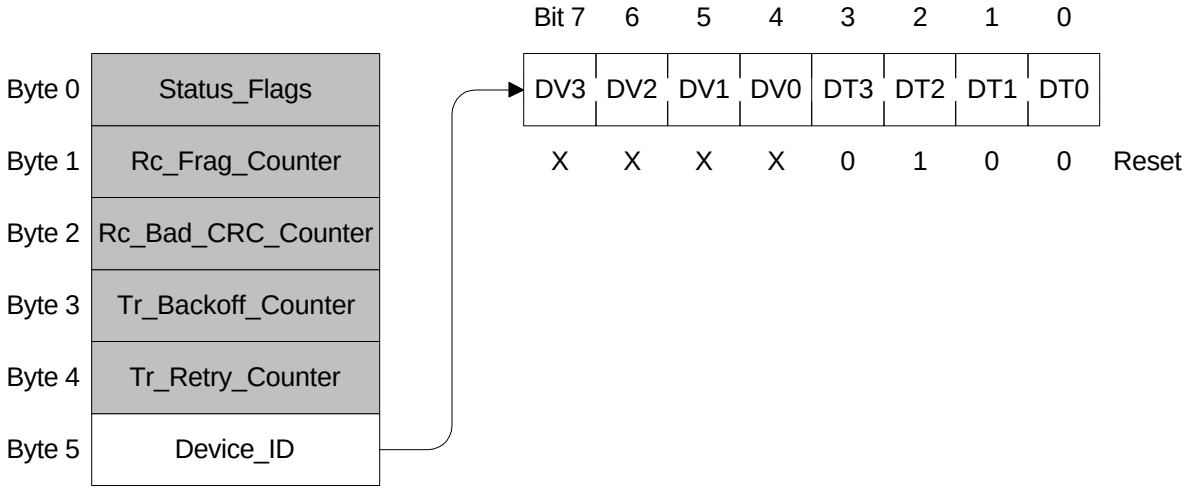
Byte 2: ***Rc_Bad_CRC_Counter*** an 8-bit free-running counter starting at zero after reset. When this counter overflows, it is posted in the Status_Flags bit BCR; see description for byte 0 of the *Status_Info* data structure. Counts the number of packets received with CRC errors.



Byte 3: ***Tr_Backoff_Counter*** an 8-bit free-running counter starting at zero after reset. When this counter overflows, it is posted in the Status_Flags bit BO; see description for byte 0 of the *Status_Info* data structure. Counts the number of transmission attempts that backed off due to contentions, collisions, or abort deferrals.



Byte 4: ***Tr_Retry_Counter*** an 8-bit free-running counter starting at zero after reset. When this counter overflows, it is posted in the Status_Flags bit RAO; see description for byte 0 of the *Status_Info* data structure. Counts the number of retry ADRACK and ACK packets transmitted beyond the initial attempts.



Byte 5: ***Device_ID*** allows the host to determine the type and version of device. The values for the bit fields are given on the next page.

DT3 - DT0: ***Device_Type*** indicates the Intellon IC device type.

DT3	DT2	DT1	DT0	*Device_Type*
0	0	0	0	CEThinx
0	0	0	1	P400
0	0	1	0	R400
0	0	1	1	P200
0	1	0	0	P300
0	1	0	1	Reserved
0	1	1	0	Reserved
0	1	1	1	Reserved
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

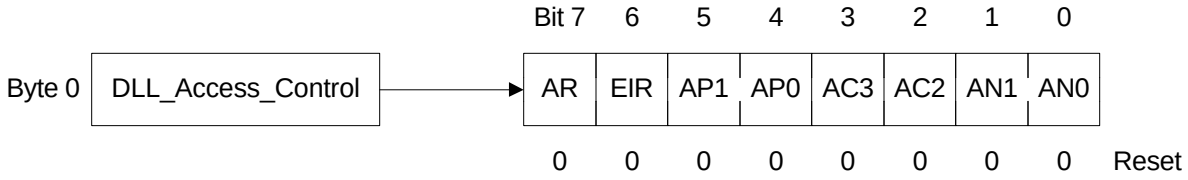
DV3 - DV0: ***Device_Version*** indicates the device's firmware version.

DV3	DV2	DV1	DV0	*Device_Version*
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

DLL_Access_Control Data Structure

The write-only *DLL_Access_Control* register specifies:

- The number of ADRUNACK packets to be transmitted per host transmit session.
- The time between channel accesses.
- Whether an immediate retry is transmitted for ACK and ADRACK transmissions .
- Whether multiple channel accesses are attempted for ADRACK and ADRUNACK packets.



Byte 0: ***DLL_Access_Control*** allows the host to determine channel access behavior for the different Data Link services. Information on each bit field is given below.

AR:*Auto_Retry* enables multiple channel access retries for ADRACK and ADRUNACK packets. Multiple channel access retries are different from immediate retries, in that, multiple channel access retries must contend for the channel before transmitting.

0 No multiple channel access retries attempted.

1 Multiple channel access retries.

EIR: ***Enable_Immediate_Retry*** enables immediate retry for ACK and ADRACK packets. An immediate retry is a retransmission during the 6^{th} unit symbol time (UST) after the first transmission if no IACK is received. Because the original packet "owns" the channel for 10 USTs, the immediate retry does not need to contend for the channel and is therefore transmitted without a preamble.

0 No immediate retry transmitted.

1 Immediate retry transmitted if no response in required time.

AP1 - AP0: Ch*_Access_Period* specifies the minimum required time in milliseconds between attempted transmit channel accesses and is used to limit the duty cycle of packet transmissions based on the device's power supply design.

CAP1	**CAP0**	***Ch_Access_Period***
0	0	0 ms
0	1	100 ms
1	0	200 ms
1	1	300 ms

AC3 - AC2: ***Access Control bits 3 to 2*** reserved for future use and must be set to 0.

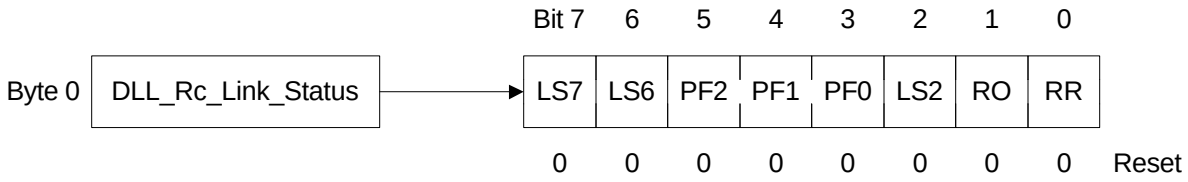
AN1 - AN0: ***Ch_Access_Num*** specifies the number of ADRUNACK channel accesses for each host ADRUNACK transmit requested.

AN1	**AN0**	***Ch_Access_Num***
0	0	1
0	1	2
1	0	4
1	1	8

DLL_Rc_Link_Status Data Structure

The read-only *DLL_Rc_Link_Status* register indicates:

- Whether the received packet has a different source or destination address compared to the most recently received packet.
- Whether an overrun condition occurred (an UNACK or ADRUNACK packet was received before a prior received packet was retrieved by the host).
- Whether a refused condition occurred (an ACK or ADRACK packet was received before a prior received packet was retrieved by the host).
- The type of Physical Layer failure detected.



Byte 0: ***DLL_Rc_Link_Status*** allows the host to determine status of received packets. Information on each bit field is given below.

LS7 - LS6: ***Link Status bits 7 to 6*** reserved for future use and will be set to 0.

PF2 - PF0: ***Phy_Layer_Fail_Status*** if *Physical_Layer_Failure* in *Interface_Flags* data structure is TRUE, this field indicates the cause of Physical Layer failure.

PF2	PF1	PF0	*Phy_Layer_Fail_Status*
0	0	0	Unexpected end-of-packet was detected
0	0	1	Transceiver interface error
0	1	0	Reserved
0	1	1	Reserved
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

LS2: ***Link Status bit 2*** reserved for future use and will be set to 0.

RO: ***Rc_Overrun*** in *DLL* mode ONLY and if *Rc_Except* in *Interface_Flags* data structure is TRUE, indicates that a valid correctly addressed UNACK or ADRUNACK packet was ignored and discarded due to any of the following:

Host did not retrieve the previously received valid correctly addressed packet.

The received UNACK packet was received and the receive association timer had not yet expired.

The received ADRUNACK packet did not match the receive association addresses and the receive association timer had not yet expired.

0 No packet overrun condition exists.

1 Packet overrun occurred.

RR: *Rc_Refused* in *DLL* mode and if *Rc_Except* in *Interface_Flags* data structure is TRUE, indicates that a valid correctly addressed ACK or ADRACK packet was refused (FAIL/ADRIACK "remote busy" response sent) and discarded due to any of the following:

Host did not retrieve the previously received valid correctly addressed packet.

