AltaCore-1553™
MIL-STD-1553 Protocol Engine Specifications/Users Manual

Part Number: 14101-00000-G5
Cage Code: 4RK27 • NAICS: 334118

Alta Data Technologies LLC
4901 Rockaway Blvd, Building A
Rio Rancho, NM 87124 USA
(tel) 505-994-3111 • www.altadt.com

CUSTOMER NOTES:

Revision Control History

Rev G5 Release Date: 9 September 2014

Note to the Reader and End-User:

Contact:

We welcome comments and suggestions. Please contact us at 888-429-1553 (toll free in US) or 505-994-3111 or visit our web site for support submit forms at www.altadt.com or email us at alta.info@altadt.com or alta.support@altadt.com.

AltaCore-1553™

Table of Contents

<~~~>

AltaCore-1553™
Document Introduction

Introduction

This document is the Specifications/User's Manual Document for Alta Data Technologies' (ADT) ***AltaCore-1553*** Notice II-IV compliant protocol engine (PE) product (and supports most pre-notice II standard derivations, such as 1553A). This document specifies the design of the product and provides an overview for the reader who wants a detailed understanding of the PE 1553 design. The reader should also reference the ***AltaAPI***manual (found on the CD or Alta web site) and source code for detailed descriptions of the software/backplane interface to ***AltaCore***.

> **NOTE:** For most customer applications, the PE's design and processes are transparent to the application as the ***AltaAPI***manages PE setup and execution. Most customers can skip this entire manual and simply refer to the ***AltaAPI***manual and example programs for their 1553 application requirements. It is also recommended that the reader review ***AltaView*** and ***AltaRTVal*** products to learn about advanced avionics analyzers and validation products.

This document is divided into sections that describe the major data structures of ***AltaCore-1553***:

- Introduction (This Section)
- Global Registers (Card Level)

- PE Channel Root Registers (Channel/Device Level)
- Bus Controller (BC)
  - Playback (PB)
  - Signal Generator (SG)
- Remote Terminal (including Map Monitor - RT)
- Sequential Monitor (SM)
- Common Data Packet (CDP)
- Interrupts (Int)

This Introduction section provides an overview of the ***AltaCore*** architecture and should be reviewed before diving deeper in to the data structures sections.

Terminology

- This manual assumes the reader is very familiar with the 1553B Notice II standard and popular variants. Please see the Alta web site to download our tutorial and reference and the 1553B standard (and obtain links to the SAE site to purchase the latest version).

- msec = milli seconds; µsec = micro seconds; ηsec = nano seconds.
- An Alta **card/board** can contain one or more ***devices***
- A ***device*** is an independent functional unit. Each 1553 ***channel*** is a ***device***
- A 1553 ***channel*** is a complete dual-redundant interface (with bus A and bus B connections) to the 1553 bus
- The ***AltaCore-1553 Protocol Engine (PE)*** implements the 1553 protocol for an individual 1553 ***channel***
- When discussing "**Global**" registers or values, these apply to ALL channel/devices. The term "**ROOT"** register or value is for a channel/device.

AltaCore-1553 Architecture and Document Basics

AltaCore is a 32-bit based PE that supports 1-4 1553 dual redundant, independent channels. The pure FPGA 32-bit base design greatly enhances backplane/processor throughput as compared to most 16-bit 1553 interfaces. ***AltaCore*** has a Common Data Packet (CDP) data structure, which as its name suggests, is a common data structure for BC, RT and Monitor data buffers. CDP's provide maximum flexibility between 1553 functions and greatly reduces host application complexity – reducing integration time. Each of ***AltaCore***'s BC, RT, Monitor and Interrupt functions have several execution options to provide the most advanced 1553 interface available on the market.

Memory Mapped Architecture

The ***AltaCore*** protocol engine (PE) utilizes dedicated on-chip (FPGA on-chip memory) and/or on-board Extended Memory (depending on the configuration).

The memory of the PE/Card appears as a contiguous memory map to the host backplane regardless of the on-chip or on-board configuration (additional PE FPGA logic provides the arbitration for host access and various PE channel access). For most Alta card designs, an ***AltaCore*** product uses high speed Quad Data Rate RAM, but the design is flexible to be integrated to most common RAM bus architectures.

The ***AltaAPI***or customer's application addresses each 1553 channel through memory offset or software pointer schemes. The ***AltaAPI***User's manual provides details of host driver and memory mapping requirements. Your Alta Product Hardware Manual (on the CD or the Alta web site download) details the memory configuration for your product (most card level products have one mega byte of dedicated on-board memory per 1553 channel), pin-outs and special configurations. Figure Intro-1 show the basic memory mapping configuration for a 4 channel 1553 interface with one megabyte of RAM per channel (standard configuration for most PCI, PCI Express, PMC, cPCI, etc…). Special configurations may vary.

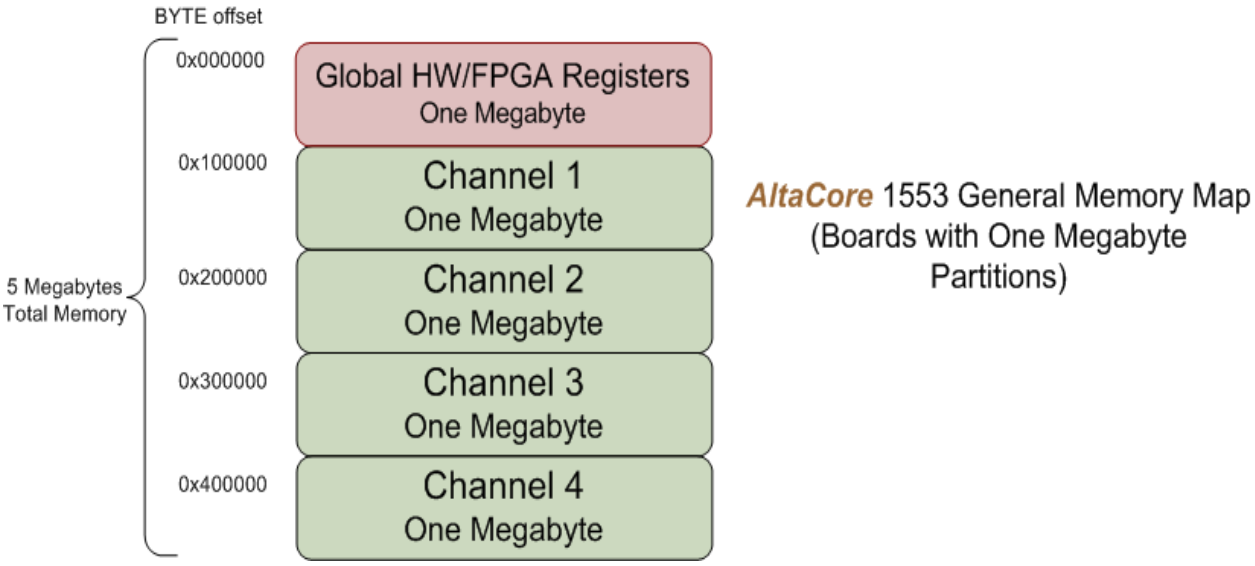


Figure Intro-1: Basic ***AltaCore*** Memory Map

Root Data Structures

Each of the major data structure elements (BC, RT, Monitor and Interrupt) begins with setup words known as ***ROOT*** registers. Each channel has a common Root register set called PE Channel Root registers that provide setup and control information for all 1553 functions of that channel. Each of the 1553 functions of a channel also have a Root register set that provides the base setup and address pointers to data structures to control the respective 1553 function and buffer the bus data (in CDPs). (Root registers reside in the FPGA on-chip memory and have fixed address offsets relative to the card's and channel's base address.)

BC, RT, Monitor and Interrupt Data Structures: Control Blocks, CSRs and CDPs

BC (including Signal Generation and Playback), RT, Sequential Monitor and Interrupt functions have two levels of data structures that control execution: Root Registers and Control Blocks (in extended, on-board memory) that direct sub-functions and provide 1553 data buffering. Each Root level function and respective Control Block have Control and Status Registers (CSR's) that provide setup options and execution feedback. Control Blocks for each function will reference (point to) linked-list Common Data Packets (CDPs) that provide 1553 data buffering and time stamping. (Most of these data structures reside in on-board high speed RAM memory external to the FPGA.)

All options and data structures are detailed in the following sections.

NOTE: Several data structures of the PE are pointers to other tables or buffer data structures. The user must take great care to never populate a utilized structure with a null pointer or improper value (a value outside the on-board memory range). For example, if an RT is turned-on for responses and the Root or secondary pointer arrays are not properly programmed, then the PE will have unpredictable results (like any design that has a bad pointer reference). The PE will not check for null or improper values and bad pointers will cause unpredictable behavior. The ***AltaAPI***properly manages and provides allocation/destruction functions for the user application.

Memory Offset References

Memory references in this document are in byte offsets. The PE memory (a common word) is a 32-bit atomic machine, 4-byte aligned.

Packed Word and Bit Data Structures

AltaCore data structures are typically packed 32-bit word structures that may have embedded packed bit or single bit settings. The right most bit is LSB bit 0 (zero) and the left most bit is MSB 31. The general rule is that a bit value of 1 (one) is an "on/set/positive" and a 0 (zero) is "off/clear/negative."

All data structures will be labeled "**W**" (Write) if the user's application (or ***AltaAPI***) has the capability to write the value of the respective data element (word, bit or packed bits). Unless noted in the description, assume all data structures may be read by the user's application. The safest way to set control bits is to read the respective CSR/Register and OR in the selected bits and then write back. Great care must be taken to not set or change values of active registers during execution.

Execution Times, Bus Timing, Time Tags

Execution times provided herein are respective to the PE and do not include backplane transfer times. In general, after execution of a transmit function, the user should allow 10 µsecs settling time (process overhead) for accurate future transmission timing (for example, the user needs to provide 10 µsecs of non activity at the end of a BC minor frame to assure accurate Minor Frame/Minor Frame timing to 500 ηsec). There are some special cases that require more than 10 µsecs and the reader should reference the respective section for detailed timing requirements. Generally, Intermessage Gap times of 4-6 µsecs are required for extremely accurate BC, Playback and Signal Generation Timing.

The user will have several options for programming gaps (dead bus between words or messages). These values are typically 32-bit, 100 ηsec tick values.

All BC, RT and Monitor functions have data packet time stamps (a time value placed on the message when decoded) and these are 64-bit, 20ηsec time ticks (stored in two 32-bit words – Time High & Low).

Reserve Bits and Words

Many bits and words are "Reserved" for future use or are used by the PE execution. All Reserved BITS should assume a value of zero. Reserve Words should not be written to. If a reserve word is part of a larger data structure word packet, assume a value of zero for initialization, but then do not write to this word. For updating packed bit-word structures, the user's application should first read the word and then logically OR-in the desired one bits (or AND-out the zero bits) and then write back to the same location (please see the ***AltaAPI*** source for numerous examples) The API source code is provided and provides the proper set and rules for memory access – for custom applications, please use the API as a design guide.

Figure Color Codes

In the following figures, a brown outline/white box signifies a bit/word/area that is programmed by the User. The light gray is usually a reserved bit/word/area and should not be changed by the User (the PE will update or use this element for processing). The CDP diagrams show mixed brown/gray words that are update by both the PE and User programming. Most gray/reserved areas should be set to zero and most of these areas can be read by the User without affecting PE processing.

AltaCore-1553 Global Card Level Registers

Global Card Level Registers

The area that occupies the first mega byte of the card memory map contains backplane and global card level settings and status values that affect processing for all channels. This area also contains the digital discrete, time and Built-In-Test (BIT) settings and values of the card.

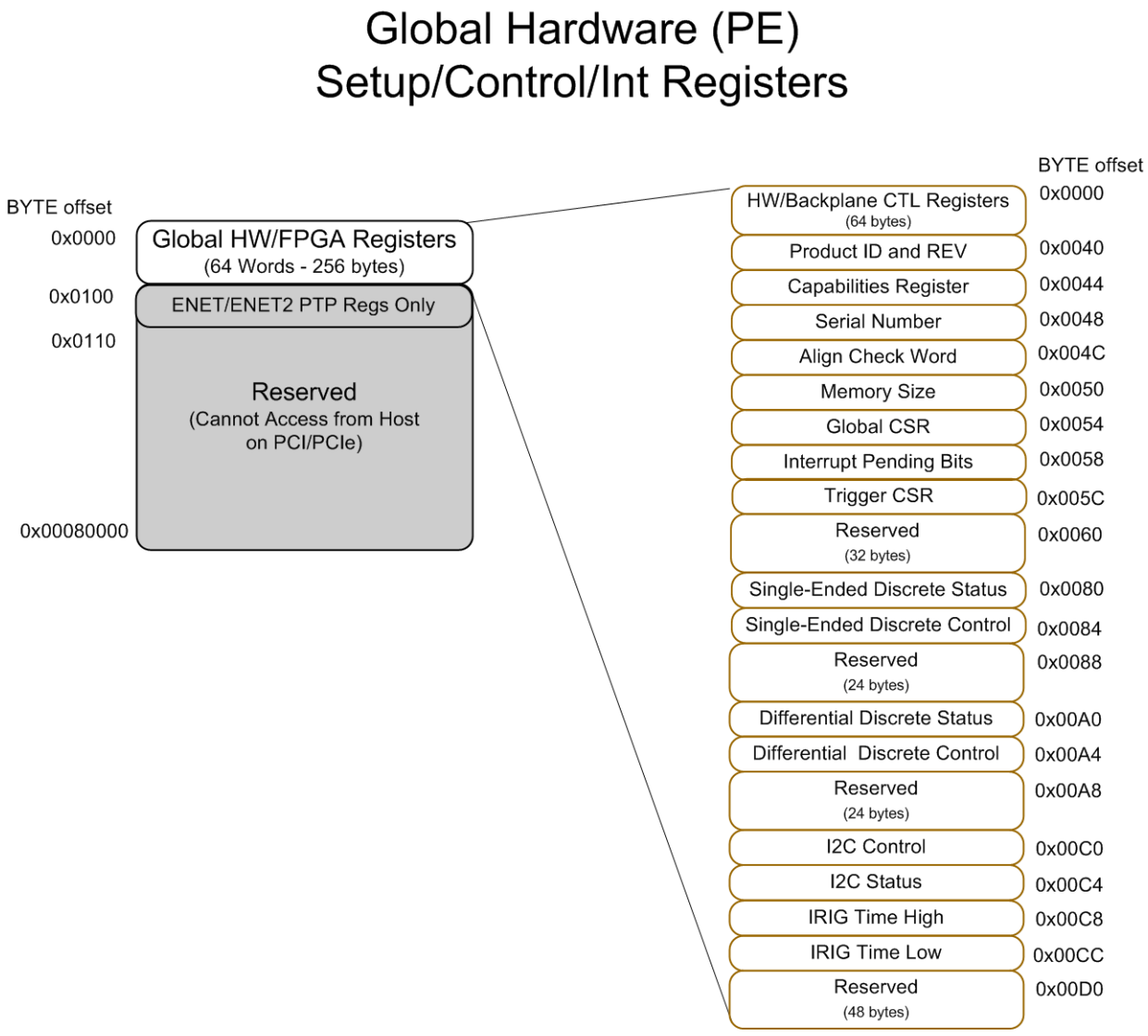


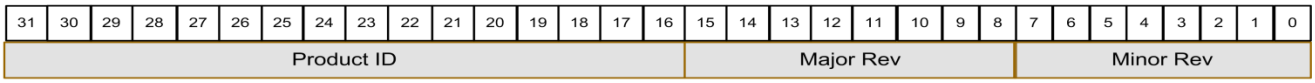
Figure Global Registers-1: Global Card Level Registers

HW/Backplane CTL Registers (16 words - 0x00000000-0000000F)

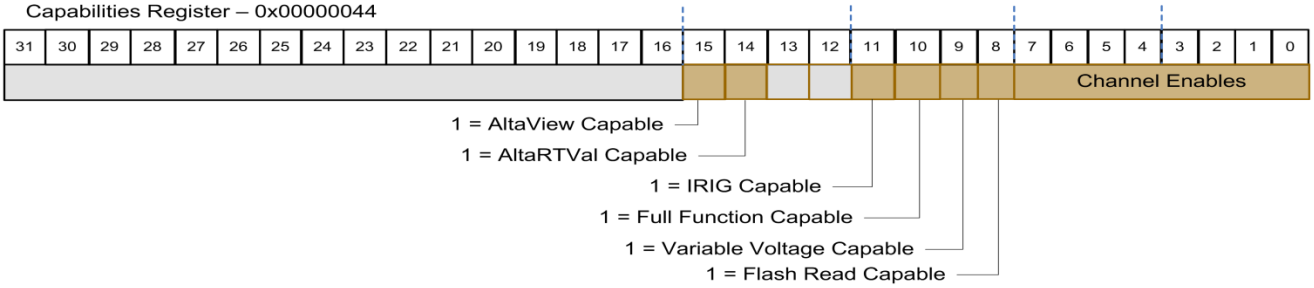
The first 16 words of the memory map are reserved for backplane configuration registers. Please see your product's Hardware Manual for details on these registers.