The received ACK packet was received and the receive association timer had not yet expired.

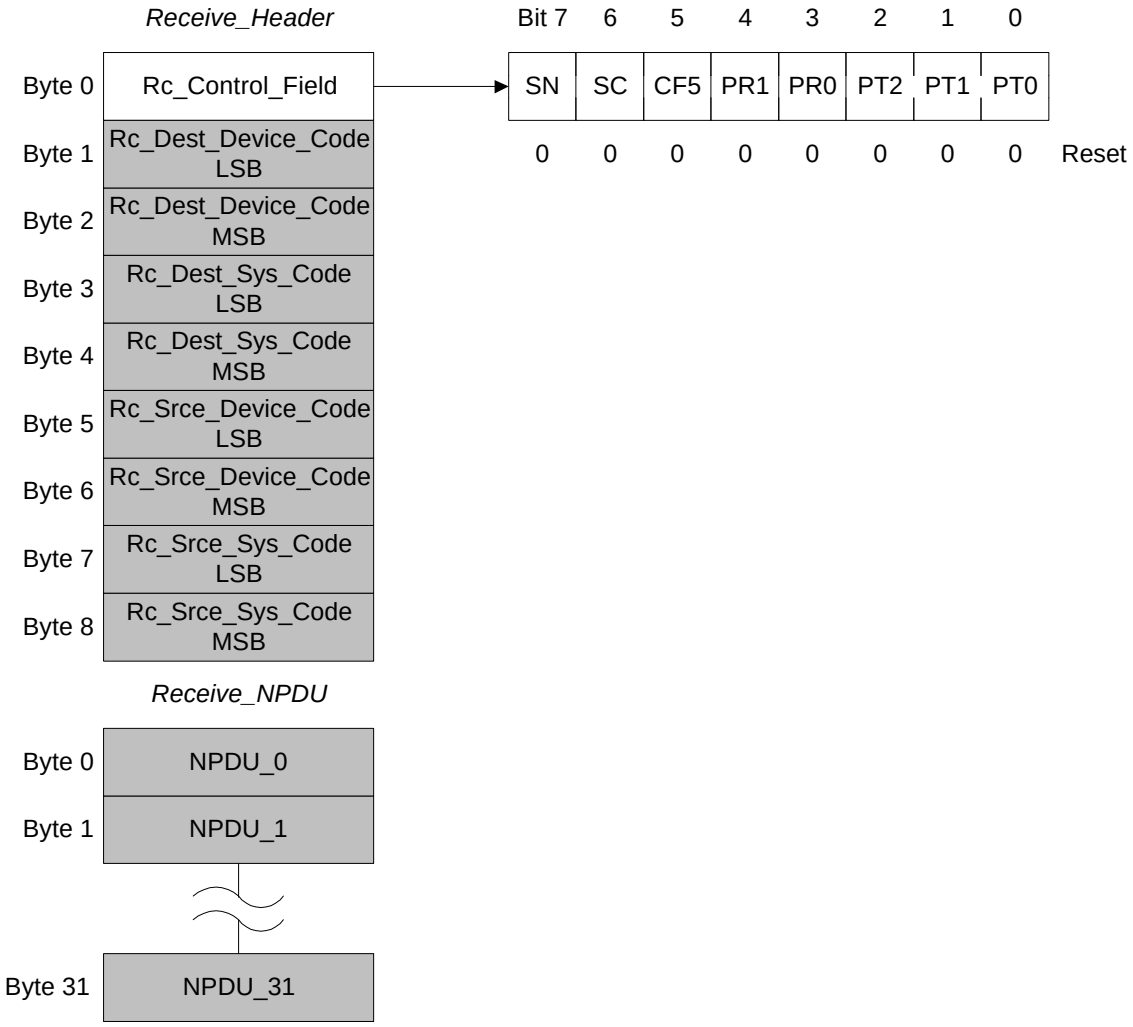
The received ADRACK packet did not match the receive association addresses and the receive association timer had not yet expired.

0 Packet not refused.

1 Packet refused.

Receive_Packet Data Structure

The *Receive_Packet* data structure is composed of the *Receive_Header* and *Receive_NPDU* data structures appended together. The *Receive_Header* has 9 bytes that contain information on such things as packet priority, Data Link services type, destination address, and source address. The *Receive_NPDU* is variable in length, with a maximum of 32 bytes, which contains the information used by the application.



Byte 0: ***Rc_Control_Field*** determines type of service and the priority of the received packet. Information on the bit fields is given below.

SN:*Sequence_Number* contains the sequence number of current packet (used with ADRACK, ADRUNACK and ADRIACK packets only).

SC:*Service_Class* indicates the service class bit value of the current packet.

0 Basic (not extended) service.

1 Extended service.

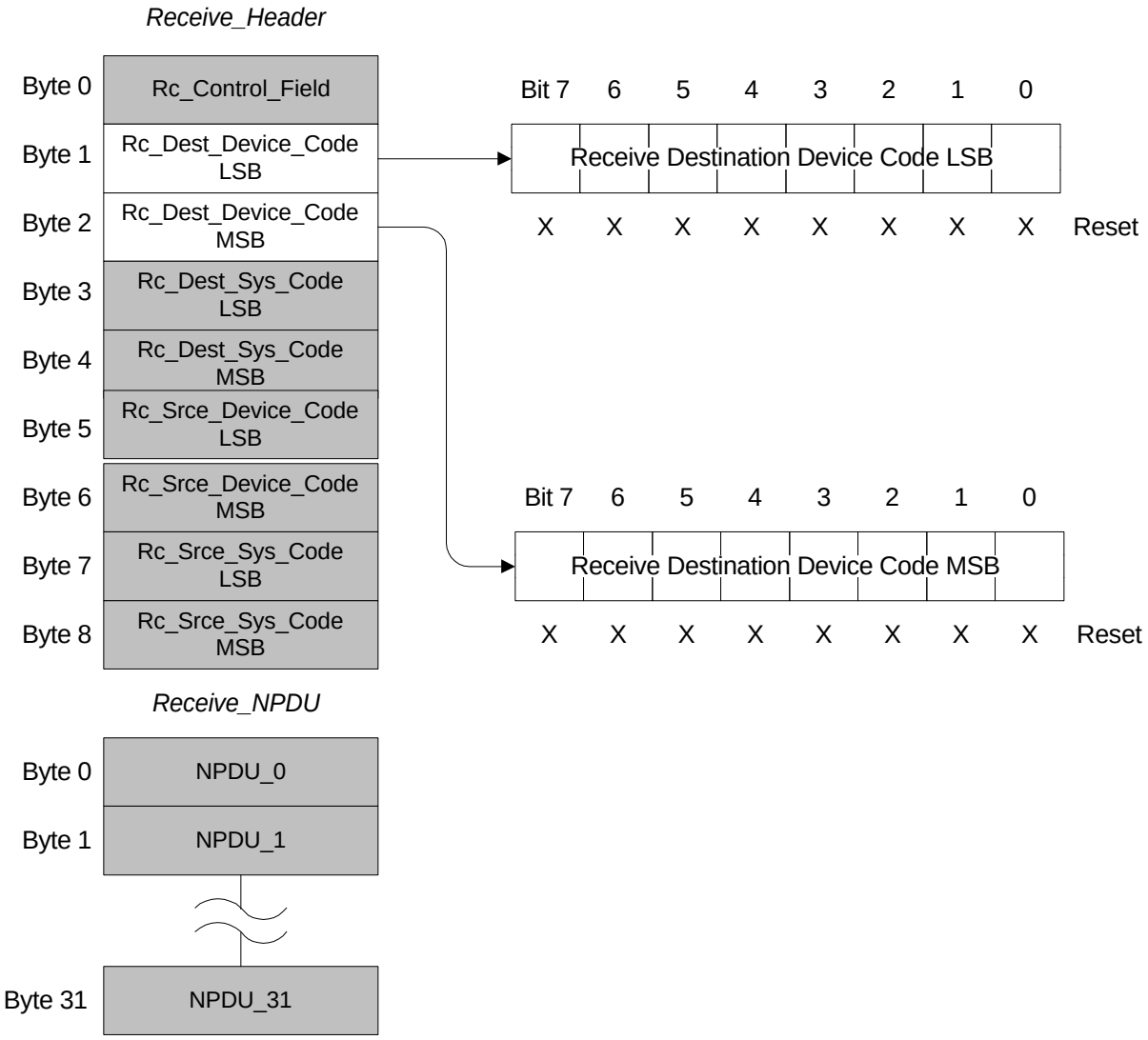
CF5: ***Control Field bit 5*** reserved.

PR1 - PR0: ***Packet_Priority*** indicates the transmit priority field value of the current packet.

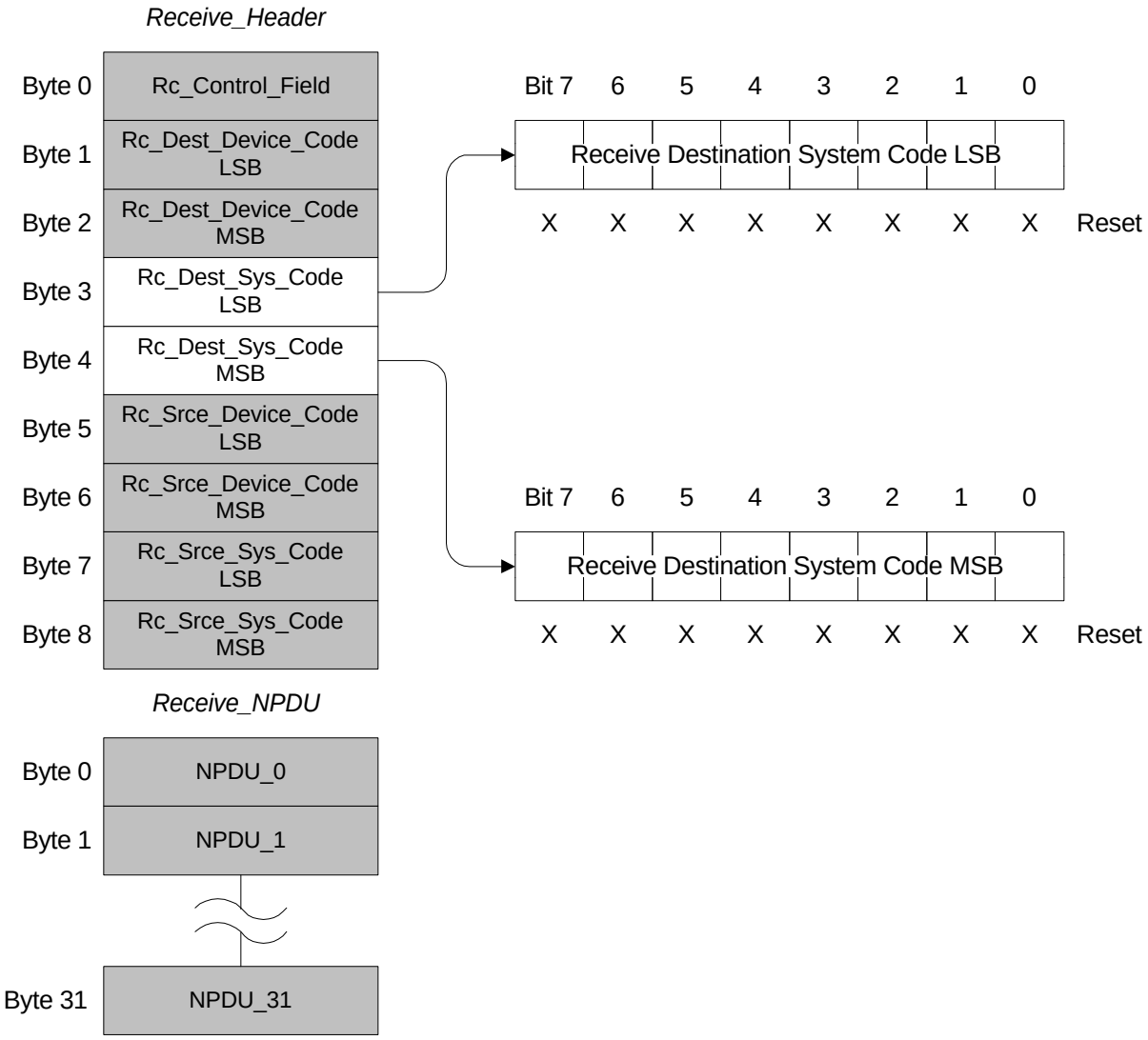
PR1	PR0	*Priority*
0	0	High
0	1	Standard
1	0	Deferred
1	1	Reserved

PT2 - PT0: ***Packet_Type*** indicates the packet type (DLL service) of the current packet.

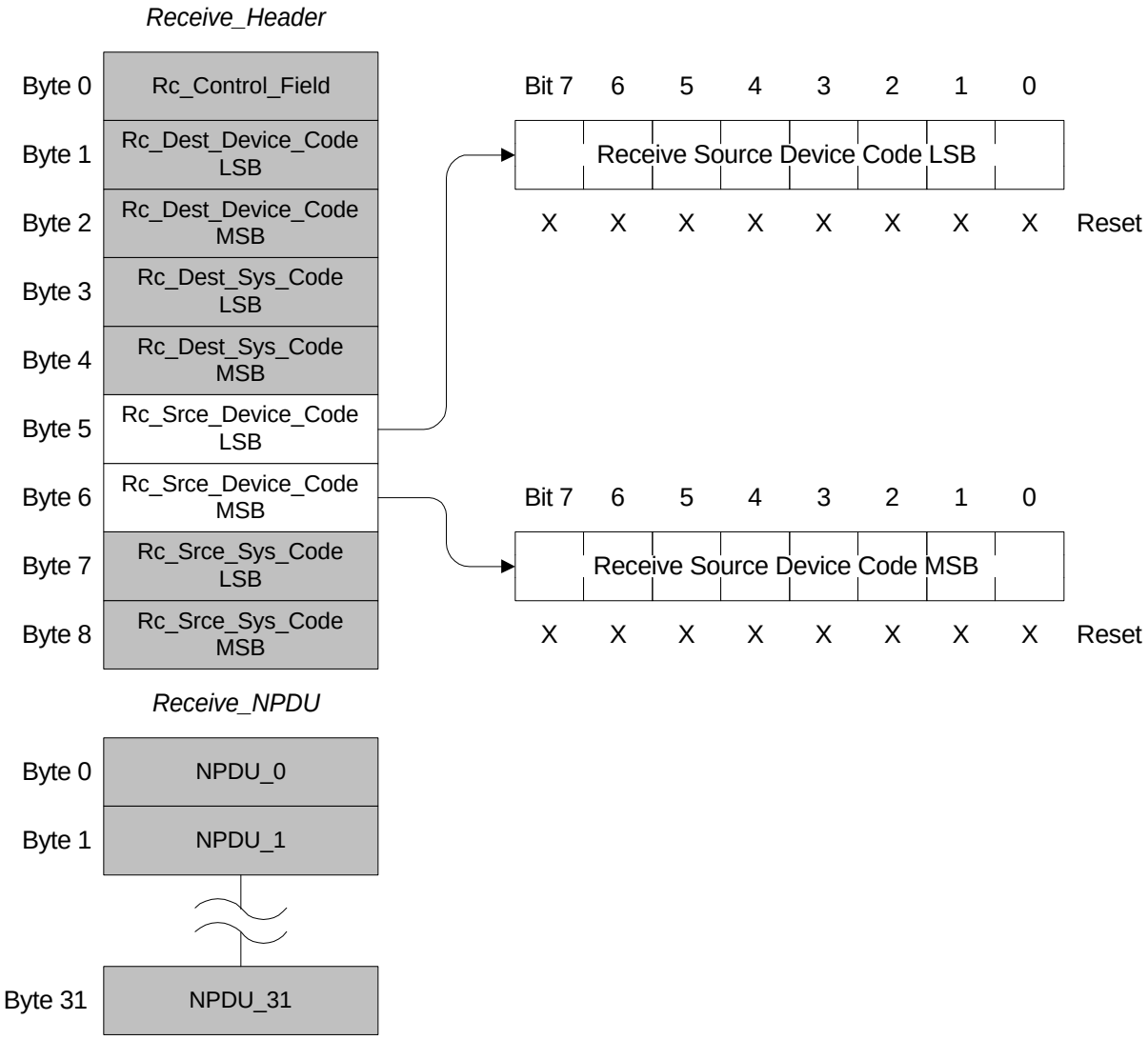
PT2	PT1	PT0	*Packet_Type*
0	0	0	IACK
0	0	1	ACK
0	1	0	UNACK
0	1	1	Reserved
1	0	0	FAILURE
1	0	1	ADRACK
1	1	0	ADRIACK
1	1	1	ADRUNACK