Global Registers

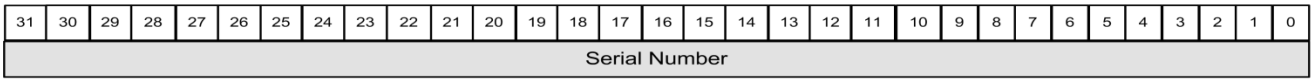
Prod ID and REV – 0x00000040



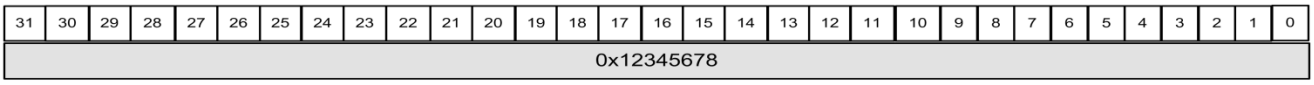
Capabilities Register – 0x00000044



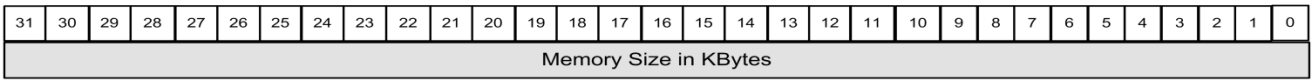
Serial Number – 0x00000048



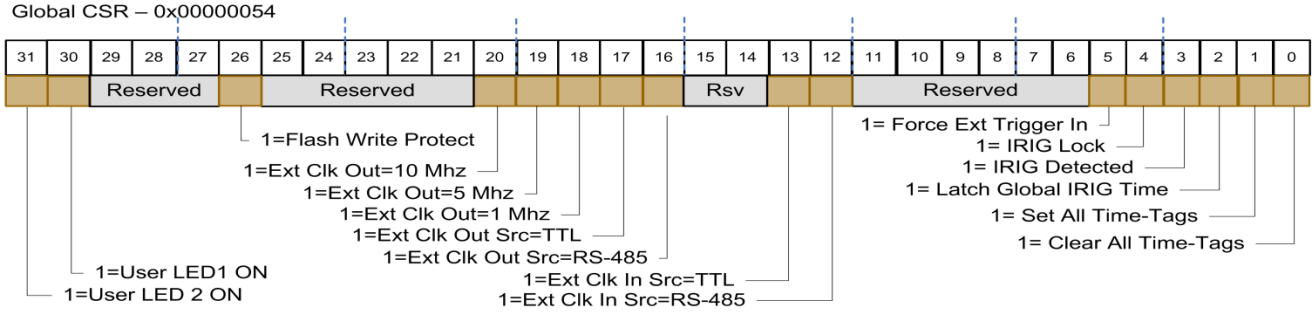
Align Check Word – 0x0000004C



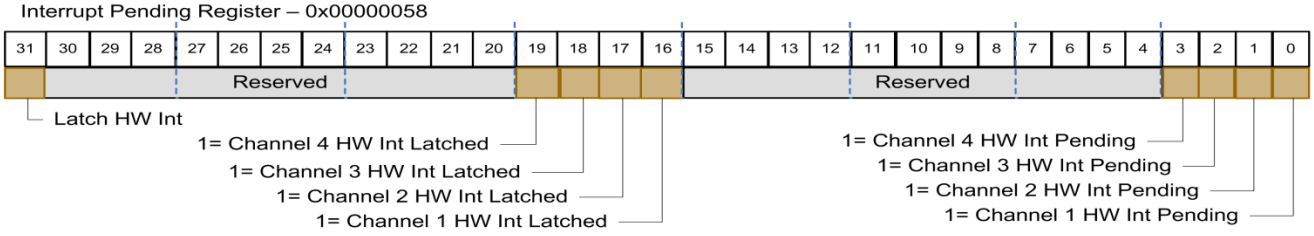
Memory Size – 0x00000050



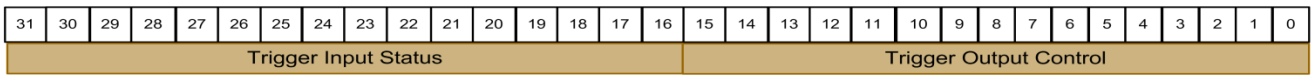
Global CSR – 0x00000054



Interrupt Pending Register – 0x00000058



Trigger CSR Register – 0x0000005C



*Note: The number of input and output triggers is equal to the number of channels on the board.

Global Registers (cont.)

Single-Ended Discrete Status Register – 0x00000080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bi-Directional Discrete Input Status (1=not active, 0=active)

Single-Ended Discrete Output Register – 0x00000084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bi-Directional Discrete Output Control (1=drive output low)

Differential Discrete Status Register – 0x000000A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Discrete Input Status (1=active, 0=not active)															

Differential Discrete Output Register – 0x000000A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Output Enable (1=Enable Transmit)																Output Control (1 = drive output high)															

*Note: The number of single-ended and differential discretes varies from board to board. See individual board HW manuals for details on discrete counts.

I2C Control Register – 0x000000C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												Reserved												Write Data							

1= Start Command
1= Stop Command
1= Read Command
Write Command

I2C Status Register – 0x000000C4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																Reserved								Read Data							

1= ACK Received
1= Busy
1= Transfer In-Progress

IRIG Time High– 0x000000C8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												BCD Years												BCD Days							

IRIG Time Low – 0x000000CC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BCD Hours								BCD Minutes								BCD Seconds							

Figure Global Registers-2: User Global Registers

Product ID and Rev: 0x00000040

This register provides the ***AltaCore***/FPGA Program ID and Revision.

Capabilities Register: 0x00000044

This register provides settings for product options and capabilities. This allows Alta software to verify purchased or configured configuration.

Bits 0-5: Channel Enables

The channel enable bits indicate which channels are enabled on the board. For instance, if a two channel board was purchased, Bits 0 and 1 should be set to indicate that Channels 1 and 2 are enabled.

Bit 8: Flash Read Capable

This bit is set to a one if the Flash Read option is enabled on the board. This option enables the user to program setup data into on board flash. On power-up or reset, the setup data is copied from Flash to PE memory space. For more information regarding this feature, please contact Alta.

Bit 9: Variable Voltage Capable

This bit is set to a one if the Variable Volt option is enabled on the board.

Bit 10: Multi-Function Capable

This bit is set to one if the Multi-Function option is enabled on the board. When the Multi-Function bit is set to a one, BC, RT, and BM functions can be run simultaneously. When this bit is zero, the PE is operating in Dual-Function mode where only BC/BM or RT/BM functions can run simultaneously. For example, BC and RT functions cannot be enabled at the same time.

Bit 11: IRIG Capable

This bit is set to one if IRIG capability is enabled on the board.

Bit 14: ***AltaRTVal*** Capable

This bit is set to one if ***AltaRTVal*** capability is enabled on the board. This option must be purchased on a board-by-board basis to allow ***AltaRTVal*** software to run with the board.

Bit 15: ***AltaView*** Capable

This bit is set to one if ***AltaView*** capability is enabled on the board. This option must be purchased on a board-by-board basis to allow ***AltaView*** software to run with the board.

Bits 16-31: Reserved

Serial Number: 0x00000048

This register provides the Born on Date and Incremental Serial Number of the card (the two left most segments seen on the card label – “MMYY-NUMBER”). The 16 MSBs make-up 4 BCD nibbles that provide the MMYY (M=Month, Y=Year) digits. The 16 LSBs represent the serial number 65535/99999 value.

Alignment Test Register

This register provides a simple 1,2,3,4,5,6,7,8 value in each nibble so that an operating system driver developer can check for proper word/byte alignment.

Memory Size: 0x00000050

This register tells the User the size in K bytes for the total card. Only the lower 16 bits are used and for most cards this should be a value of 0x00001400 (5120 Kbytes = 5 Mbytes).

Global CSR: W: 0x00000054

This word is provided for user applications that want to control all channel time tags and HW Interrupt enables from one location. All of these settings are duplicated at the channel level for multi applications. This is a packed bit structure whose bits are defined in the following paragraphs.

Bit 0: Clear All Time Tags: W

This bit is set to one by the user to direct the PE to clear/zero all channel time tags (PE Time High & Low clocks), which are utilized to time tag all 1553 messages. Time clearing occurs <1µsec from this bit being set. This bit is self clearing.

Bit 1: Set all Time Tags: W

This bit is set to one by the user to direct the PE to load all local PE Time High & Low registers (offset 0x001C/20) at the same time (with the user loaded values). This allows all channels to have their clocks jammed with user time simultaneously. Otherwise, the user application would need to perform this bit setting for each PE channel and time values would be skewed. Time loading occurs <1µsec from this bit being set. This bit is self clearing.

Bit 2: Latch Global IRIG Time: W

This bit is set to a one by the user to load the current IRIG time into the IRIG Time Global Registers (offsets 0x00C8/CC). This bit is self clearing.

Bit 3: IRIG Detected: R

This read only bit set to a one indicates that activity has been detected on the IRIG input signal. Once activity has been detected this bit will remain set until no activity has been detected for 5 seconds. This bit does not mean IRIG lock has

occurred (see bit 4 below), it is simply an indicator that “something” has been detected on the input signal within the last 5 seconds.

Bit 4: IRIG Lock: R

This read only bit set to a one indicates that the IRIG decoder has locked on a good IRIG signal. To achieve IRIG Lock the IRIG decoder must detect at least one valid IRIG time frame. IRIG Lock will be lost if at any point the IRIG decoder detects an invalid time frame. Once IRIG Lock is lost this bit will get set to zero and the IRIG decoder will go into search mode looking for another valid IRIG time frame.

Bit 5: Force Ext Trigger In: W

This bit set to a one by the user forces an input trigger to all channels simultaneously. Setting this bit overrides the external trigger signal on each channel. This bit provides the means for a software trigger as opposed to and external HW trigger. This bit is self-clearing.

For example, the user can setup the BC on each channel to wait for an external input trigger. Once the BCs are setup, enabled and waiting for a trigger, this bit can be set to start all BCs simultaneously.

Bits 6-11: Reserved

Bit 12: Ext Clk In Src=RS-485: W

Setting this bit to a one will select the differential “DDISC1” RS-485 lines as the external clock input. See the individual board HW manual for pin assignment for the DDISC1 (+) and (-) signals.

Bit 13: Ext Clk In Src=TTL: W

Setting this bit to a one will select the differential Ext TTL Clock line as the external clock input. See the individual board HW manual for pin assignment for the External TTL Clock I/O Signal.

Bits 14-15: Reserved

Bit 16: Ext Clk Out Src=RS-485: W

Setting this bit to a one will select the differential “DDISC1” RS-485 lines as the external clock output. See the individual board HW manual for pin assignment for the DDISC1 (+) and (-) signals.

Note that if the RS-485 option is used for the external clock, the input and output clocks use the SAME PINS. Therefore, the input clock for all channels on the board are automatically connected to the output clock, so you can test the

external clock on those channels without any external connection. Of course, if you need to synchronize with other boards then one board should be configured to output a clock signal and connected to the appropriate external clock input lines on the other boards (which are configured to accept the external clock signal and will NOT generate an output clock).

Bit 17: Ext Clk Out Src=TTL: W

Setting this bit to a one will select the differential Ext TTL Clock line as the external clock output. See the individual board HW manual for pin assignment for the External TTL Clock I/O Signal.

Note that if the Ext TTL Clock option is used for the external clock, the input and output clocks use the SAME PIN. Therefore, the input clock for all channels on the board are automatically connected to the output clock, so you can test the external clock on those channels without any external connection. Of course, if you need to synchronize with other boards then one board should be configured to output a clock signal and connected to the appropriate external clock input lines on the other boards (which are configured to accept the external clock signal and will NOT generate an output clock).

Bit 18: Ext Clk Out=1 MHz: W

Setting this bit to a one will set the Ext Clock output frequency to 1 MHz.

Bit 19: Ext Clk Out=5 MHz: W

Setting this bit to a one will set the Ext Clock output frequency to 5 MHz.

Bit 20: Ext Clk Out=10 MHz: W

Setting this bit to a one will set the Ext Clock output frequency to 10 MHz.

Bits 21-25: Reserved

Bit 26: Flash Write Protect: R

If this bit is set to a one all non-volatile memory on board will not be writable. Generally this means the board was ordered with a –N option.

Bits 27-29: Reserved

Bit 30: User LED1: W

Setting this bit to a one will turn on User Led1. See the individual board HW manual for User LED1 locations.

Bit 31: User LED2: W

Setting this bit to a one will turn on User Led2. See the individual board HW manual for User LED2 locations.

Summary of Multi-application verses Single Application Interrupt Options with *AltaCore*

Multi Applications Scenarios: (usually one application per channel for interrupts): All Interrupt signals are handled at the PE channel level. You only need to read the next section and the Interrupt Section for details.

Single Application Scenarios: You have a choice to handle interrupt management at the channel level or handle interrupt pending at the global level, which is why the following word and bits are provided. If you want to handle interrupts strictly at the channel level, then treat this as a multi-application scenario. If you want to handle interrupt pending signals at a single global level (which may save time on host reads/writes), then read the following paragraphs and the Interrupt Section of this manual for this supported option.

Global Interrupt Register: W: 0x00000058

This register is provided for applications that want to control all channel interrupt hardware actions from one location verses managing each PE channel individually (multi-application processes would need to manage each channel as an independent, logical device). These bits are duplicated at the PE channel level. This is a packed bit structure whose bit values are defined in the following paragraphs. A brief discussion follows to provide the general overview of the bit values and proper execution action of reading the bits.

This register provides a pending and latching bit value for each of the four possible 1553 channels. Bit 31 can be set to one by the user to “latch” active interrupt signals and immediately clear the cards channel interrupt pending signal to the host backplane/system (all channel interrupt pending signals, or bits, are ORed together to set the backplane interrupt pending signal). So bit 31 provides the application a single bit to clear the card’s interrupt pending bit to not hold-off other possible interrupt pending conditions in the system (from other cards). This would not be multi-application safe as independent application would need to service their respective channel: there is a mechanism to clear interrupt pending at the channel level, too.

In the PE Root Control Word (channel level), there is a “Hardware (HW) Interrupt On” bit that controls the interrupt pending condition to drive the card level backplane/system interrupt hardware line. Please see PE Root Registers – Control and Status Word for channel level interrupt settings.

Bits 0-3: Interrupt Pending

These bits, one per 1553 PE channel, are set to one by the PE when an interrupt event(s) has posted for the respective channel (please see the diagram for the channel mapping). When an interrupt has occurred (or when the host application conducts a polling event), the user application reads this register to see what channel caused the interrupt. When the channel level Interrupt Pending bit in the Root PE Status Word is cleared (set to zero), or when bit 31 of this word is set to one, then the respective pending bit is cleared (and if the hardware interrupt is turned on, then the backplane/system, interrupt pending signal is cleared). The user does not clear these bits to clear an interrupt signal.

Bits 4-15: Reserved

Bits 16-19: Interrupt Latched: W

These bits are provided as a copy of the pending bits in this word when the hardware interrupt signal is cleared (from writing a one to bit 31). After the user interrupt service routine has written a one to bit 31, these bits hold the current value of the pending bits (please see the diagram for the bit position for respective channel). This allows user applications that want to handle interrupts at the global level to quickly clear the hardware event from the backplane/system and then read these bits to see which channel caused the interrupt. The user application should clear these respective bits for the next interrupt event handler. Multi-application processes would read interrupt pending events only at the PE Channel Level (described above).

Bits 20-30: Reserved

Bit 31: Latch Hardware Int: W

This self clearing bit is set to one by the user to direct the PE to clear the hardware interrupt pending signal to the backplane/system, clear the interrupt pending bits in this word and copy their value to the interrupt latch bits in this word (all action described in the previous paragraphs). Software polling interrupt event handlers do not touch this bit.

Trigger CSR: W: 0x0000005C

This word is provided for user applications that want to know the state of the input triggers and is also used to override the output triggers. Normally the output triggers are controlled by the PE when the “generate output trigger” bit is set in a PE control register. The number of input and output triggers is equal to the number of channels on the board.

Bit 15-0: Trigger Output Control: W

Trigger outputs can be driven low (Active) by setting the corresponding bit in this register to a one. Bit0=Trigger1, Bit1=Trigger 2, etc.

Bit 31-16: Trigger Input Status:

These bits show the status of the input triggers. A one for a given bit indicates that the trigger is in-active and a zero indicates the trigger is active. Bit16=Trigger1, Bit18=Trigger 2, etc.

Discrete Configurations - README

See your product's Hardware Manual for Discrete Configuration and Pin-Outs. Products may vary in configuration and capability.

Single-Ended Discrete (SDISC) Status Register: 0x00000080

This word shows the input status of the bi-directional avionics discretes. A one for a given bit indicates that the discrete is in-active and a zero indicates the discrete is active. Bit0=Discrete1, Bit1= Discrete2, etc.

NOTE: The input voltage should not exceed 30 Volts. Exceeding this rating can damage the board.

NOTE: For MIL-STD-1760 External RT Address Discrete usage, one of the SDISC lines is reserved for "RT Enable." Please see your hardware manual for pin-outs for 1760 discretes and this enable line. If this line is not pulled-down (active), then the PE will not automatically use the other 6 SDISC lines (RT Address and Parity) to obtain the RT Address line. Please see the External RT Address discussion in the AltaAPI manual (PDF Bookmark: **Single RT and Multiple RT Configurations)** and contact the factory for the application note discussing External RT Address and 1760 usage.

Single-Ended Discrete Output Register: W: 0x00000084

This word is used to control the output state of the bi-directional avionics discretes. Discrete outputs can be driven low (Active) by setting the corresponding bit in this register to a one. Bit0=Discrete1, Bit1= Discrete2, etc.

NOTE: The bi-directional Avionics Discretes can sink up to 1 Amp when active. Exceeding this rating can damage the board.

Differential Discrete (DDISC)Status Register: 0x000000A0

This word shows the input status of the differential RS485 discretes. A one for a given bit indicates that the discrete is active and a zero indicates the discrete is in-active. Bit0=Discrete1, Bit1= Discrete2, etc.

Differential Discrete Output Register: W: 0x000000A4

This word is used to control the output state of the differential RS485 discretes.

Bit 15-0: Output Control: W

Discrete outputs can be driven high (Active) by setting the corresponding bit in this register to a one. Bit0=Discrete1, Bit1= Discrete2, etc. The corresponding transmit enable bit must be set before this bit is valid. Bit0=Discrete1, Bit1= Discrete2, etc.

Bit 31-16: Output Enable: W

These bits control the output enables of the uni-directional differential RS485. If a given bit is set to a one, the discrete output is enabled (transmit mode). If a given bit is set to a zero, the discrete output is disabled (receive mode). Bit16=Discrete1, Bit17= Discrete2, etc.

I2C Control Register: W: 0x000000C0

This word is used to control the I2C logic on the board. See the individual board HW manual for more details on this register. The user's application does not need to know details of this register – the ***AltaAPI*** handles all overhead and management of these registers – please see source code of ***AltaAPI*** for details.

I2C Status Register: 0x000000C4

This word provides the status of the I2C logic on the board. See the individual board HW manual for more details on this register. The user's application does not need to know details of this register – the ***AltaAPI*** handles all overhead and management of these registers – please see source code of ***AltaAPI*** for details.

IRIG Time High: 0x000000C8

This word shows current IRIG Time High value. To latch the current IRIG time in this word, the Latch IRIG Time (Bit 2) in the Global CSR should be set prior to reading this word.

IRIG Time Low: 0x000000CC

This word shows current IRIG Time Low value. To latch the current IRIG time in this word, the Latch IRIG Time (Bit 2) in the Global CSR should be set prior to reading this word.

NOTE: The IRIG times latched in the registers described above are one second behind. The user/API's code will need to adjust the time +1 second. Check your AltaAPI release notes for this +1 second adjust made to the IRIG read function.

IRIG Code Formats

Alta IRIG receive capability supports the following formats.

- IRIG-B002, Format B, DC Level Shift, no index count interval, BCDTOY
- IRIG-B122, Format B, Amplitude Modulated, 1khz Carrier, BCDTOY
- IRIG-B006, Format B, DC Level Shift, no index count interval, BCDTOY, BCDYEAR
- IRIG-B126, Format B, Amplitude Modulated, 1khz Carrier, BCDTOY, BCDYEAR

PTP IEEE-1588 Registers: 0x0100-0x010C

These 4 registers are for ENET/ENET2 devices for key PTP IEEE-1588 Time Sync and Control-Status (CSR) registers. ENET devices can synchronize time via 1588 grand masters in a similar manner to IRIG synchronization. If engaged (via user software API settings), this overrides IRIG sync decoding.

For details on implementation, please contact Alta for access to the PTP IEEE-1588 Design Implementation Application Note and reference the appropriate sample C programs. Contact Alta at alta.support@altadt.com or 888-429-1553 x2.
<~~~>

AltaCore-1553 Root PE Channel Registers

Introduction

There are several control and status settings at the channel level for 1553 functions and events. The following figure provides an overview of the Root registers memory map. This section will describe the Root PE Registers that are common for all 1553 BC, RT and Monitor functions. The subsequent sections of manual will detail the Root registers for the respective functions (as shown in the figure below).

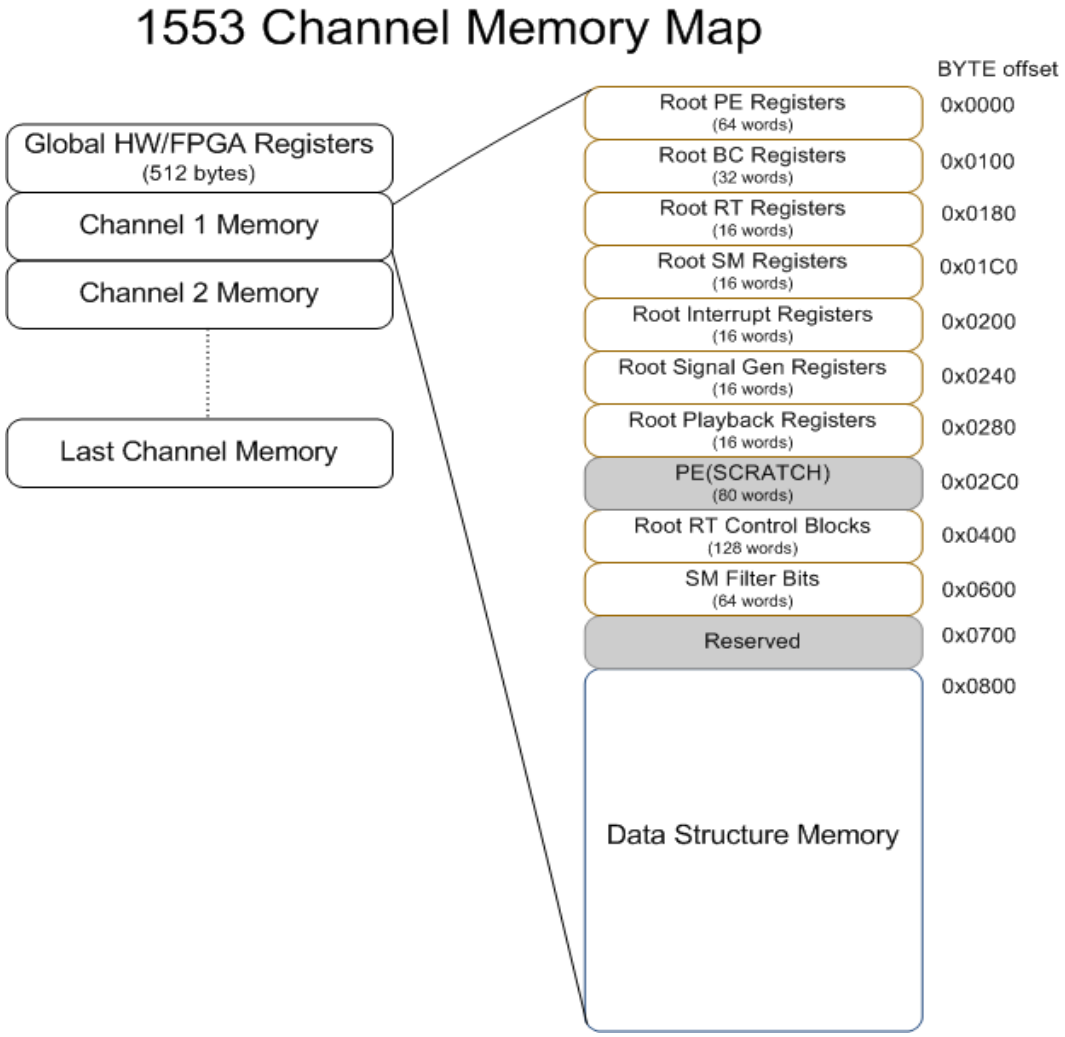


Figure PE Root-1: Root PE Registers (per Channel)

Please see Figure PE Root-2 (on the next page) for Root PE Registers diagram.

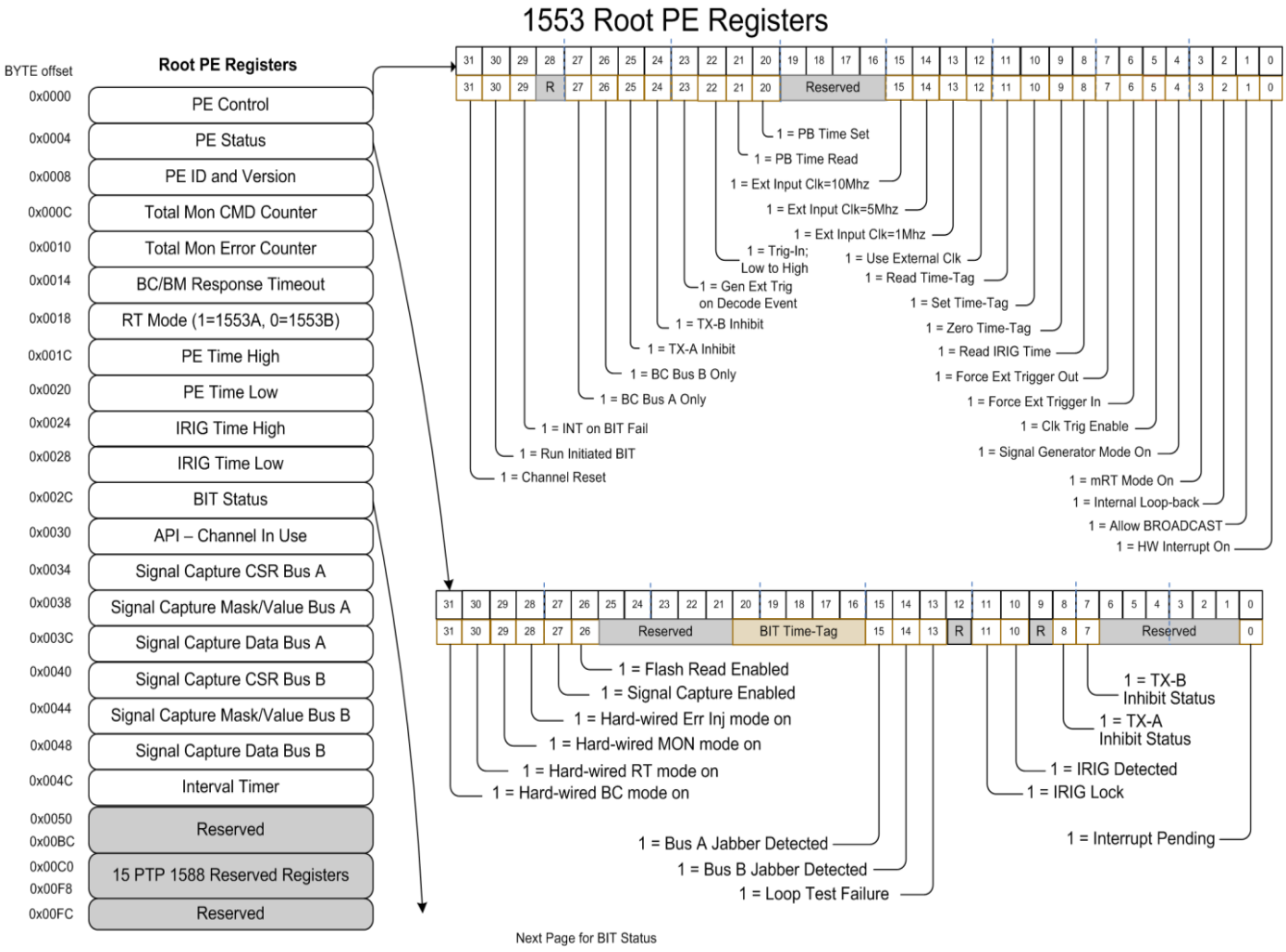


Figure Root PE-2: Root PE Registers: Control and Status Word

Root PE Control Word: W: 0x0000

This register is a packed bit structure and provides key PE control for the user. The following paragraphs detail the bit settings.

Bit 0: HW Interrupt On: W

This bit is set to one by the user to direct the PE to enable hardware interrupts to the backplane (or interrupt line of an FPGA core build). Interrupt events from BC, RT, Monitor, Playback and Signal Generator are logged in the Interrupt Queue regardless of this setting (see the respective section for details). Setting this bit to zero requires the user's host application to poll the interrupt queue to service events. Setting this to a one will require the user's host application to have an operating system interrupt service routine (ISR) to handle the backplane interrupt and then read the Interrupt Queue to service the events.

Bit 1: Allow Broadcast: W

This bit is set to one by the user to direct the PE to allow RT Address 31 to be treated as a standard 1553B Broadcast message. Setting this bit to a zero will direct the PE to treat address 31 as a normal RT address and all PE functions will expect BC messages addressed to RT 31 to have a normal RT Status response. This bit affects BC, RT and Monitor operations for proper responses (RT mode) and message decode mode (BC and Monitor). Normal setting for 1553B is one – 1553A would have this set to zero.

Bit 2: Internal Loop Back: W

This bit is set to one by the user to direct the PE to invoke a loop-back transmission check of all BC and RT transmissions. The PE has a built-in loop-back comparison circuit that verifies that word values have been properly through the transceiver (prior to the transformer) of the card. This feature provides greater assurance to the user that the card is functioning properly. It is recommended to set this bit to one for most operations.

Bit 3: mRT Mode: W

This bit is set to one by the user to direct the PE to operate in multi RT (mRT) mode. If the channel is in single RT mode, then all data buffer setup (RT Filter Table, SA-BC Control Block and respective CDP) are referenced at the RT 0 (zero) data structures. For most application this should be set to one. Single RT mode is for special application/configuration modes.

Bit 4: Signal Generator Mode On: W

This bit is set to one by the user to direct the PE to start the Signal Generator (see the Signal Generator (SG) section of this document for details). The BC cannot be operating during SG mode.

Bit 5: Clock (Clk) Trigger Enable: W

This bit is set to one by the user/API to direct the PE to NOT allow the time tag or playback clock increment until an external trigger has been detected. This bit is self clearing. This feature is very useful for synchronizing clocks between channels and different I/O within a system. For example, the user may want to use a 10 MHz clock (TTL or RS-485), jam the start times for all channels, and then provide a trigger for all channels to allow their clocks to start incrementing. This can be used to keep time tags and the playback transmission in sync to 1-2 clock pulses.