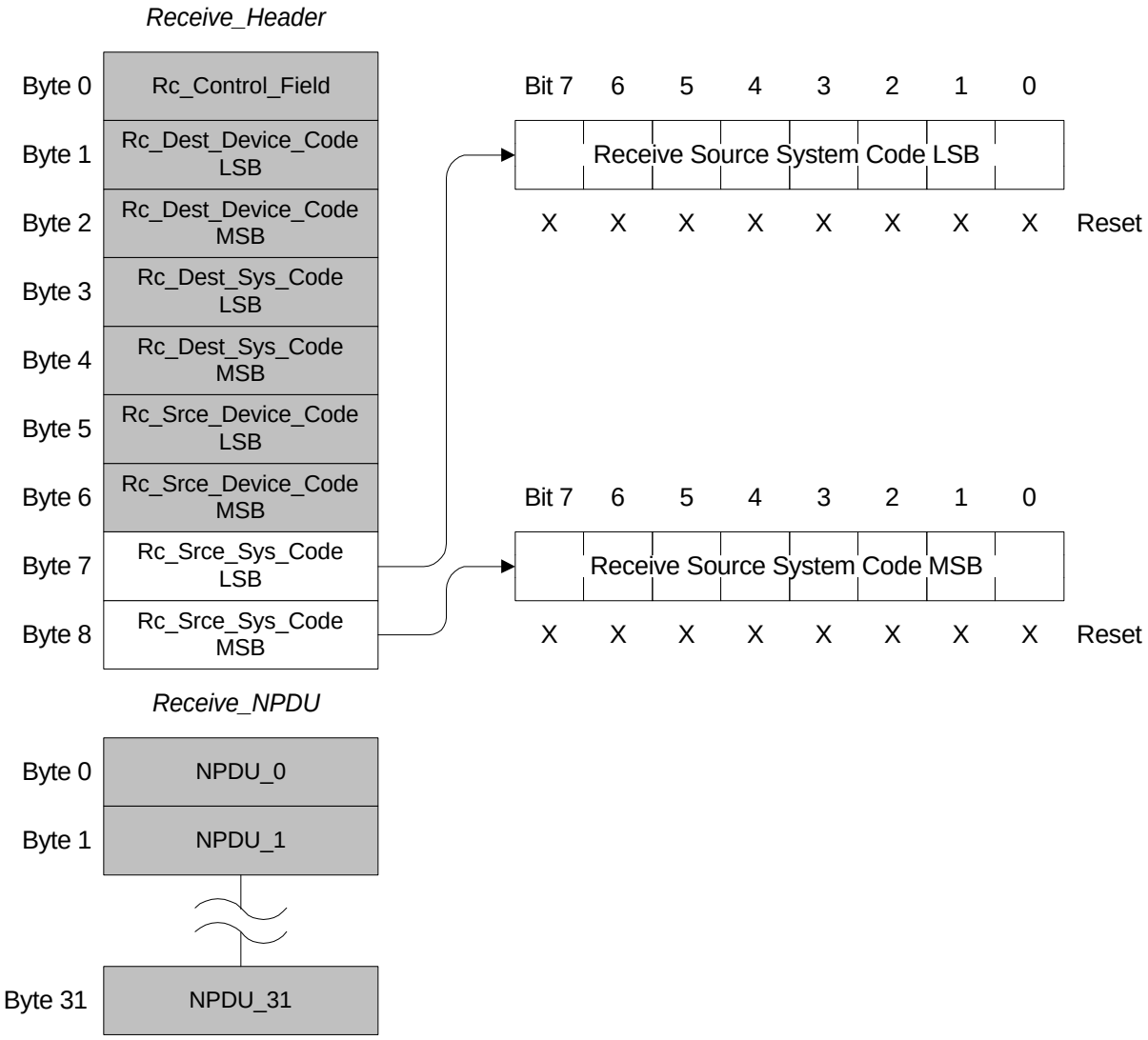
Bytes 1 & 2: ***Rc_Dest_Device_Code*** must match the group address, or device address, or the broadcast address (0x0000), in order for a packet to be received. The group and device addresses are defined in the *Layer_Config_Info* data structure. An alternate use for *Rc_Dest_Device_Code* is given after completion of ACK packet transmission, in which case, it contains the information field from received IACK/FAIL packet.



Bytes 3 & 4: ***Rc_Dest_Sys_Code*** must match the system address, or the universal broadcast address (0x0000), in order for the packet to be received. The system address is defined in the *Layer_Config_Info* data structure.

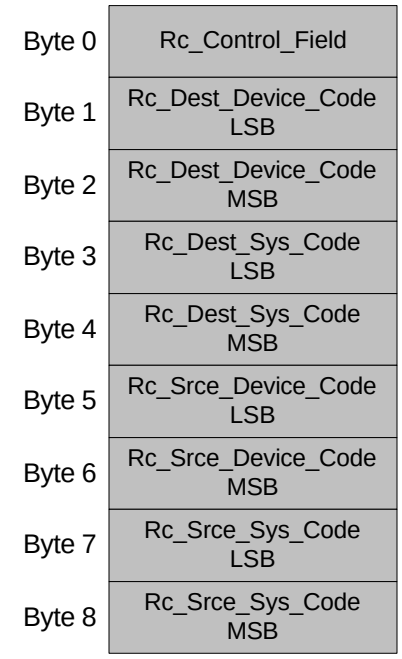


Bytes 5 & 6: ***Rc_Srce_Device_Code*** contain the device address of the originating node. An alternate use is given after completion of ADRACK packet transmission, in which case, *Rc_Srce_Device_Code* contains the information field from received ADRIACK packet.

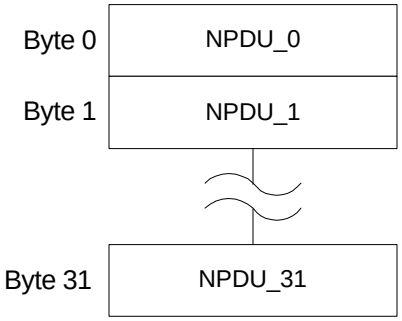


Bytes 7 & 8: ***Rc_Srce_Sys_Code*** contain the system address of the originating node.

Receive_Header



Receive_NPDU



Bytes 0 – 31: ***NPDU_0 - NPDU_31*** contains the message information that is processed by the host. In CEBus terminology, these bytes comprise the Network Protocol Data Unit (NPDU)

Transmit_Packet Data Structure

The *Transmit_Packet* data structure is composed of the *Transmit_Header* and *Transmit_NPDU* data structures appended together. The *Transmit_Header* has 9 bytes which determine such things as packet priority, Data Link services type, destination address, and source address. The *Transmit_NPDU* is variable in length, with a maximum of 32 bytes, which contains the message sent to the destination node. The *Transmit_Packet* data structure has the same format as the *Receive_Packet* data structure given in section 4.5. Any behavioral differences are noted below:

Bytes 0: ***Tr_Control_Field*** have the same meaning as the flags defined in *Rc_Control_Field* given in section 4.5.

Bytes 1 & 2: ***Tr_Dest_Device_Code*** contain the device address of the receiving node.

Bytes 3 & 4: ***Tr_Dest_Sys_Code*** contain the system address of the receiving node.

Bytes 5 & 6: ***Tr_Srce_Device_Code*** contain the device address of the transmitting node. If the host writes 0x0000, a null address will be transmitted. If in the DLL mode and the host writes 0xFFFF, then this field is replaced with the *Device_Addr* from the *Layer_Config_Info* data structure.

Bytes 7 & 8: ***Tr_Srce_Sys_Code*** contain the system address of the transmitting node. If the host writes 0x0000, a null address will be transmitted. If in the DLL mode and the host writes 0xFFFF, then this field is replaced with the System*_Addr* from the *Layer_Config_Info* data structure.

Bytes 0 – 31: ***NPDU_0 - NPDU_31*** contain the message sent to the receiving node.

Host Interactions

Host interactions are of the following four basic types:

- Initialization
- Write Transaction
- Read Transaction
- Attention Sequence

Read and write transactions were illustrated in sections 3.0. This section discusses the steps necessary for initialization and response to an attention sequence.