Bit 6: Force External Trigger In: W

This bit is set by the user to direct the PE to set external triggers for all devices and any of their PE functions that may be stalled for ext trigger (for example the user can set Wait for Ext Trigger on BC and BM). This bit is extremely useful to ensure all BCs start within 5 µsecs of each other. Most competitive products

require software control of each channel resulting in huge delays between channels. This bit is self clearing.

Bit 7: Force External Trigger Out: W

This bit is set by the user to direct the PE to override its output trigger. Normally the output trigger is controlled by the PE when the “generate output trigger” bit is set in a PE control register. This bit allows a software application to generate an output trigger on demand. This bit is self clearing.

Bit 8: Read IRIG Time

This bit is set to one by the user to direct the PE to the latch (for reading) the PE’s internal 64-bit, 20 ηsec time and the most current IRIG Time High & Low values to the Root PE Registers Time High & Low (offsets 0x001C/20) and IRIG Time & Low registers (offsets 0x0024/28). See IRIG Time High & Low Registers later in this section for more discussion on IRIG time. The latency is <1µsec. If IRIG Time is not being used, then the value in the IRIG time registers is unpredictable. This bit is self clearing. Time Tag Discussion (Time High & Low)

Each PE Channel has a 64-bit, 20 ηsec internal timer that is used to time tag all messages in all modes (Time High & Low in every Common Data Packet, CDP, see the CDP Section of this manual). The PE provides the user a 2-word (Root Time High & Low – Root PE Register offset 0x001C/20) window to access the internal PE timer. These Root Time High & Low registers are used to program or read the internal timer per the bit selections discussed in the ensuing paragraphs.

NOTE: The IRIG times latched in the registers described above are one second behind. The user/API’s code will need to adjust the time +1 second. Check your AltaAPI release notes for this +1 second adjust made to the IRIG read function.

Bit 9: Zero Time-Tag: W

This bit is set to one by the user to direct the PE to zero/clear the PE Time High & Low registers (offsets 0x001C/20), which is used for time stamping messages. The PE clears this bit when the clock registers are cleared. This bit is self clearing.

Bit 10: Set Time-Tag: W

This bit is set to one by the user to direct the PE to read the user written values in Time High & Low and jam/set the internal PE time registers (offsets 0x001C/20), which is used for time stamping messages. The PE will clear this bit when the read/jam is finished. The delay from this bit being set to the internal time tag register being jammed is <1µsec (latency). This bit is self clearing.

Bit 11: Read Time-Tag: W

This bit is set to one by the user to direct the PE to latch the internal time to the PE Time High & Low registers (this allows the user to read the time values and is an excellent “heart beat” check on the PE). The latency is <1µsec. The PE will clear this bit when the time values have been written. This bit is self clearing.

Bit 12: Use External Signal:W

This bit is set to one by the user to direct the PE to use the selected (RS-485 or TTL) external input to drive the clock count (PE Time High & Low).

Bit 13: Ext Input Clk = 1 MHz : W

Setting this bit to a one will indicate to the PE that the external input clock frequency is 1 MHz.

Bit 14: Ext Input Clk = 5 MHz: W

Setting this bit to a one will indicate to the PE that the external input clock frequency is 5 MHz.

Bit 15: Ext Input Clk = 10 MHz: W

Setting this bit to a one will indicate to the PE that the external input clock frequency is 10 MHz.

Bit 16: RS-485 Trigger In Enable: W

Setting this bit to a one will indicate to the PE that the second RS-485 discrete pair (DDISC2) should be used for trigger input (including clock start trigger). This will redirect from the standard TRIG IN single-ended signal for the channel. This feature is only available on select cards; please review your hardware manual for capability and pin-outs.

Bit 17: RS-485 Trigger Out Enable: W

Setting this bit to a one will indicate to the PE that the second RS-485 discrete pair (DDISC2) should be used for trigger output. This will redirect from the standard TRIG OUT single-ended signal for the channel. This feature is only available on select cards; please review your hardware manual for capability and pin-outs.

Bits 18-19: Reserved

Bit 20: PB (Playback) Time Read: W

This bit is set to one by the user/API to direct the PE to latch the current PB clock time to the PE Root Time High and Low Registers (offsets 0x001C/20) so the user application can read the time.

Bit 21: PB (Playback) Time Write: W

This bit is set to one by the user/API to direct the PE to read the Root PE Time High and Low Registers and jam the PB clock time (offsets 0x001C/20) so the user can set the PB clock time.

Bit 22: Trig (Trigger) In Active Low to High: W

This bit is set to one by the user/API to direct the PE to invert the normal trigger polarity to an active low to high signal (normally a trigger event is active from high to low). The trigger pulse (in this case a high signal) should be active for >2usec.

Bit 23: Gen (Generate) Trig (Trigger) on Decode Event: W

This bit is set to one by the user/API to direct the PE to generate an output trigger when a decode event (when the 1553 message decoder) is waiting for time gapped event (i.e. waiting for Status Word response). This can be very useful to trigger a Signal Generator output to transmit an RT Response Test Pattern (for detailed BC protocol testing).

Bit 24: TX B Inhibit

This bit is set to one by the PE to if the transceiver has been hard wired to inhibit transmission on Bus B. This is provided so the user's application can read inhibit status for specially configured cards that have transmission disabled for monitor only applications.

Bit 25: TX A Inhibit

This bit is set to one by the PE to if the transceiver has been hard wired to inhibit transmission on Bus A. This is provided so the user's application can read inhibit status for specially configured cards that have transmission disabled for monitor only applications.

Bit 26: Bus B Only: W

This bit is set to one by the user/API when Bus A decoding is to be ignored (encoding or transmission will still occur on this bus). This setting is only appropriate for validation testing (bus switching tests).

Bit 27: Bus A Only

This bit is set to one by the user/API when Bus B decoding is to be ignored (encoding or transmission will still occur on this bus). This setting is only appropriate for validation testing (bus switching tests).

Bit 28: Reserved

Bit 29: INT on BIT Fail :W

This bit set to one will cause an interrupt to be generated when a BIT failure is detected. Note that an interrupt will only get generated on each instance that bits 7-0 of the Bit Status Register (offset 0x002C) are set. For example, if a Loop Test

Failure is detected multiple times, only one interrupt will get generated for that failure until the bit is set to zero by the user and another Loop Test failure is detected causing the bit to be set again. The user should clear the BIT status register once it has detected and processed the BIT failure.

Bit 30: Run Initiated BIT: W

This bit is set to a one by the user to direct the PE run Initiated BIT. This bit is self clearing.

Bit 31: Reset Channel: W

This bit is set to one by the user to reset the PE channel. A channel reset will force all PE registers to a power-on state. For example, POST BIT will run, Time-Tag registers will get reset to zero, etc. This bit is self clearing.

> NOTE: POST BIT performs a memory test which may cause some of the user data to be cleared. Once a channel reset performed, the contents of the PE registers and memory must be initialized again by the User.

Root PE Status Word

This word provides the user status on four key indicators: Interrupt Pending for the Channel, IRIG Signal Detection and Hardwire 1553 Function Enablers On/Off. This is a packed bit structure whose values are detailed in the following paragraphs.

Bit 0: Interrupt Pending: W

This bit is set to one by the PE to indicate that an interrupt event is pending in the channels interrupt queue. If the Root PE Control Word Bit 0, HW Interrupt On, is set to one, then a backplane/system interrupt signal has been set. This bit MUST BE cleared (set to zero) by the user to clear the backplane/system signal. Even if the HW Interrupt On bit is not on (so the user has decided against the channel being able to generate a backplane/system interrupt signal), the user should still clear this bit during an interrupt queue polling event so the bit can be utilized for the next interrupt event.

Bits 1-9: Reserved

Bit 10: IRIG Detected: R

This read only bit set to a one indicates that activity has been detected on the IRIG input signal. Once activity has been detected this bit will remain set until no activity has been detected for 5 seconds. This bit does not mean IRIG lock has occurred (see bit 11 below), it is simply an indicator that “something” has been detected on the input signal within the last 5 seconds.

Bit 11: IRIG Lock: R

This read only bit set to a one indicates that the IRIG decoder has locked on a good IRIG signal. To achieve IRIG Lock the IRIG decoder must detect at least one valid IRIG time frame. IRIG Lock will be lost if at any point the IRIG decoder detects an invalid time frame. Once IRIG Lock is lost this bit will get set to zero and the IRIG decoder will go into search mode looking for another valid IRIG time frame.

Bit 12: Reserved

Bit 13: Loop Test Failure: R

This bit indicates that the PE has detected a failure on its Loop Test. The Loop Test decodes all PE transmissions for protocol errors and bit value errors (checks all command, data and status word bit values). This bit will get set if a protocol or data bit value error has occurred. This test is run in the background when data is transmitted by the PE.

> **Note:** Loop testing is not performed on data words that have errors injected on them.

Bit 14: Bus B Jabber Detected: R

This bit set to a one indicates that a contiguous transmission of greater than 800us has been detected by the PE on bus B. When a jabber condition is detected by the PE for a given bus, the bus will be shutdown and disabled from transmitting. To enable transmission, one of the following events must occur:

1. This bit is cleared by the user to enable the bus to transmit.
2. A mode command overrides the transmitter shutdown on an alternate bus.
3. A mode command resets the RT.

Bit 15: Bus A Jabber Detected: R

This bit set to a one indicates that a contiguous transmission of greater than 800us has been detected by the PE on bus A. When a jabber condition is detected by the PE for a given bus, the bus will be shutdown and disabled from transmitting. To enable transmission, one of the following events must occur:

1. This bit is cleared by the user to enable the bus to transmit.
2. A mode command overrides the transmitter shutdown on an alternate bus.
3. A mode command resets the RT.

Bits 16-20: Bit Time Tag: R

These bits contain the 5 least significant bits of the time-tag register and are used by the PE for Time-Tag BIT testing.

Bits 21-25: Reserved

Bit 26: Flash Read Enable: R

This bit set to a one indicates that the Flash Read option is enabled for this PE. This option enables the user to program setup data into on board flash. On power-up or reset, the setup data is copied from Flash to PE memory space. For more information regarding this feature, please contact Alta.

Bit 27: Signal Capture Enabled: R

This bit set to a one indicates that an analog-to-digital (ADC) converter is present on this channel and that it is capable of capturing the bus signal.

Bits 28-31: Hard-wired Functional Status

These bits are set to one to indicate that the respective function, BC, RT, Monitor or Error injection are allowed with this card configuration. The ***AltaCore*** has the capability to have these functions hard-wired off, but most standard builds have these functions hardwired on – set to one (although the cards capabilities are also soft controlled through the capabilities register for different purchased configurations). Please contact the factory if you have a requirement have on of these functions hard-disabled (for example some airborne records cannot have any transmission capability and this function would provide a level of protection against erroneous transmissions).

Root PE Total Mon Command Counter: W: 0x000C

This register is incremented by the PE whenever a Command Word is detected. The user can write and clear/set this register as desired. This is very useful to compare Command counts for test and embedded applications. This value is reset to zero when the Monitor On/Off bit is set to on (one).

Root Total Mon Error Counter: W: 0x0010

This register is incremented by the PE whenever an error is detected for a 1553 message (only one increment per message). The user can write and clear/set this register as desired. This is very useful to look for error counts in test and embedded applications. This value is reset to zero when the Monitor On/Off bit is set to on (one).

Root BM/BC Response Timeout: W: 0x0014

This register is set by the User to define the RT Status Response Timeout for BC and BM message decoding. The value is programmable from 12-32 µsec in 100 ηsec ticks. This value is the raw dead bus time. The 1553 standard adds 2 µsecs to this time for

official definition of Response Timeout, which in the standard, is measure from mid Parity of the last word of the command message to mid sync of the RT Status Word.

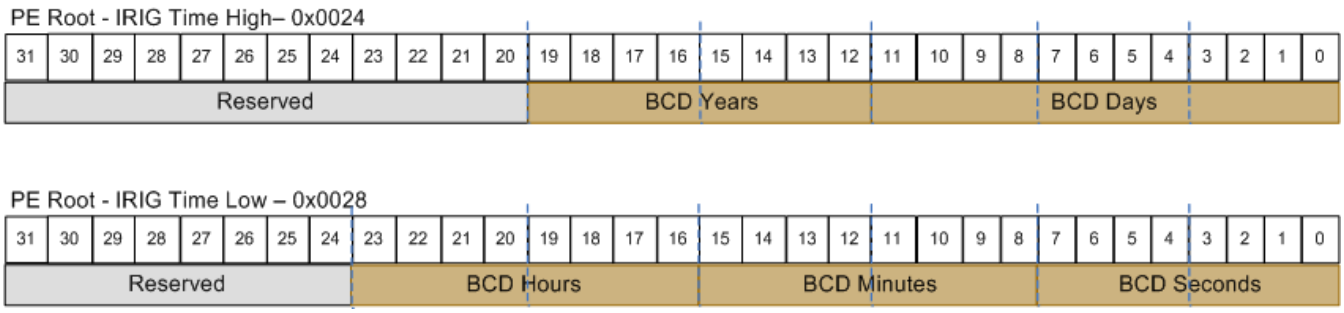
Root RT Mode (1553A/B Settings for RTs): W: 0x0018

This register is set by the user to define RTs that are the older 1553A standard protocol. The User sets a one in the bit position that corresponds to the RT address (bits 0-31 for addresses 0-31, respectively).

Root Time High & Time Low: W: 0x001C/20

These two registers make-up a 64-bit, 20 ηsec time value and provide a user window to the internal 64-bit time register of the PE (used to time stamp message). These registers are used for reading the current internal PE Time registers or these registers are set by the user to jam/set the internal PE Time registers with the held value. Root PE Control Word bits (described early in this section) control the action of these registers for read and write. The Time High register holds the 32 MSB bits and the Time Low register holds the lower 32 LSBs. The resolution is 20 ηsec/bit.

Figure Root PE-3:IRIG Time Register Format



Root IRIG Time High & Low: 0x0024/28

These two words provide a read-only user window to the PE's internal 64-bit IRIG-B Time Word. IRIG-B decoding is built-in to most ***AltaCore*** PE configurations, but the function is turned-on/off via the Global Hardware Capabilities Register. If IRIG Time is not being used, then the value in the IRIG time registers is unpredictable.

The following diagram provides the structure of these words (IRIG-B time is encoded **DAY-HOUR-MINUTE-SECOND** in BCD nibble digits).

When a valid IRIG-B one second string has been decoded, the PE will automatically latch the current PE Time High & Low, and the IRIG Time High & Low Registers into internal registers. To read the current PE Time High & Low and correlating IRIG Time High & Low Registers, the user's application must set the Read IRIG Time bit in the Root PE Control Word to force the most current values of the internal registers to be latched to the Time High & Low and the IRIG Time High & Low registers: this allows the

user's application to correlate the time between the PE's internal clock and the decoded IRIG signal. Only the PE's internal clock is used for time stamping messages.

NOTE: The IRIG times latched in the registers described above **are one second behind.** The user/API's code will need to adjust the time +1 second. Check your AltaAPI release notes for this +1 second adjust made to the IRIG read function.

NOTE: Alta IRIG decoder supports:

- IRIG-B002, Format B, DCLS, BCD TOY
- IRIG-B006, Format B, DCLS, BCD TOY, BCD YEAR

- IRIG-B122, Format B, AM, 1KHz carrier, BCD TOY
- IRIG-B126, Format B, AM, 1KHz carrier, BCD TOY, BCD YEAR

NOTE: IRIG Time Low register (0x0028) is also used to latch the **Root Interval Timer**. Please see Interval Timer description below (0x004C).

Root BIT Status: 0x002C

This word provides the status of BIT operation for the PE.