Initialization

Upon power-up, or following a reset command, the SSC P300 performs an internal diagnostic and setup sequence. Commands cannot be sent to the device until this sequence is complete. The following sequence of commands may be used to check for the ready condition:

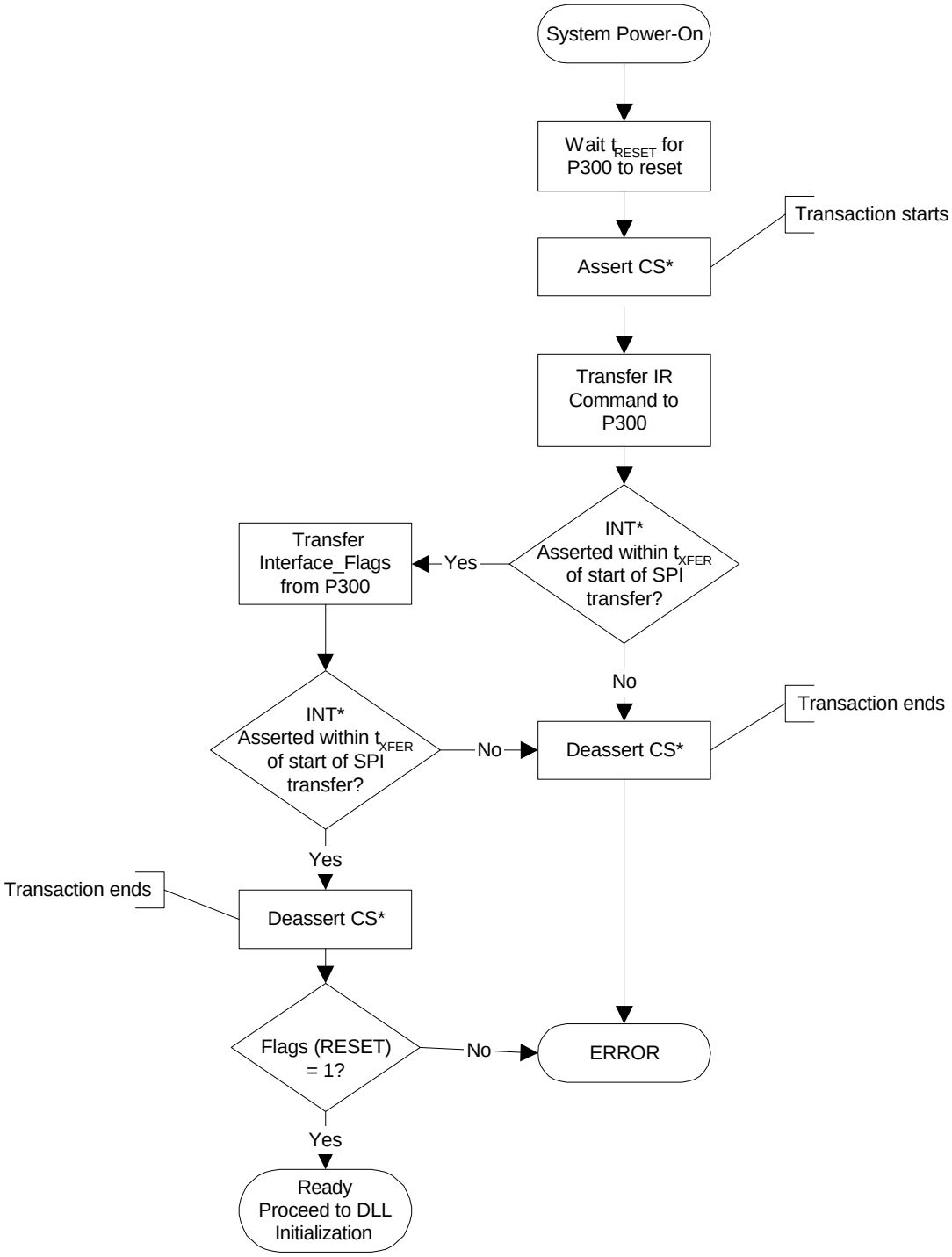


Figure 6. Initialization Flow Diagram

At this point, the reset is confirmed, the host issues a Layer_Management_Write command to set up the address fields and other parameters in the *Config_Info* structure. It is good practice to verify the Data Link system parameters. To do this, the host can issue a Layer_Mangement_Read command and test the returned data against the expected values. The Data Link information is volatile and must be written explicitly by the host, after loss of power or after a reset command. Figures 6 and 7 illustrate the steps to initialize the P300.

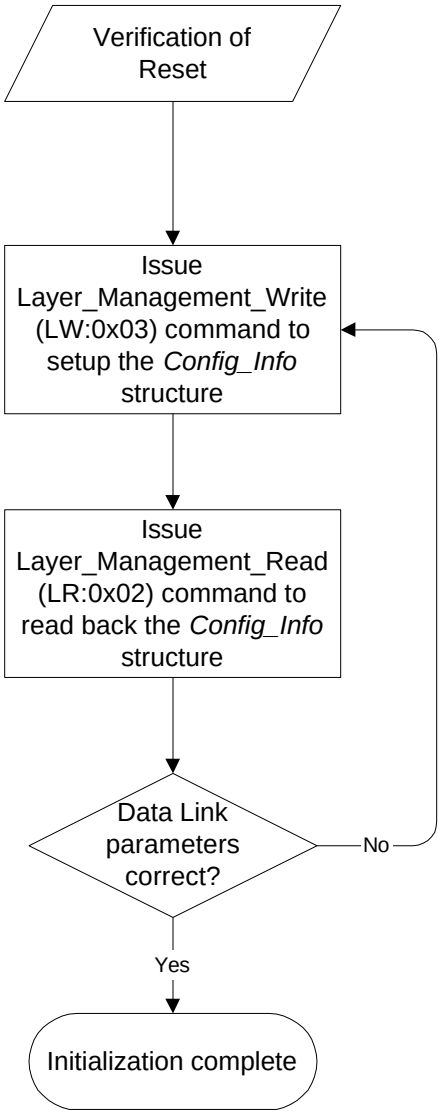


Figure 7. Initialization and Verification of *Layer_Config_Info* Data Structure

Attention Sequence

The P300 will assert the INT* signal to notify the host processor that data or status is available. The most common causes of the attention sequence are the reception of a CEBus packet or the completion of a transmitted CEBus packet. The attention sequence should be followed by a host Interface_Read command. The Interface_Read command will return the *Interface_Flags* data structure. These flags indicate the following conditions: a packet received with or without exception, a packet transmitted with or without exception, physical layer failure or host interface error, status counters overflow and the reset state of the device. It is important that all of the flags are processed, since reading the *Interface_Flags* clears all flags. The flow chart in Figure 8 illustrates the response to the attention sequence. The flow chart in Figure 9 shows the steps needed to process the *Interface_Flags*.