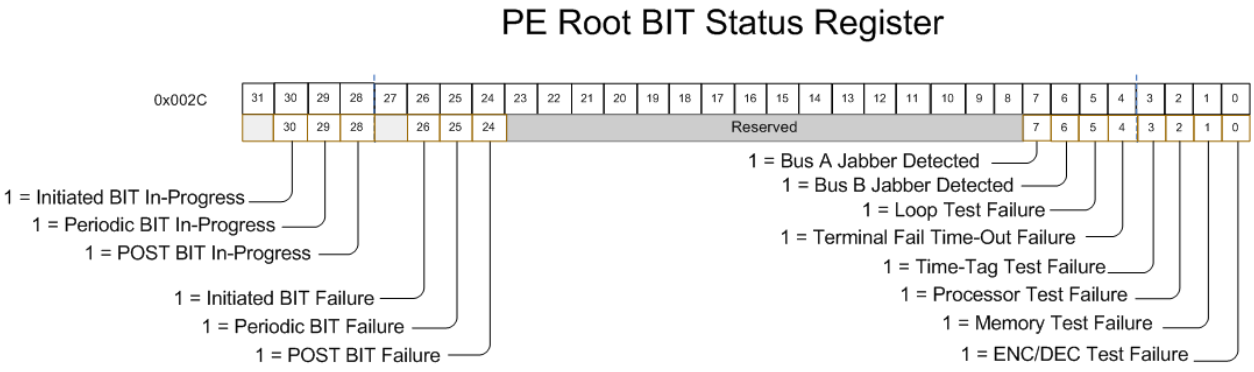


Figure Root PE-4: BIT Status Register

Bit 0: Encoder/Decoder Test Failure

This bit indicates that the PE has detected a failure on its Encoder/Decoder Wrap Test. The Encoder/Decoder Test only runs during POST BIT. When this test is in progress, the bus is operated in internal mode, i.e. data is not transmitted on the external 1553 bus.

Bit 1: Memory Test Failure

This bit indicates that the PE has detected a failure during on its Memory Test. The Memory Test only runs during POST BIT.

Bit 2: Processor Test Failure

This bit indicates that the PE has detected a failure on its Processor Test. The Processor Test runs during POST, Periodic, and Initiated BIT.

Bit 3: Time-Tag Test Failure

This bit indicates that the PE has detected a failure on its Time-Tag Test. The Time-Tag Test runs during POST, Periodic, and Initiated BIT.

Bit 4: Terminal Fail Timeout Failure

This bit indicates that the PE has detected a failure on its Terminal Fail Timeout Test. The Terminal Fail Timeout Test only runs during POST BIT.

Bit 5: Loop Test Failure

This bit indicates that the PE has detected a failure on its Loop Test. The Loop Test decodes all PE transmissions for protocol errors and bit value errors (checks all command, data and status word bit values). This bit will get set if a protocol or data bit value error has occurred. This test is run in the background when data is transmitted by the PE.

> Note: Loop testing is not performed on data words that have errors injected on them.

Bit 6: Bus B Jabber Detected

This bit indicates that a contiguous transmission of greater than 800us has been detected by the PE on bus B. This test is run in the background when data is transmitted by the PE.

Bit 7: Bus A Jabber Detected

This bit indicates that a contiguous transmission of greater than 800us has been detected by the PE on bus A. This test is run in the background when data is transmitted by the PE.

Bits 23-8: Reserved

Bit 24: POST BIT Failure

This bit indicates that the PE has detected a failure during POST BIT.

Bit 25: Periodic BIT Failure

This bit indicates that the PE has detected a failure during Periodic BIT.

Bit 26: Initiated BIT Failure

This bit indicates that the PE has detected a failure during Initiated BIT.

Bit 27: Reserved

Bit 28: POST BIT In-Progress

This bit indicates that the PE is currently running POST BIT.

Bit 29: Periodic BIT In-Progress

This bit indicates that the PE is currently running Periodic BIT.

Bit 30: Initiated BIT In-Progress

This bit indicates that the PE is currently running Initiated BIT.

Bit 31: Reserved

Signal Capture Discussion

Not all Alta 1553 boards/channels include the ADC needed for the Signal Capture Feature. See the individual board HW manual for information regarding the availability of this feature.

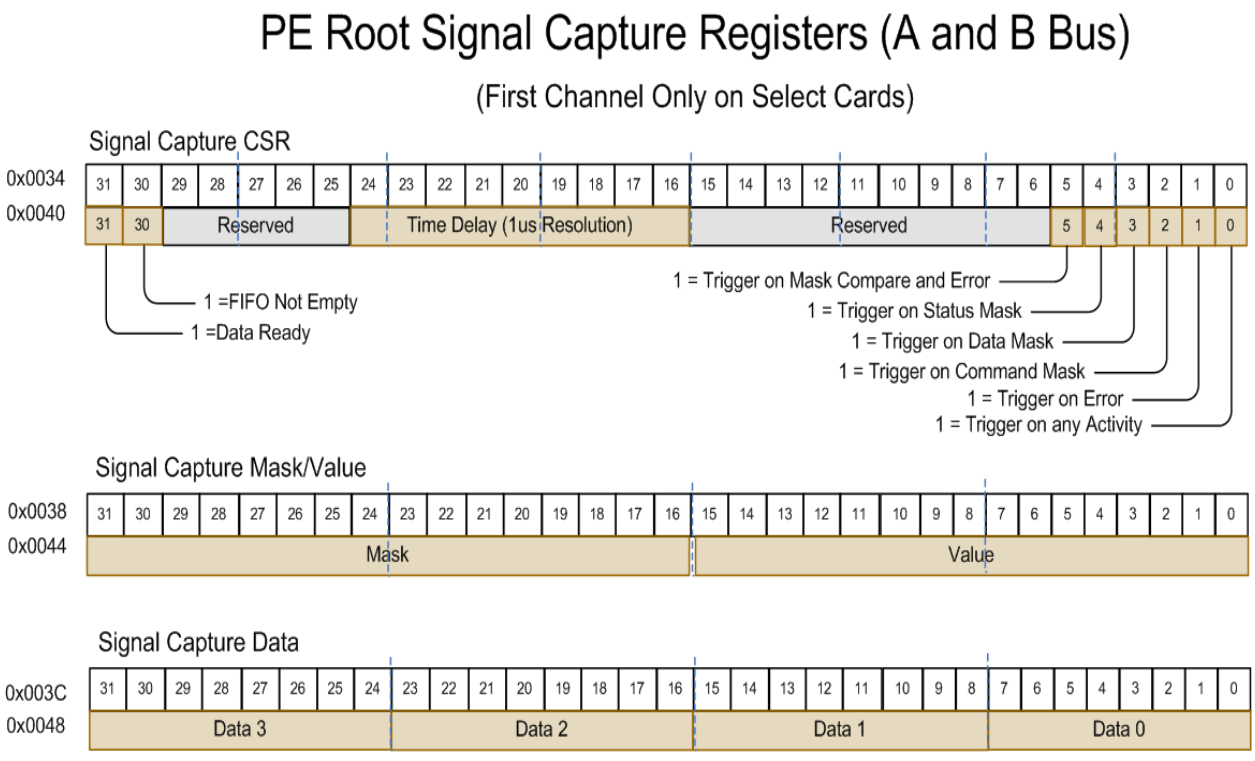


Figure Root PE-4: Signal Capture Registers

The Signal Capture Feature uses an analog-to-digital converter (ADC) to capture the electrical signal on the selected 1553 stub (A or B). The Signal Capture feature will capture 2048 samples at a rate of 20MHz, or 50 nanoseconds per sample. Therefore the sample buffer contains 102.4 microseconds of data. Each sample is an 8-bit (256

step) value representing the differential voltage on the 1553 stub. The Signal Capture data should be accurate to within 500mV, but is not calibrated. The FIFO is in words (512 words) with 4 ADC bytes per word (the byte positions are provided below – "Root Signal Data").

To convert the raw ADC data to a stub voltage representation, use the following formula:

Stub Voltage = (ADC data – 128) * 1.79 (xformer ratio) * 32 (voltage divider) * 2mv (step voltage)

Reducing the above gives:
Stub Voltage = (ADC data – 128) * 0.11456

NOTE: The Signal Capture provides simple voltage and timing data and i**s NOT a calibrated, precision instrument.** The Signal Capture Feature does NOT replace a calibrated oscilloscope for voltage or timing measurements on the 1553 stub. If more precise information is needed regarding the electrical signal on the 1553 bus, a calibrated, precision oscilloscope should be used.

The following steps should be performed to acquire Signal Capture data from the PE.

1. Set the Trigger Information Signal Capture CSR (for Bus A and/or B).
2. Wait for Data Ready bit to get set in the Signal Capture CSR
3. Read data from the Signal Capture Data Register. Note: The Data Register is 32-bit word contains four ADC byte samples.
4. Keep reading data until the FIFO Not Empty bit is set to zero by the PE.

Root Signal Capture CSR Bus A: 0x0034

The trigger bits will self clear once trigger is met. Trigger events (except for trigger on anything)are captured at the end of the 1553 word/event from which the event occurred (so the trigger is marked at the end of the word).

Bit 0: Trigger on Any Activity: W

This bit set to a one will cause the signal capture detection circuit to trigger on any activity detected on the 1553 stub.

Bit 1: Trigger on Any Error: W

This bit set to a one will cause the signal capture detection circuit to trigger on any 1553 protocol error detected on the 1553 stub.

Only One of Bits 2-4 may be set at a time. ONLY One bit 0-4 can be set at a time. Bit 5 should only be set if one of the trigger on Mask bits (2-4) are set.

Bit 2: Trigger on Command Mask (Command Word – BC): W

This bit is set to one by the user/API for the Signal Capture to trigger on the BC Command Word with the Mask/Value settings in word (0x0038/0x0044).

Bit 3: Trigger on DATA Mask: W

This bit is set to one by the user/API for the Signal Capture to trigger on the DATA Word with the Mask/Value settings in word (0x0038/0x0044).

Bit 4: Trigger on STATUS Word Mask (Status Word - RT): W

This bit is set to one by the user/API for the Signal Capture to trigger on the STATUS Word with the Mask/Value settings in word (0x0038/0x0044).

Bit 5: Trigger on Error with Mask: W

This bit is set to one by the user/API for the Signal Capture to trigger on an Error with one of bits 2-4 set with Mask/Value.

Bits 6-15: Reserved

Bits 16-24: Trigger Position: W

These bits set the word trigger position within the 512 word FIFO (4 ADC bytes per word). For example, a value of 0x100 hex would set the trigger for 50% so that values could be seen before and after the trigger event.

Bits 25-29: Reserved

Bit 30: FIFO Not Empty

This bit set to a one indicates that the Signal Capture data FIFO is not empty and that more data is available to be read by the user.

Bit 31: Data Ready: W

This bit set to a one indicates that a trigger has occurred and that Signal Capture data is available to the user.

Root Signal Data A: 0x003C

Bits 0-7: Data 0
Bits 8-15: Data 1
Bits 16-23: Data 2
Bits 24-31: Data 3

Root Signal Capture CSR Bus B: 0x0040

Bits the same as Bus A

Root Signal Data B: 0x0044

Bits the same as Bus A

Root Signal Data B: 0x0048

Bits the same as Bus A

Root Interval Timer: 0x004C

The Interval Timer provides a 24-bit, 1 µsec timer (~16 sec max) that can generate external triggers (pulse per second generator) and be synchronized to an external trigger. This can be ideal for an RS-485 or TTL external trigger output (controlled by the Root PE Control Register 0x0000) for synchronizing events (BC, BM, PB or SG) events across channels or cards. Also provided is an option to generate a hardware interrupt when the time value is reached (may be useful for applications where the user cannot have a software timer/signal to activate an event).

This register contains 6 control bits in the LSB positions and the 24 MSBs are the 1 usec clock ticks on when the timer would reset back to zero (so the timer will count up from zero and then reset when the 24-bit value is reached – this reset can cause the interrupt or external trigger). The bit definitions follow:

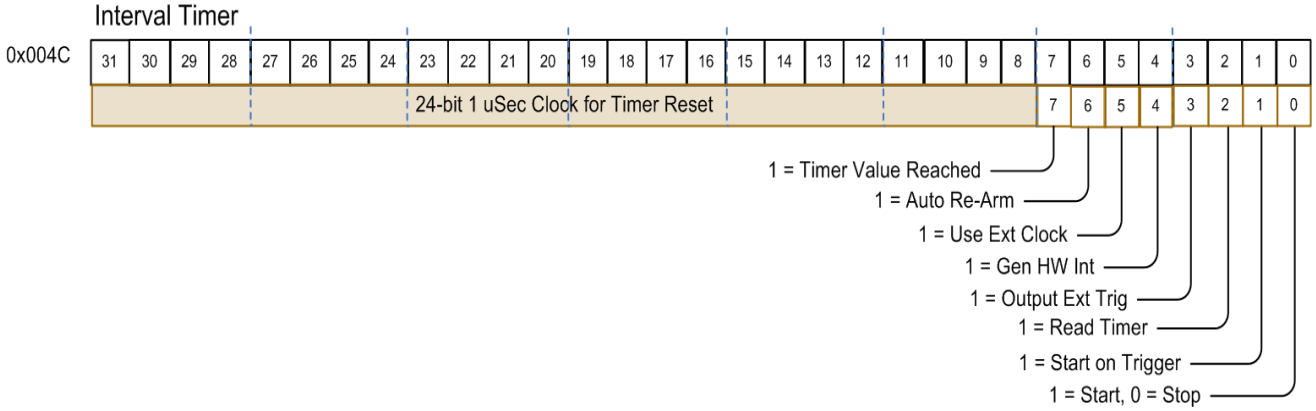


Figure Root PE-5: Interval Timer Register

Bit 0: Timer Start/Stop: W

This bit is set to one by the user/API to start the timer and set to zero by the user/API to stop/reset the timer. The user/API must stop the timer to change any other control bit values, then start again. This bit must be set to one along with Bits 1-4 & 6 options.

Bit 1: Start on Ext Trig: W

This bit is set to one by the user/API to have the timer not start until an input trigger is detected. Bit 0 must also be set to one for this action. (Also see bits 4 & 6 option for external trigger and hardware interrupt).

Bit 2: Read Timer: W

This bit is set to one by the user/API to have the PE Timer and Interval Timer latched for reading/comparison. The PE Timer values are stored in the PE Time

High/Low (0x001C, 0x0020) registers and the Interval Timer is stored in **the IRIG Time Low (0x0028) register**, and is LSB justified (bits 0-23 have the timer value). This bit is self-clearing. The PE latches the PE 64-bit timer value with each increment of the Interval Timer, so the read values in PE and IRIG time registers are from the last latched occurrence (this allows the user's program to use the difference between the two timers to track any clock drift).

Bit 3: Output Ext Trig: W

This bit is set to one by the user/API to cause an external trigger output at the timer reset rate (24-bit 1 µsec time). For TTL or RS-485 trigger selection, see Root PE Control Register 0x0000 settings. Bit 0 must also be set to one for this action.

Bit 4: Gen HW Int: W

This bit is set to one by the user/API to force the PE to generate a hardware interrupt for this channel device at the program time rate. **BE CAREFUL WITH THIS SETTTING. An interrupt time that is too fast will probably cause the computer/OS to crash!!** See Interrupt Section later in this manual for description on how this event is logged into the Interrupt Queue. (Also see bits 1& 6 option below for external trigger and hardware interrupt).

Bits 5: Use External Clock: W

This bit is set to one by the user/API to force the Interval Timer to be clocked by the external clock settings of the Global Device CSR. This allows the Interval Timer to clock user Pulse Per Second (PPS) or clock events and synchronize between the PE 64-bit clock**. Maximum input rate is one µsec (NRZL clock or PPS).**

Bits 6: Auto Re-Arm: W

This bit is set to one by the user/API to force the Interval Timer to rearm the start timer value when the user time interval is reached. This bit is often used with bits 1& 4 above to have an external trigger generate a hardware interrupt, and then rearm the timer to wait for the next trigger input. Bit 0 must also be set to one to start the action, but does not need to be reset each time. **The Time Interval Reset Value (bits 8-31) should be set to a minimum of 2.** The hardware interrupt would not occur until after the time value has expired (and within 5 uSec of the time expiration).

The Global CSR word controls the TTL or RS-485 clock input setting (bits 12 and 13). The 1, 5 or 10 MHz settings do not matter in the Root PE or Global CSR. **Example**: if Bit 13 of the Global CSR is set, then user would have the TTL

input for driving the Interval Timer. There is an example API program to show these possible settings.