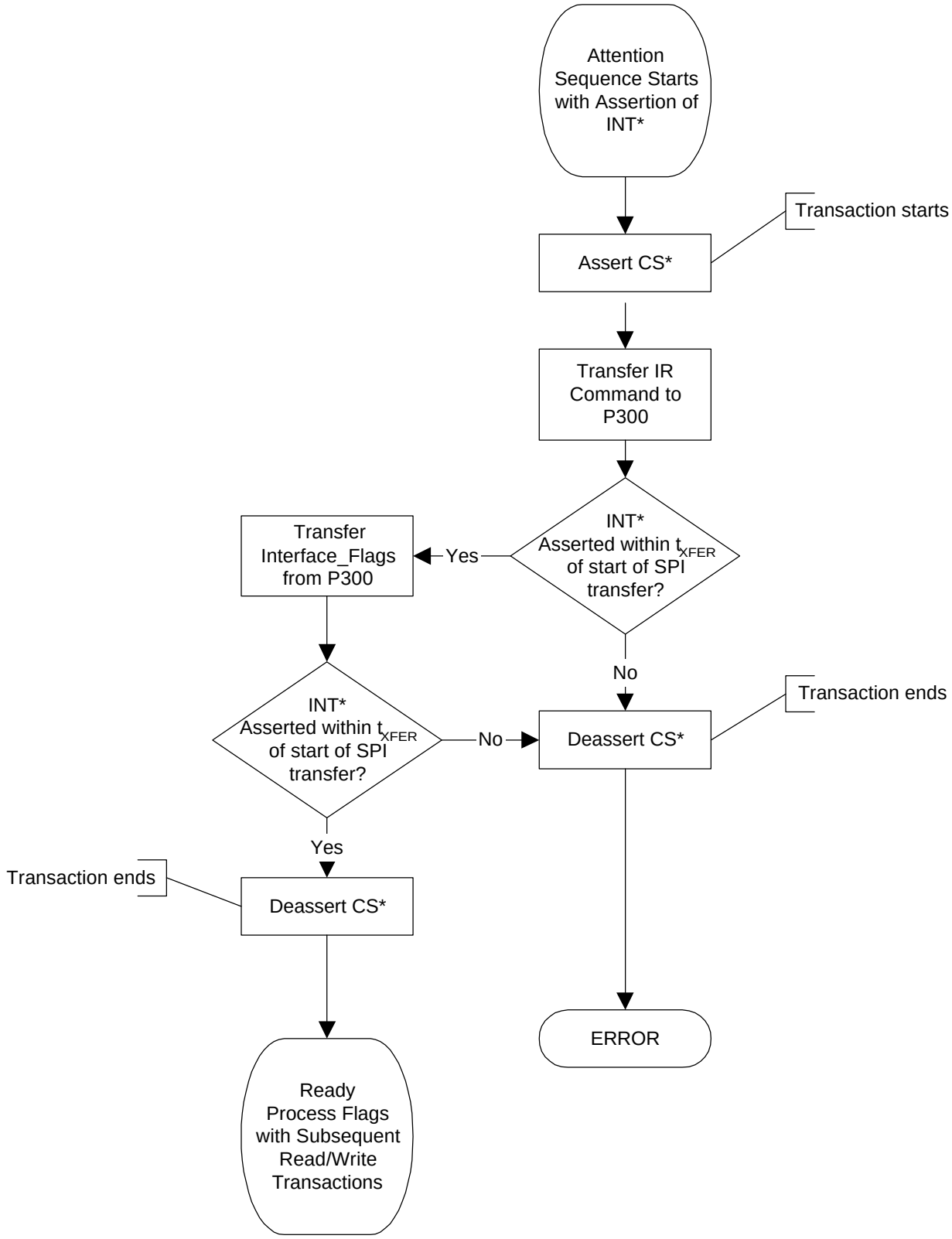


Figure 8. Attention Sequence Flow Diagram

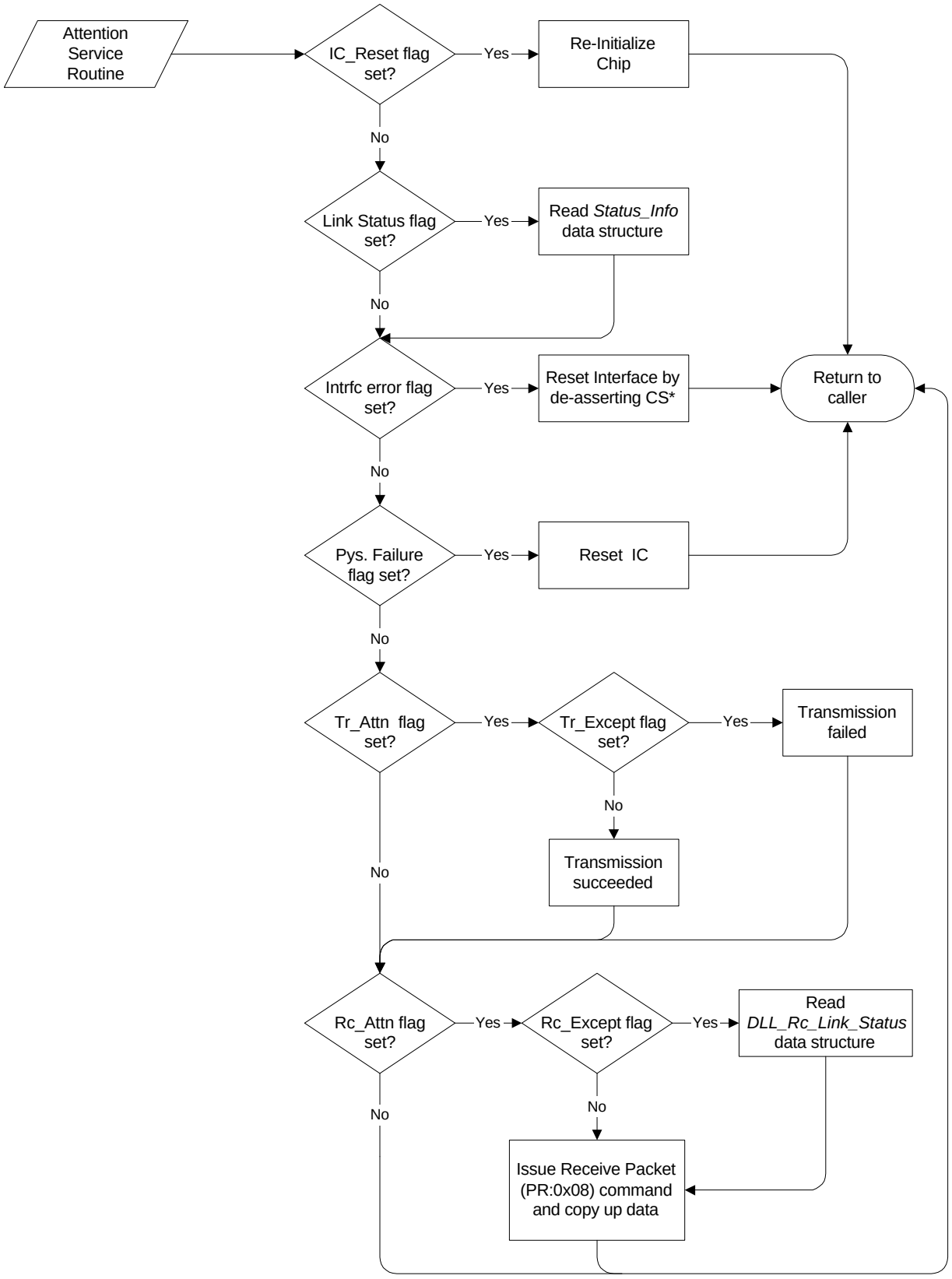


Figure 9. Processing the *Interface_Flags*

Transaction Timing

Values for timing parameters are given in Table 4 following the timing diagrams.

Write Transaction Figures 10, 11 and 12:

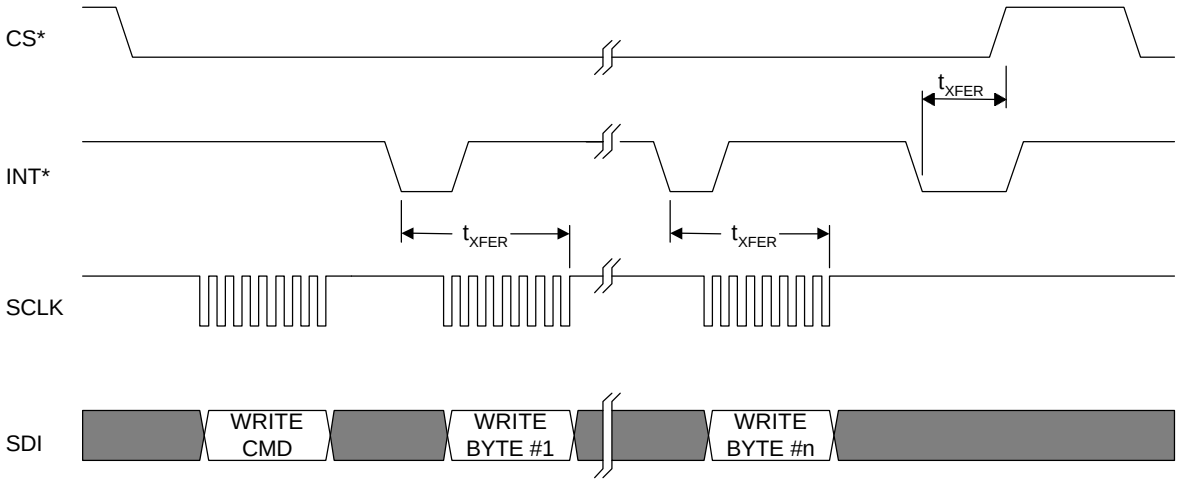


Figure 10. Write Transaction Timing Diagram

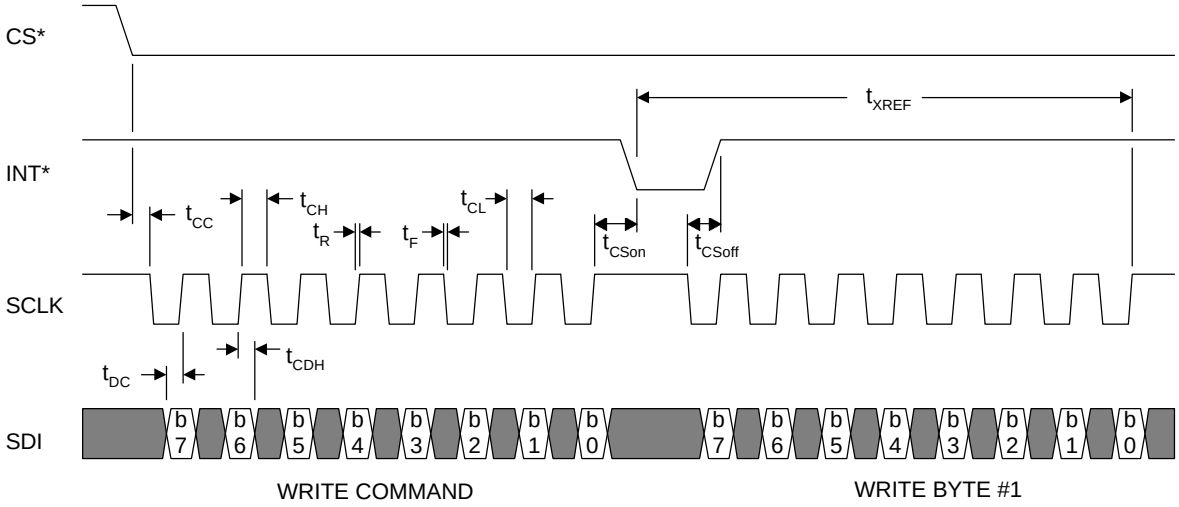


Figure 11. Write Byte #1 Timing Diagram

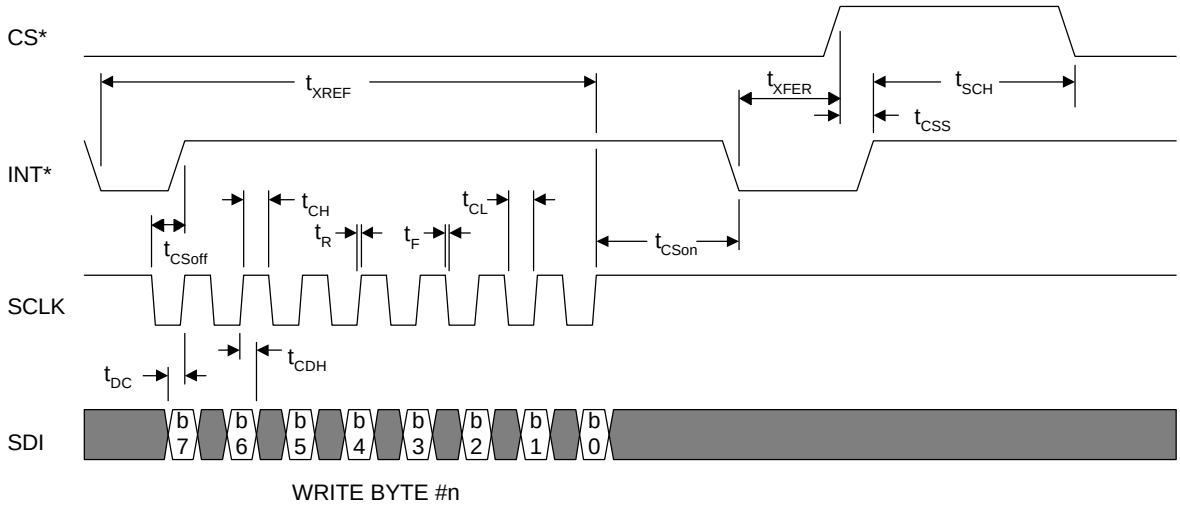


Figure 12. Write Byte #n Timing Diagram

Read Transaction Figures 13, 14 and 15:

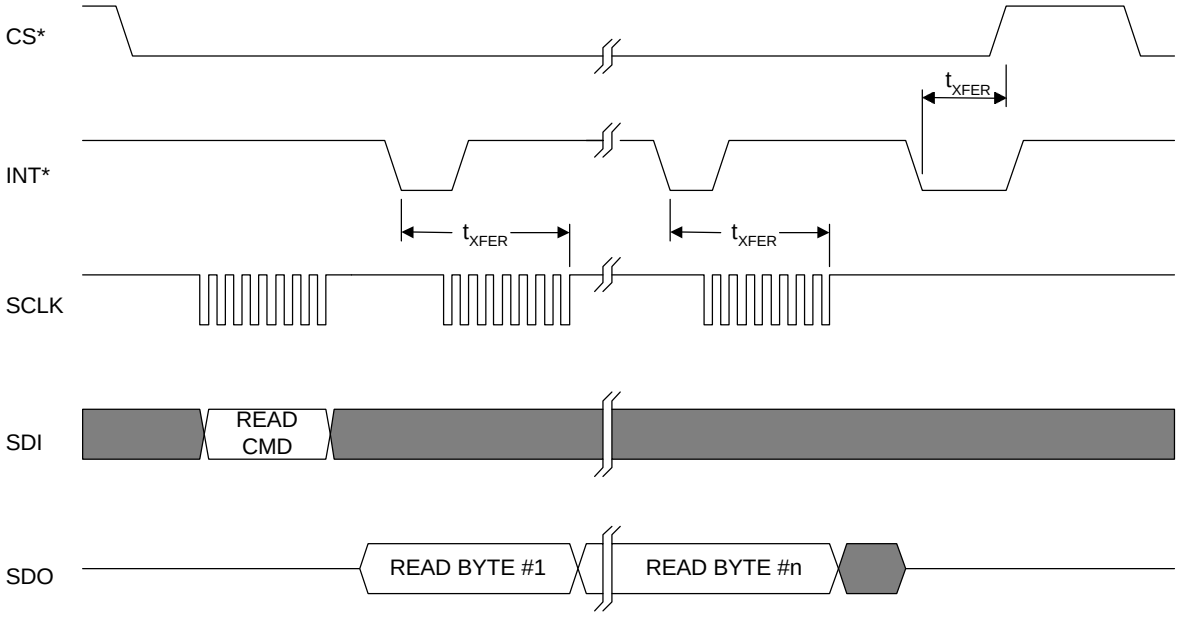


Figure 13. Read Transaction Timing Diagram

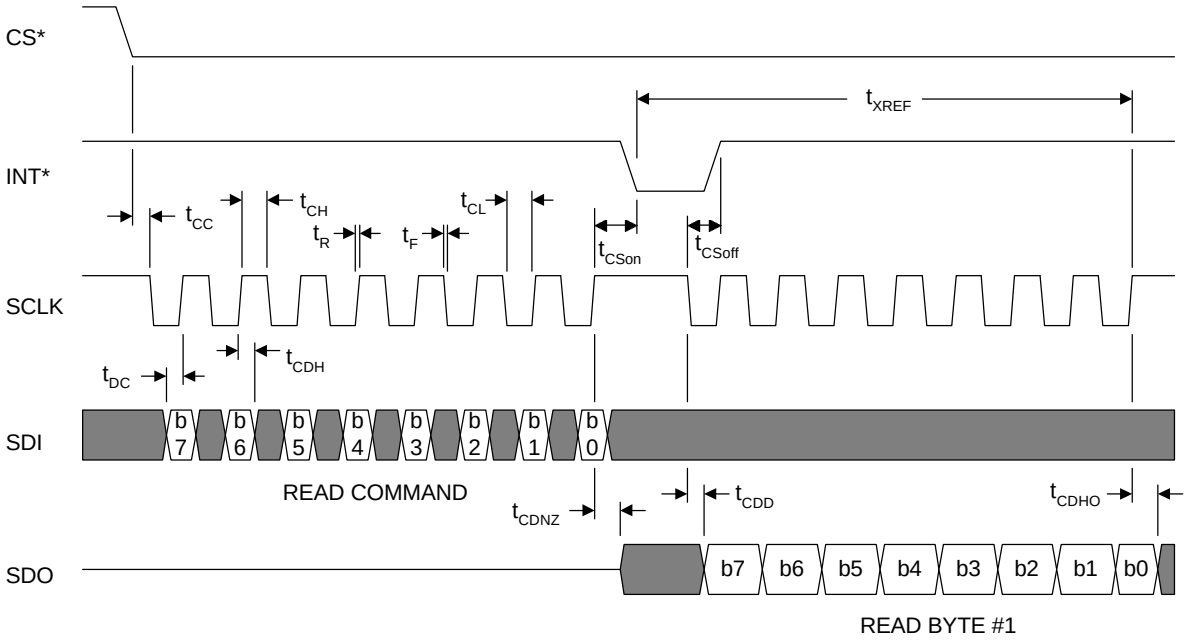


Figure 14. Read Byte #1 Timing Diagram

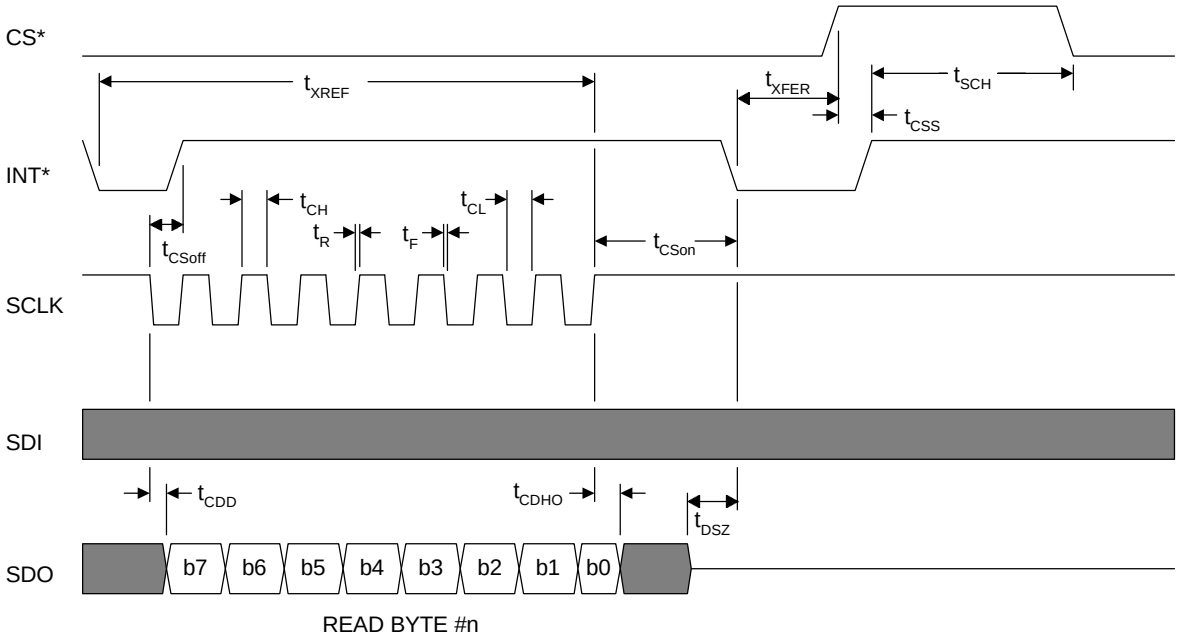


Figure 15. Read Byte #n Timing Diagram

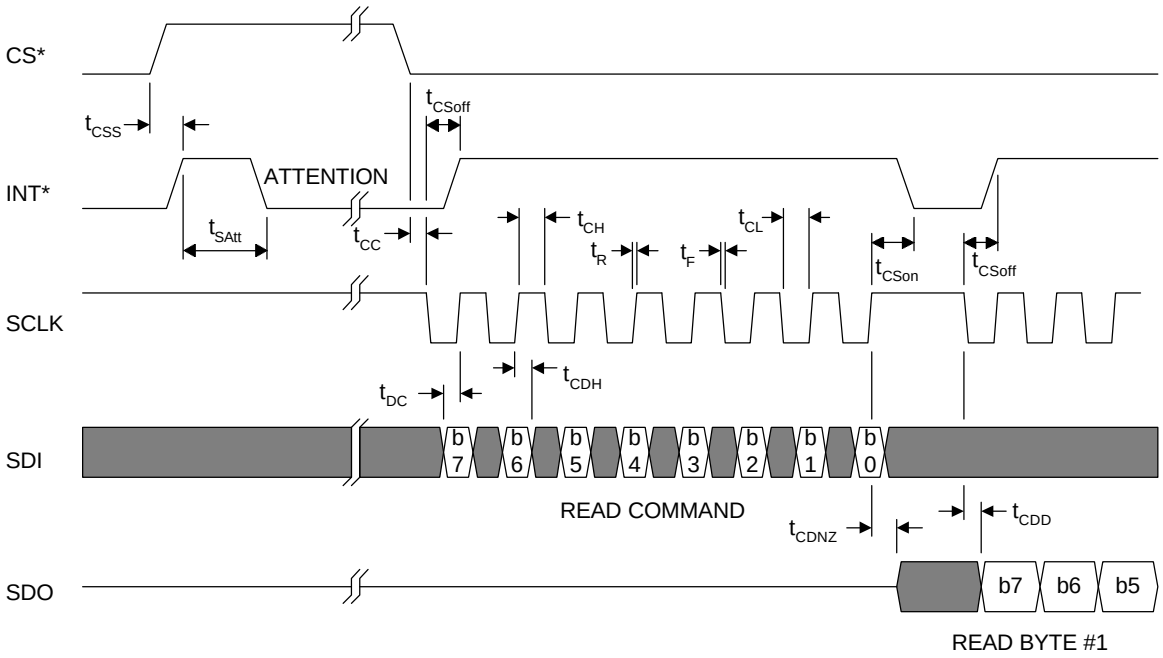


Figure 16. Attention/Read Byte #1 Timing Diagram

SPI Timing Parameters and Characteristics

Table 4. Timing Characteristics

(-40°C to 85°C; V_{CC}=5V)

Symbol	Parameter	Min	Max	Units	Notes
t_{RESET}	RESET to READY delay		10	ms	
t_{XFER}	Control Transfer Delay (1 msec typ.)		10	ms	
t_{SCH}	CS* Inactive Time	0		µs	2
t_{CC}	CS* to SCLK Setup	50		ns	2
t_{CL}	SCLK Low Time	250		ns	2
t_{CH}	SCLK High Time	250		ns	2
t_R, t_F	SCLK Rise and Fall		200	ns	
t_{CSon}	SCLK to INT* Setup		500	µs	1,2
t_{CSoff}	INT* to SCLK Delay		250	ns	1,2
t_{DC}	SDI to SCLK Setup	50		ns	1, 2
t_{CDH}	SCLK to SDI Hold	70		ns	1, 2
t_{SAtt}	INT* Attention Delay	2		µs	
t_{CSS}	CS* to INT* Delay		100	ns	1, 2
t_{CDNZ}	SCLK to SDO Delay		500	µs	1, 2, 3
t_{CDD}	SCLK to SDO Delay		100	ns	1, 2, 3
t_{CDHO}	SCLK to SDO Hold	70		ns	1, 2
t_{DSZ}	INT* to SDO High Z Setup		500	µs	1, 2

NOTES:
1. Measured at V_{IH}=2.0V or V_{IL}=0.8V and 10msec rise maximum rise and fall time.
2. Measured with 50 pF load.
3. Measured at V_{OH}=2.4V or V_{OL}=0.4V Modes.

SSC P300 Mechanical Specifications

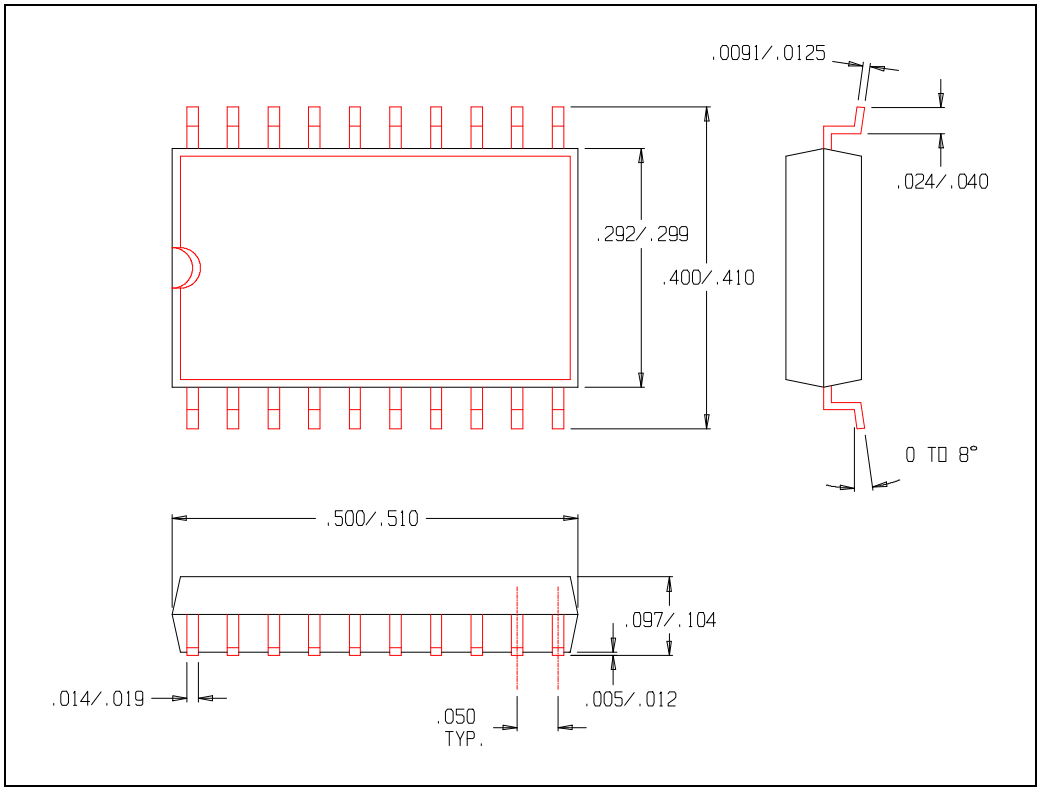


Figure 17. SSC P300 IC Mechanical Drawing

Supplemental Information

- White paper #0027 “CEBus Power Line Encoding and Signaling”
- Application Note #0069 “Using Intellon’s Serial Peripheral Interface”
- Application Note #0064 “Your First CEBus Packet”
- “Intellon SSC P200 and P300 Hardware Design Reference”

Ordering Information

SSC P300	PL Network Interface Controller

Intellon
CONNECTING THE FUTURE™

5100 West Silver Springs Boulevard
Ocala, Florida 34482

Phone: (352) 237-7416
Fax: (352) 237-7616

Internet
http://www.intellon.com
ftp://ftp.intellon.com