The user may also want to see the BM Control Word setting to automatically have the Interval Timer copied to CDP Reserved Word 0x0030 (This option allows the user to have their own timer/clock/PPS event and then have both the PE 64-bit time stamp and user driven Interval Timer in the CDP).

> **NOTE:** When Bit 5 is set for external clock, then bits 1, 3, 4, 7 and 8-31 are ignored and will not function. Also, when Bit 5 is on, the PE CSR bit to clear the PE timer will also clear the Interval Timer.

Bit 6: RESERVED

Bits 7: Time Value Reached: W

This bit is set to one by the PE when the 24-bit time value has been reached. The user should reset this bit to zero as desired for their application.

Bits 8-31: Time Interval Reset Value: W

These 24 bits are set by the user/API to the desired 1 µsec tick value has been reached – the timer then resets back to zero and starts counting up again as long as Bit 0 is set to one. The maximum value is ~16 seconds (2^{24}/1000000). When the timer reaches maximum value, then the options in bits 3-4 can execute and Bit 7 will be set to one.

PTP IEEE-1588 Registers: 0x00C0-0x00F8

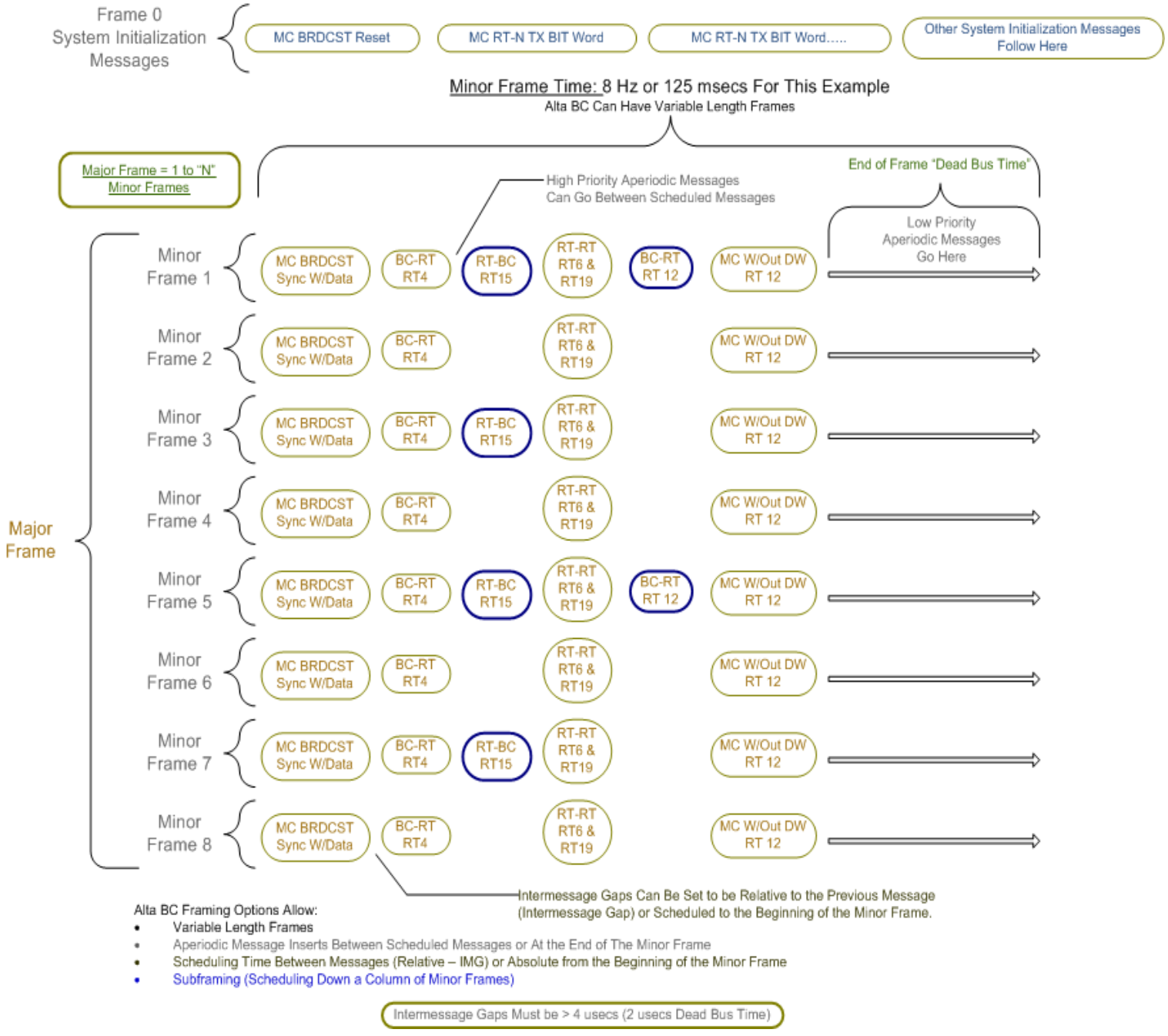
These 15 registers are for ENET/ENET2 devices for key PTP IEEE-1588 Time Sync and Control-Status (CSR) registers. ENET devices can synchronize time via 1588 grand masters in a similar manner to IRIG synchronization. If engaged (via user software API settings), this overrides IRIG sync decoding.

For details on implementation, please contact Alta for access to the PTP IEEE-1588 Design Implementation Application Note and reference the appropriate sample C programs. Contact Alta at alta.support@altadt.com or 888-429-1553 x2.

<~~~>

BC Data Structures

This section reviews the data structures for programming BC, Playback and Signal Vector transmission. Although these functions are fairly distinct in operations, all involve acting as the BC on the 1553 channel, and thus, are lumped together in this section. Only one of these functions can be started as the transmitting BC at a time. BC programming is reviewed first – please see other paragraphs in this section for playback and signal vector review. The following figure shows a high level example of a



BC major frame and sample ***AltaCore*** PE capabilities.

BC Basics

1553 BC message transmission is controlled through a series or Root PE and Root BC registers, and a key linked-list data structure known as BC Control Blocks (BCCB). Through these Root registers and BCCBs, the user has a wealth of options for transmission time framing, 1553 message selections, frequency subframing and intelligent processing options to offload host processing requirements. The previous diagram provides an example of BC programming options (there are many more!).

Before the user can execute a BC list, one or more BCCBs must be defined and loaded to onboard memory for PE processing. BCCBs not only select 1553 messages for transmission, but can also provide advanced frame timing and logic tree decision making (based on 1553 word values, time values and other system initiated settings like triggers) for BC execution. Once the user has defined all desired BCCBs for their BC logic/timing tree, the Root BC registers allow the user to start/stop/control/monitor BCCB transmission, execute on-demand or "aperiodic" BCCB messages and control memory banking (which allows paging of data for transmission). This section will start discussion with the Root BC registers and then detail BCCBs.

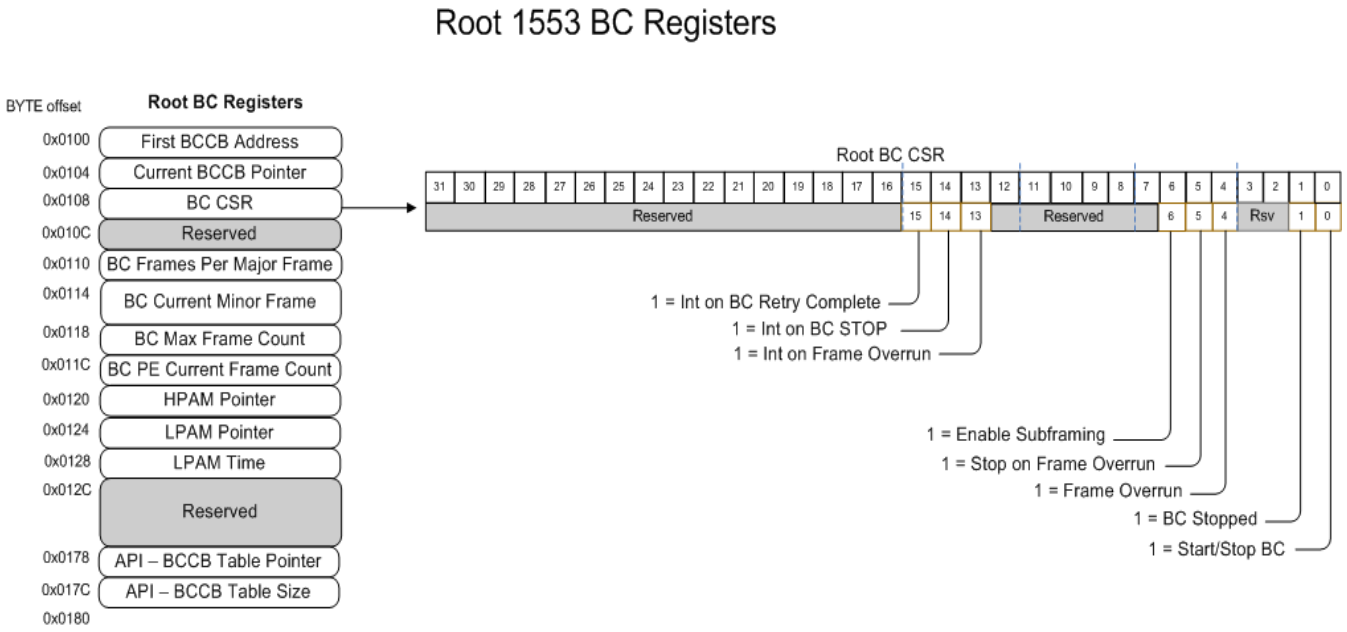


Figure BC-1: Root BC Registers and Root BC CSR

Root First BCCB Address: W: 0x0100

This word is set by the User with the address of the first BCCB to be executed when Bit 0 of the Root BC CSR is set to one (1) by the user. This word is read by the PE only when Bit 0 of the Root BC CSR has been set to one (and the BC is currently halted).

Root Current Transmission BC Control Block Pointer: W: 0x0104

This word is set by the PE whenever the next BCCB is sequenced for execution. The User can read this word to see where the BC transmission sequence is currently located. If the BC is stopped by User, this register will contain the address of the last BCCB. When the PE halts transmission (due to end of chain – null pointer in the last BCCB Head Pointer, frame overflow option, error halt option or Root BC CSR Start/Stop Bit being cleared to zero), then this value will be set to 0x00000000. Bit 1 of the Root BC CSR (described next) is set to one when ALL BC processing has completed – this bit should be read by the user to clearly see when the PE has finished BC processing.

Root BC Control and Status Register: W: 0x0108

A Root BC CSR word controls global BC functions such as start/stop, retry options, interrupts options, error detection flags and framing controls. The following paragraphs describe the control/status bits.

Bit 0: Start/Stop BC: W

1=Start BC. This Bit is set by the User to start and stop the BC transmission. When this bit is set to 1, the PE will start BC processing with the BCCB pointed to by the First BCCB Address at location 0x00000100. The User clearing this bit to zero will cause the BC to halt at the end of the current BCCB (including any message time value). Other bit options in this Root BC CSR and BCCB CSR will direct BC transmission personality (wait for external or software triggers, etc...).

> **NOTE:** BCCB transmission will start within 2 µsecs of this bit being set. When turning the BC off, the user must allow 10 µsecs after Bit 1 (BC Stopped Bit described next) has been set to one before turning the BC back on again (this extra time ensures the PE is ready for BC transmission).

Bit 1: BC Stopped

1=BC Stopped. This bit is set to one by the PE when BC transmission has stopped for any reason and has completed all BC execution overhead. The User should clear this bit to zero before the start/restart of the BCCB.

Bits 2-3: Reserved

Bit 4: Frame Overrun: W

1=Frame Overrun. This bit is set to one by the PE and indicates a Frame Time Overrun as occurred. The PE will compare the actual frame time to the user defined frame time and shall set this bit if the actual time is greater. The user should clear this bit prior to starting the BC.

Bit 5: Stop of Frame Overrun: W

1=Stop on Frame Overrun. This bit is set to one by the User to direct the PE to stop BC transmission on a Frame Overrun. The PE compares frame time at the end of each BCCB to control this function.

Bit 6: Enable Subframing: W

1=Enable Subframing. This bit is set to one by the User and directs the PE to check frame counts of the BCCB for message subframing (transmitting of a message at sub intervals of the minor frame). If this bit is set, the user MUST designate the starting, ending (stop) frame and the frame increment for EVERY BCCB message (the Start/Increment/STOP values are explained in the BCCB section.

> **NOTE:** If designated with this bit, subframing will be checked for all BCCBs EXCEPT Low Priority and High Priority Aperiodic Messages (LPAM and HPAM).

> **NOTE:** Subframing cannot be used with branching. However, the user can create subframe execution logic with managing the BCCB Head Pointer (and a single common CDP can be pointed to by multiple BCCBs). Alta API example programs show how to accomplish this BCCB and CDP pointer management.

Bit 7-12: Reserved

Bit 13: Interrupt of Frame Overrun: W

1=Interrupt on Frame Overrun. This bit is set to one by the User to direct the PE to generate an interrupt when BCCB transmission has exceeded the User programmed frame time. Bit 5 above controls if the PE will halt BC transmission on a frame overrun condition.

Bit 14: Interrupt of BC Stopped: W

1= Interrupt on BC Stopped. This bit is set to one by the User to direct the PE to generate an interrupt whenever the PE has stopped BC transmission (for any reason).

Bit 15: Interrupt of Retry Complete: W

1= Interrupt on Retry Complete. This bit is set to one by the User to direct the PE to generate an interrupt when the PE has completed a Retry Sequence.

Bits 16-31: Reserved

Root BC Frames Per Major Frame: W: 0x0110

This word is set by the User to inform the PE the total number of frames in a major frame. The subsequent Root Current Minor Frame Count Register is incremented every time a frame is completed (a frame ends when bit 7 of the last BCCB of the frame is set to one and any Retries or LPAM message have completed). When the Root Current Minor Frame Count value equals this register, then the PE will reset the Root Current Minor Frame Counter to one. This sequence is used by the PE to determine when designated BCCB subframe messages are to be transmitted.

> **NOTE:** The User MUST NEVER write to this register once BC processing has begun.

Root BC PE Current Minor Frame Count : 0x0114

This word is set by the PE to count the latest completed frame. The User can read this value to see the latest minor frame (not current) that was completed. The PE will reset this value to one when this register equals Root BC Frames Per Major Frame (described in previous paragraph). This register is used to compare against BCCB Start/Increment/Stop word values to determine when subframed messages are transmitted (when the values match, then the message will be transmitted).

This value should be initialized (generally to a 1) by the API.

Root BC Max Frame Count : W: 0x0118

This word is set by the user to direct the PE to stop executing BC frames/message when “N” frames have been completed. A value of zero will indicate transmit forever (or until a programmed BC stop condition, such as a BCCB null Head Pointer or Stop on Error, has been met). When the subsequent register, Root BC PE Current Total Frame Count value equals this value, then the PE will halt BC transmission.

> **NOTE:** The User MUST NEVER write to this register once BC processing has begun.

Root BC PE Current Total Frame Count: 0x011C

This word is set by the PE to count the latest completed frame. The User can read this value to see the latest frame (not current) that was completed. This value is used for comparing to the previous described register, Root BC Max Frame Count. This register will roll over after 32-bit unsigned. This value should be initialized (generally to a 1) by the API.

Root High Priority Aperiodic Message: W: 0x0120

This register is set by the User and is the pointer to a BC chain that will executed at the end of the current BCCB block being completely executed. The PE checks this register

for a non zero value when the current BCCB is done executing, and if non zero, then this value is used as the pointer to the BCCB chain to be executed. HPAM or LPAM messages may add 10-15 µsecs of gap time per instance.

> **NOTE:** The last message of an HPAM chain must have a NULL (zero) BCCB Head Pointer – the PE looks for a NULL pointer to return to the original BC chain.
>
> **NOTE:** High Priority chains CANNOT be nested in an LPAM chain. If subframing is implemented, LPAM and HPAM messages will be execute with respect to the subframe logic (start, stop and rep count of the BCCB message). If you always want your LPAM/HPAM chain to execute with subframing, program the start to the first frame, stop to the last frame value and rep value to one. LPAM and HPAM cannot have embedded framing or branching, but retries can be active.
>
> **NOTE:** The user must be careful to avoid minor frame time overruns – it is the user's responsibility to program the chains correctly for minor frame timing.
>
> **NOTE:** When the PE has started a High Priority chain, the PE will write a value of 0XFFFFFFFF to this register. When the HPAM chain is finished, the PE will write a value of 0x00000000 to this register.
>
> **NOTE:** HPAMs are checked for execution prior to every new BCCB message, including start of frames. However, a BCCB that is currently executing will complete its' complete sequence (including retries, branches, branch returns, etc…).

Root Low Priority Aperiodic Message (LPAM) Pointer: W: 0x0124

This word is set by the User and is a pointer to a BCCB linked-list chain that is to be executed at the end of a minor frame with enough time left to execute the chain (See note below for zero value in time register). The User must set the time of the chain in the next word (0x0128). At the end of a frame (designated by the user by setting the end of frame designator bit in the last BCCB CSR), the PE will check this register for a non-zero value, and if the PE determines that there is enough time remaining in the minor frame (by subtracting the value in Root LPAM Time register from the PE's internal frame count-down timer and looking for a great than value), the PE will execute the BC chain pointed to by this register value. The LPAM BCCB chain can be on infinite length, but the User must provide the accurate time for the chain, including possible retries, to ensure proper operations (a frame overrun condition can occur if the LPAM starts and then exceeds the frame time).

NOTE: The last message of an HPAM chain must have a NULL (zero) BCCB Head Pointer – the PE looks for a NULL pointer to return to the original BC chain.

NOTE: When the PE has started an LPAM chain, the PE will write a value of 0XFFFFFFFF to this register. When the LPAM chain is finished, the PE will write a value of 0x00000000 to this register.

NOTE: High Priority chains CANNOT be nested in an LPAM chain. If subframing is implemented, LPAM and HPAM messages will be execute with respect to the subframe logic (start, stop and rep count of the BCCB message). If you always want your LPAM/HPAM chain to execute with subframing, program the start to the first frame, stop to the last frame value and rep value to one. LPAM and HPAM cannot have embedded framing or branching, but retries can be active.

NOTE: A time of zero in the Root LPAM Time register will direct the PE to execute the chain no matter what the time length (the PE will transmit this chain at the end of the current minor frame regardless of time permitting in the frame).

NOTE: At the end of the minor frame (designated by the user by setting the end of frame bit in the last BCCB CSR), the LPAM is checked AFTER the High Priority Message option.

Root LPAM Time: W: 0x0128

This register is set by the User and is the time for the BC chain pointed to by LPAM 0x00000246. The time is 32-bit, 100 ηsec, so the chain can be no longer than this max time (6+ minutes). For this value, the user must include total bus time of transmission including possible BC Retries and RT response/transmission time.

NOTE: The user must add an extra 30 µsecs to the actual LPAM transmission time to ensure proper PE execution (this allows for possible RT Response Time-Out of the last LPAM message and PE overhead processing).

API - BCCB Table Pointer: W: 0x0178

The API allows the user to reference messages (BC Control Blocks) by a message number rather than requiring the user to work directly with memory offsets on the board for the BCCBs. Therefore, the API creates a table in board memory where it stores the memory offset to the BCCB for each message number. This register is used to store the pointer to the start of this table. By placing this information in board memory (rather than keeping it in a software variable in the API), this enables the user to find the BC Control Blocks when inspecting board memory to troubleshoot problems. The user can

look at a memory dump to verify that the data structures are setup correctly and can follow this pointer to find all BC Control Blocks that the API has allocated on the board.

API - BCCB Table Size: W: 0x017C

This register is used with the previous register to define the BCCB table used by the API to manage BC messages. This register provides the size (in 32-bit words) of the BCCB table. This allows the API to check for invalid message numbers (a message number that exceeds the BCCB table size is invalid).

BC Control Blocks (BCCB) – Extended Memory

BCCBs are data structures that direct PE BC transmission formats and frame timing. Definitions of BC messages (and optional processing actions) are encoded in the BCCB data structures that are linked-list together to form a one-way chain of execution. The user has unprecedented options for processing, including: variable length framing, frequency subframing, indefinite conditional branching (for detailed IF/THEN/ELSE logic) and memory pages for BC data word transmissions. The following paragraphs describe the BCCB data structure and these options.

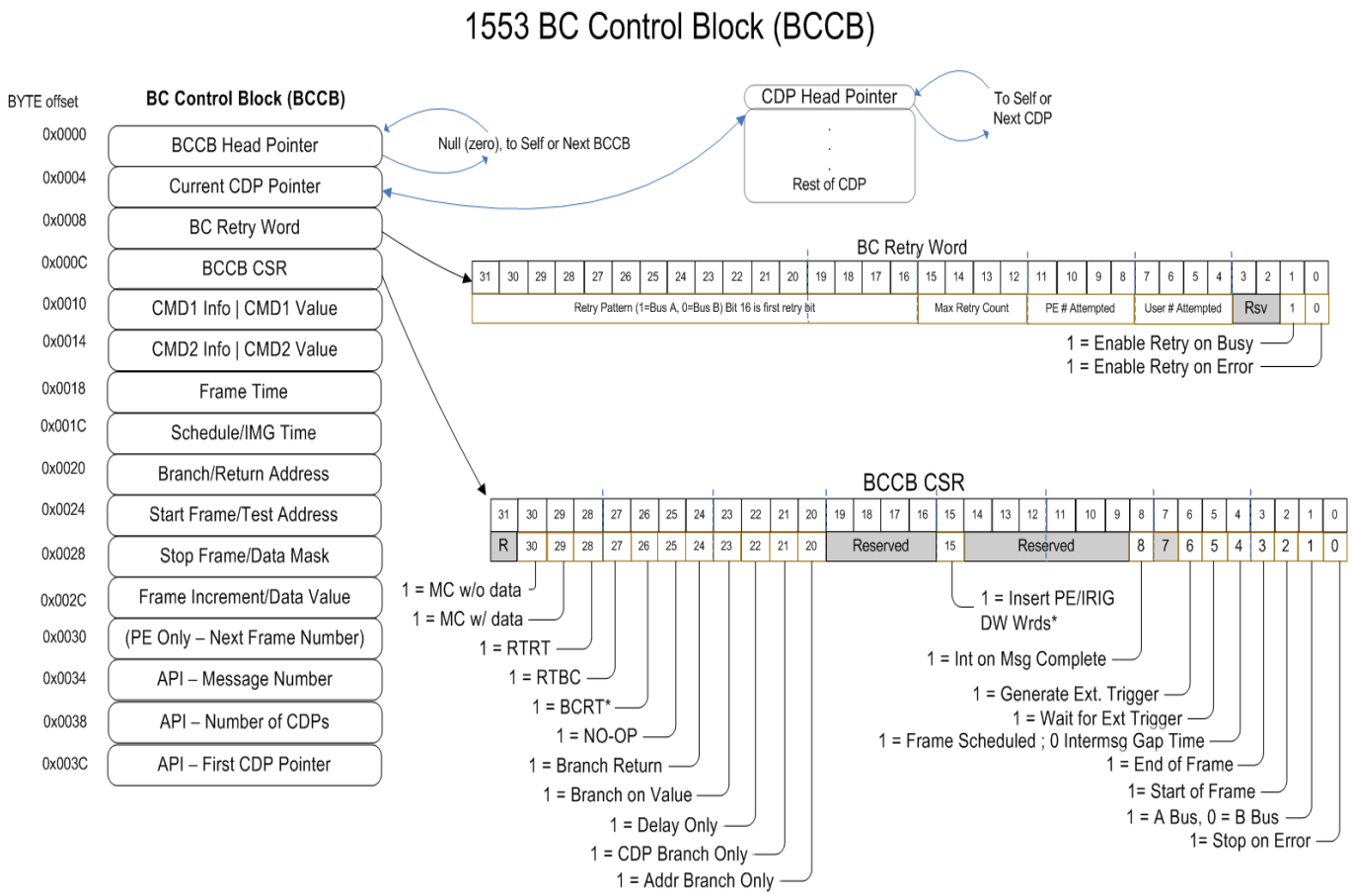


Figure BC-2: BC Control Block

BCCB Head Pointer: W: 0x0000

The first word the BCCB is a pointer to the next BCCB (head pointer of BCCB link-list). If a loop of execution is desired, then the last BCCB head pointer would point back to the desired location for repetition.

> **NOTE:** The PE provides the Current Transmission BC Control Block Pointer so the User can monitor the current location of BC execution. The User can change this value on-the-fly/real-time (to insert or delete a link of the chain in real-time), but a change must occur within 10 µsecs of the end of the current message WHILE the current message is executing. Remember, this time does not include host processing or backplane delay time – the user must take great care in changing this value on the fly.

Current Common Data Packet (CDP) Pointer: W: 0x0004

This word is the address to the current BCCB's CDP (head pointer of CDP link-list). This value will assume the next address of the CDP linked list when the current BCCB has completed processing. See CDP section for details.

> **NOTE:** The PE provides the Current CDP so the user can monitor the current location of BC execution. The User can change this value on-the-fly/real-time (to insert or delete a link of the chain), but a change must occur at least 10 µsecs before PE execution of this change.

BCCB Retry Word: W: 0x0008

16 Bit Retry Pattern | 8 Bit Reserve | 4 Bit Retry Count | 4 Bit Last Count Attempted by PE: W: 0x0008

This word is a packed bit data structure that controls retries for the respective BCCB. When a retry condition has occurred, then the PE will re-transmit the respective BCCB Command Message as directed by bit settings in this work. Retries may add up to 10-15 µsecs of gap time per instance. The user should NOT write to this register during a retry sequence as this will affect PE processing/tracking bits.

Bit 0: Enable Retry on Error: W

1=Enable Retry on Error: This bit is set to one by the User to direct the PE to retry (re-transmit) the BCCB command message per the BCCB Register Retry Settings (see next paragraphs) when a protocol error (invalid response) has been detected. This is not the case for illegal messages.

Bit 1: Enable BC Retry on Busy Bit: W

1=Enable Retry on Busy Bit. This bit is set to one by the User to direct the PE to retry (re-transmit) the BCCB command message per the BCCB Register Retry

Settings (see later in this section) when the busy bit in the RT response Status Word is set to one.

Bits 2-7: Reserved

Bits 4-7: Retry # Attempted: W

These bits are set by the PE to inform the User of the number (#) of retries attempted for the last retry instance. Once read, the User should clear these bits when the respective BCCB is not executing.

Bits 8-11: PE Retry # Attempted

These bits are set by the PE to track the number (#) of retries attempted for the last retry instance. The user can read these bits, but should not write or change the values (so do not write to this register in the middle of a retry instance). These bits are for internal PE processing only.

Bits 12-15: Retry Count: W

These bits are set by the User to direct the PE to perform 1-16 of retries based on the 0 and 1 bit settings (Enable BC Retry on Error or Busy). A value of zero will cause no retries to be attempted.

Bits 16-31: Retry Pattern: W

These bits are set by the User to direct the PE as to the retry bus pattern that should be executed. Bit 16 is the LSB and Bit 31 is the MSB. Each bit position corresponds to the Retry Count (for example bit offset 3 will correspond to retry attempt 3) and a value of 1 directs the PE to transmit on the “A” bus from the previous transmission and a value of 0 directs the PE to transmit on the “B” bus as the previous transmission.

NOTE: Remember, that if you switch the bus selection in the BC CSR, and you are using retries, then you probably want to invert or switch the Retry Pattern to match your desired pattern.

NOTE: The IMG for retries is a minimum of 40 uSec and maximum of what is programmed in the Delay Time register of the BCCB. In some cases with large IGM Delay Times, this may cause large duplicate IMGs. This can be alleviated by adding “Delay Only” BCCB in front of the 1553 message BCCB; the Delay Only can hold the desired IMG/Frame Schedule for the 1553 message, and the 1553 message BCCB can hold the retry IMG value.

BCCB Control and Status Register: W: 0x000C

This word is a packed bit data structure that directs execution of the BC message described by this BCCB. The bits are defined as follows.

Bit 0: Stop On Error: W

1=Stop on Error. This bit is set to one by the user to direct the PE to stop transmission after the current BCCB message if a protocol error is detected. If this bit is set to zero, then the PE will continue BCCB transmission sequencing (including retries, branching, etc..) regardless of error conditions.

Bit 1: Bus Indicator: W

1=A Bus| 0=B Bus. This bit is set by the User to designate which bus the BC Command message will be transmitted on. A value of one will direct the PE to transmit on the A bus and a value of zero will direct the PE to transmit on the B bus.

Bit 2: Start Frame Designator: W

1=Frame Designator: This bit designates the BCCB to mark the beginning of a minor frame, which causes the minor frame count down clock value to be reloaded and message gap timers to be zeroed. The Frame Time value represents the Total Frame Time until the bit is detected in a BCCB CSR (somewhere down BCCB chain). This bit designation is required at the beginning of each minor frame. . A value of less than the actual frame time will cause a frame overrun condition (causing the frame overrun bit to be set in the Root BC CSR – and maybe an interrupt if selected by the user). Actual frame transmission time may vary depending on the number errors, retires, branches, etc. that the user programs (or conditions that are detected on the 1553 bus). The user must include all possible retry and branching time when calculating minor frame times.

> **NOTE:** The user must have a minimum of 10 µsecs of dead bus time at the end of a minor frame to ensure the accurate start (within 500 ηsecs) of the subsequent frame.

> **NOTE:** Once the BCCB execution starts to use minor framing (by the PE processing the first BCCB in a chain with the Start Frame Designator bit set), then the user may not go back to straight, non-framing BCCB chains. The user would be required to stop the BC and restart the BC to intermix framing/non framing transmission types.

Bit 3: End of Frame Designator: W

1=End of Frame Designator. This bit is set to one by the User to designate the end of a timed minor frame. This bit is mandatory for frame time operations and usage of LPAM.

Bit 4: Message Gap Type: W

1=Frame Schedule | 0=Intermessage Gap (Dead Bus). This bit tells the PE what type of time gap to execute prior to transmitting this BCCB's Command message.

A one value will direct the PE to delay transmission until a time offset from the beginning of the current minor frame has expired (the time value programmed in BCCB offset 0x001C is greater than the time from the beginning of the current frame – this called "Frame Scheduling").

A zero value will direct the PE to insert an Intermessage Dead Bus time gap (IMG). The time value for either choice is set by the user in the Message Gap word (offset 0x001C) of this BCCB (see below). The 1553 standard requires a minimum gap of 4 µsecs (which is 2 µsecs for Message Gap word programming – see note below).

> **NOTE**: If the User wants a standard 1553 IMG for all messages, then this bit must programmed to zero. The PE works with actual DEAD BUS TIME, not 1553 intermessage gap times (IMGs). 1553 IMGs are measured from mid-bit parity to mid-bit sync (2 µsecs of extra time). The PE gap times are true dead bus (idle) time for encoded transmission.
>
> For example: The minimum 1553 IMG is 4 µsecs per the specification; this would be 2 µsecs for ***AltaCore*** (PE) Message Gap Time. A simple subtraction of 2 µsecs for standard 1553 IMGs will provide the PE time.
>
> **NOTE**: DO NOT set the Frame Schedule value for the first message of a frame (set this bit to 0 for first message of the frame, then you can use/intermix IMGs and Frame Scheduling in the rest of the frame). The first message will not have a gap. If you want a gap for the first message, then use a "Delay Only" BCCB to insert a dummy message (with this bit set to 0) at the beginning of the frame; then set the Frame Schedule or IMG time in the Second Message to insert the desired gap from the beginning of the frame.

Bit 5: Wait for External Trigger: W

1=Wait for External Trigger: This bit is set to one by the User when the BCCB is not to execute until a hardware trigger line (latched from a high to low signal detected by the PE) or the Force External Trigger bit in the PE Control Word (offset 0x00000000 for the channel device). The Force Trigger bit can be used to synchronize all BCs channels on a card to start at the same time (within 5 µsecs of each other). The Force Trigger Bit is ORed with the External Trigger Line.

This is often used with the Generate HW Trigger bit above to chain multiple channel's/card's BC execution. Start accuracy shall be within 5 µsecs of trigger line going high to low (usually accuracy is within 1 usec if there is at least 10 usec of dead bus time before this BCCB message)..

On most cards the user can select LVTTL (default) or RS-485 level trigger (see Root PE Control Register Bit 16). See I/O Requirements for Card level details on pin-outs and electrical characteristics.

NOTE for Intermessage and Frame Schedule : Setting this bit to a one and setting the Schedule/IMG word (BCCB offset 0x001C) to a non-zero value will cause the message to go out at a time (specified by the Schedule/IMG word) after the trigger has occurs (does not apply to first BCCB message word of a frame).

NOTE: Frame Timing: This trigger bit overrides Frame Timing: If this bit is set and the trigger occurs, the message will be transmitted regardless of the frame time. (Often this bit is used to start/sync a minor frame and this will override frame timing). The user may still want to use Frame Time and Framing to clock the BCCB Message for Frame Scheduling and Subframing and Low Aperiodic messages. When using Ext Trigger with Frame Timing, set the Frame Time to (FrameTIme – 20 µsec: example a 20 msec frame time would be set to 199800 in 100 ηsec ticks) – this time allows for different clock time accuracy/phase between an external source and the internal PE clock.

Note: Triggers are not latched. If a trigger occurs prior to a BCCB (with wait for trigger) being ready to transmit, the message will not go out until another trigger occurs.

Bit 6: Generate External Trigger: W

1=Generate External Trigger. This bit is set to one by the User to direct the PE to toggle the External Trigger line. This is a high-to-low value for 1 µsec. On most cards the user can select LVTTL (default) or RS-485 level trigger (see Root PE Control Register Bit 17). See I/O Requirements for Card level details on pin-outs and electrical characteristics.

Bit 7: Reserved

Bit 8: Interrupt on Message Complete: W

1=Interrupt on Message Complete. This bit is set to one by the User to direct the PE to generate an interrupt when the message is complete.

Bits 9-14: Reserved

Bit 15: Insert PE/IRIG in Data Words: W

1=Insert PE/IRIG in Data Words. This bit is set to one by the User to direct the PE to automatically insert the two (32-bit) words of Latched PE Time, two (32-bit) words of Latched IRIG Time and the Current two (32-bit) Words of PE Timing in the first 12 1553 data words (16-bit) of this BC-RT message. These 12 words will be transmitted regardless of CDP content (for the first 12 Data Words) set by the user.

The Latched PE and IRIG time format is the same as described in the Root PE and IRIG Time Registers 0x001C to 0x0028. The user will need to setup and calibrate IRIG prior to this action. Please see the following PDF bookmark for format and time latching/sync details:

Root IRIG Time High & Low: 0x0024/28

The Current PE Time High/Low is the 64-bit, 20 nSec Time (same as CDP Time High and Low) of when Command Word was transmitted for this message (time value is at mid-parity of the Command Word). The user can take the delta time between the latched PE and IRIG time, and apply it to the Current PE Time to calculate IRIG time/clock drifts. There are example C programs in the AltaAPI that show this action.

This action is for ONLY BC-RT Messages AND the BCRT Bit 26 of the BCCB CSR must also be set to one, and the BCRT Command Word Count (set in the BCCB 0x0010 CMD1 Value) must be at least 12.

The format of the 12 BC-RT Data Words are:

Data Word 1 = Latched PE Time High 16 MSB
Data Word 2 = Latched PE Time High 16 LSB
Data Word 3 = Latched PE Time Low 16 MSB
Data Word 4 = Latched PE Time Low 16 LSB
Data Word 5 = Latched IRIG Time High 16 MSB
Data Word 6 = Latched IRIG Time High 16 LSB
Data Word 7 = Latched IRIG Time Low 16 MSB
Data Word 8 = Latched IRIG Time Low 16 LSB
Data Word 9 = Current PE Time High 16 MSB
Data Word 10 = Current PE Time High 16 LSB
Data Word 11 = Current PE Time Low 16 MSB
Data Word 12 = Current PE Time Low 16 LSB

Bits 16-19: Reserved

NOTE: With the next BCCB type bits 20-30, only one bit should be set.

Bit 20: Address Branch: W

1=Address Branch. This bit is set to one by the user/API to have the PE test a data mask/value pointed to by Test Address register (x0024 of BCCB) and jump BCCB execution to Branch Address (0x0020) if the test is results in a non zero value . The registers 0x0020->0x002C are used for this functions addresses and data and mask values instead of Branch Return (bit 24) or Subframing.

With multiple Address Branches, the user can build full IF-ELSE IF logic chains. The AltaAPI provides setup short-cut and example programs to demonstrate this usage.

The pseudo C logic for the compare function is shown below:

```
Temp = &Test_Address              // (0x0024)
If ((Temp & Test_Mask) == Test_Value) Then
        Goto Branch Address       // (0x0020)
Else
        Goto BCCB Head Pointer    // (0x0000)
```

This branch logic differs from other BCCB branching options where this test/jump action can work on any PE memory location (where the other BCCB branches work only on the latest BCCB message execution). This means the user can setup tests on any previous executed BCCB/CDP or other device memory values. One example of this usage would be to have a series of Address Branch checks at the end of a minor frame to test previously minor frame BC transmissions.

This BCCB has no 1553 transmission action – only a test and branch logic. Execution of this BCCB may add 5-10 uSec per instance. This branch option could be used instead of other BCCB branch options and the user would have a single branch BCCB type that can perform the same function of other (older) BCCB branch options.

NOTE: Only the BCCB CSR Message Complete Interrupt, Start and End of Frame Bits are active for this option. Branching may include Framing, but the user must end the current frame being executed prior to providing another start frame. Branching should NOT be used with subframing (see note in Subframing).

Bit 21: CDP Branch Only: W

1=CDP Branch Only. This bit is set to one by the user/API to have the PE check the last active CDP (Source) for a compare/branch action. This Branch differs to

Bit 23 Branch as this bit as NO 1553 transmission actions, and is setup differently to specify compare/mask/branch conditions. This bit option allows full IF->ELSE IF->ELSE logic to be built from last 1553 BCCB transmission.

First a review: When a BCCB transmits a 1553 message, the address of the CDP is stored in a working temporary register (not available to the user), This CDP Branch Only option takes advantage of this temporary register to make comparisons for 1-N branching decisions in the BCCB chain. So you can have multiple CDP Branch Only BCCBs chained DIRECTLY after a transmitted 1553 BCCB to make value comparisons on a word of the Source CDP.

The BCCB with this bit set will point to a CONTROL CDP(s) link list where only the Control Word, Mask and Mask Compare Registers are used for setup and only the CDP Status Word Bit Compare True bit is used for status feedback. All other values of the CDP are not used or updated in the Control CDP. The mask and compare function will be performed on the SOURCE CDP pointed to by the reserve, temporary CDP pointer register that was updated from the last BCCB 1553 message transmitted. You can chain multiple CDP Branch Only BCCBs together to work on the last transmitted/executed BCCB 1553 message.

The pseudo logic for the compare function is shown below:

```
If ((Source CDP Data & Current Control CDP MASK Value) == Current
Control CDP Mask Compare Value) Then
        Goto BCCB Branch Address
Else
        Goto BCCB Head Pointer
```

So you program this BCCB's CDP for Control Settings (word offset, Mask Value and Mask Compare Value in the Control CDP of this BCCB), but the comparison is done on the Source CDP's (from the last active 1553 transmission) word offset (set in the Control CDP's Control Word). The Mask value is used as an AND function against the reference Source CDP word offset and the result is compared against the Mask Compare Value for positive result to cause the branch to the BCCB referenced in BCCB offset 0x0020

NOTE: Only the BCCB CSR Interrupt, Start and End of Frame Bits are active for this option. Only the CDP Interrupt Without Error bit is active for the Control CDP. No Control Bits are active for the SOURCE CDP. Only the Compare Equals True bit is active for Control CDP Status Word. This option does not change or execute any of the Source CDP Control or Status Word bits.

Bit 22: Delay Only: W

1=Delay Only. This bit is set to one by the user/API to have the PE only execute the IMG/Frame Schedule time and not to execute/process any 1553 bus activity Other BCCB CSR bits (0-8) settings for interrupt, triggers and framing/subframing are executed. Branch Return/ Jump and NOOP bits (24-25) will be checked and executed prior to this bit (overrides this bit setting). This bit is checked and overrides 1553 message type bits (26-30) and Branch (23).

Bit 23: Branch: W

1=Branch. This bit directs the PE to perform a MASK/Value compare function in the current BCCB's CDP (after the current BCCB 1553 message/CDP is executed), and if a positive value is the result, then the PE will branch to the address programmed in BCCB offset 0x0020. If a non-zero value is found on the Mask/Value, then an execute branch will occur with BCCB. Multiple Branch BCCBs can be chained together. This could be helpful for building IF->ELSE logic for checking RT responses (the user must build the branch/return logic tree as the PE does not "stack" (push-pop) any addresses. The Mask values in the respective CDP must be programmed (AND Mask value, Mask Compare values and CDP word offset in the CDP CSR).

The pseudo logic for the compare function is shown below:

```
If ((Data & Current CDP MASK Value) == Current CDP Mask Compare
Value) Then
        Goto BCCB Branch Address
Else
        Goto BCCB Head Pointer
```

It may be better to use multiple CDP Branch Only (bit 21) or Address Branch (bit 20) BCCBs to make full IF->ELSE IF->ELSE logic flow from a single BCCB 1553 command transmission. Please see definitions for bit 20 & 21 above. The difference between this branch and bit 21 branch is that this bit 23 works off only the current BCCB's transmission and bit 21 CDP Branch Compare can work off the last transmission and can have multiple compares performed from the last transmission (so you could check for multiple values of a data word of a transmission). Bit 20 Address Branch can work off any device memory address. For a single transmission compare, it would be the same to use this bit 23 or a single bit 21 CDP Branch Compare. If you need multiple compares against a transmission instance, then use bit 20 or 21.

NOTE: This bit can only be set in conjunction with a 1553 type message (BC-RT, RT-BC, RT-RT, Mode Codes). This bit will be ignored by the PE with NOOP and

Branch Return settings. Please also review CDP Mask/Value compare function settings.

NOTE: Branching may include Framing, but the user must end the current frame being executed prior to providing another start frame. Branching should NOT be used with subframing (see note in Subframing).

Bit 24: Branch Return/Jump: W

1=Branch Return. This bit must be set by the User where a return from logic Branch is expected. The PE does NOT have a stack for push/pop of branch address – the user must manually program the BCCB execution tree. If this bit is set, then the PE will immediately branch (jump) to the BCCB pointed to by the Branch/Return Address in BCCB offset 0x0020 (so this bit can be used to execute a hard jump if desired). This bit is checked and overrides NO-OP and 1553 message type bits. No time delays or other BCCB option bits are checked if this bit is set.

Bit 25: NO-OP: W

1=NO-OP – This bit directs the PE to skip this BCCB. NO-Ops are usually used as "place-holders" for on-demand 1553 messages that the user may want to turn-on/off during real-time BC execution. This bit setting overrides 1553 message type bits (but not Branch Return). No other BCCB functions (Frame Schedule/Intermessage Gap, Subframing, Branching) are checked for execution – the PE simple moves down the BCCB linked-list chain and checks for the interrupt option. This message can add up to 10-15 µsecs of time in the message chain. An interrupt may be generated for a NO-OP BCCB by setting the Int on Message Complete bit in the BCCB CSR.

1553 Message Type Bits. This next 5 bits designate the 1553 message type. Only one may be set. For the command word(s), the PE will send whatever bit values are programmed by the user in BCCB offsets 0x000X (first transmitted Command Word) and 0x000X (second transmitted Command Word for RT-RT messages). The user must load the proper/desired bit values – the PE will read the programmed word count and transmit the appropriate number of data words from the current BCCB CDP.

Bit 26: BC-RT: W

1=BC-RT (including BC-RT Broadcast).

Bit 27: RT-BC: W

1=RT-BC.

Bit 28: RT-RT: W

1=RT-RT (including RT-RT Broadcast).

Bit 29: Mode Code With Data Word: W
1=Mode Code with one Data Word (including Broadcast).

Bit 30: Mode Code Without Data Word: W
1=Mode Code without Data Word (including Broadcast).

Bit 31: Reserved

CMD1 Info | CMD1 value: W: 0x0010

This word is set by the user to set the value of the first 1553 Command Word to be transmitted and set error injection options for the word. This word is used for all messages except the second Command Word of an RT-RT message (see next paragraph for CMD2). The 16 MSBs (16-31) provide the error injection information (explained in the following paragraphs, and the lower MSBs (0-15) provide the Command Word value to be transmitted (bit 15-0 corresponds to 1553 bits 4-19, respectively).

CMD2 Info | CMD2 value: W: 0x0014

This word is set by the user to set the value of the second 1553 Command Word of an RT-RT message to be transmitted and set error injection options for the word. This word is used only for the second Command Word of an RT-RT message. The 16 MSBs (16-31) provide the error injection information (explained in the following paragraphs, and the lower MSBs (0-15) provide the Command Word value to be transmitted (bit 15-0 corresponds to 1553 bits 4-19, respectively).

Frame Time: W : 0x0018

This 32-bit, 100 ηsec word (6+ minutes in time) is set by the user in BCCBs' that has the Start Frame Designator bit set (bit 22 of the BCCB CSR). Each frame time, which is designate with start and stop Frame bit settings in BCCB CSR's, may be of variable time (a unique feature to ***AltaCore***). Minimum setting should be 340 (34 µsecs).

The PE will read this value and jams an internal count-down timer for reference on minor frame time execution (and for reference in executing/flagging Scheduled Message BCCB Gap Times, "frame overrun" conditions, and to validate remaining time for possible LPAM execution). With proper programming, start frame time accuracy is within 500 ηsecs (see note below).

NOTE: This user must include all time necessary to execute the frame including BC retries, branching/returns, Low Priority and High Priority messages **PLUS** 10 µsecs of dead bus time for PE overhead processing. If the PE does not have 10 µsecs of dead-bus time at the end of the current minor frame (all BC transmission has stopped for the minor frame), then the start of the subsequent frame's first message may not be accurate to 2 µsecs.

NOTE: The minimum setting when using Framing is 340 (34 µsecs). This minimum 1553 message is 24 µsecs (Broadcast Mode Code without Data Word) and then we must allow 10-20 µsecs for end of frame PE overhead processing.

Message Time Gap: W: 001C

This 32-bit, 100 ηsec word (6+ minutes in time) is set by the user for the time gap (dead bus time) to be executed PRIOR to this BCCB's Command message transmission. Depending on the setting of bit 4 Message Gap Type of the BCCB CSR (see above), this time value will be Intermessage Gap (IMG) Dead Bus time (bit 4 set to zero) or Frame Schedule (bit 4 set to one), which is a time offset from the beginning of the minor frame (the last BCCB that had bit 2 of the a BCCB CSR set for Start Frame Designator). For the first BCCB of the minor frame, a Dead Bus time or Frame Schedule results in the same execution. The 1553 standard requires a minimum gap of 4 µsecs (which is 2 µsecs for this programming – see note below).

If the wait for trigger bit is set in the BCCB CSR for this BCCB, this word specifies the time gap after the trigger has occurred to this BCCB's Command message transmission.

NOTE: This time is DEAD BUS TIME. This is not standard 1553 intermessage gap times (IMGs). 1553 IMGs are measured from mid-bit parity to mid-bit sync (2 µsecs of extra time, but not actually dead bus period). The ***AltaCore*** gap times are true dead bus (idle) time for encoded transmission. For example: The minimum 1553 IMG is 4 µsecs per the specification; this would be 2 µsecs for ***AltaCore*** Message Gap Time. A simple subtraction of 2 µsecs for standard 1553 IMGs will provide the ***AltaCore*** program time.

NOTE: If Frame Schedule is selected, then this time represents the offset time from the beginning of the minor frame (being a time t_0 from the beginning of the minor frame). The PE has an internal count-up timer that is reset to zero at the beginning of the minor frame. The PE will wait until the frame timer is greater than this Message Gap Time value before executing the BCCB.

NOTE: The user must program a value of at least 20 for accurate message timing to 500 ηsecs. Branch/Returns, Retries, Subframing, High or Low Aperiodic messages may increase gap times 10 µsec per instance.

NOTE: IMGs may be affected by 2-5+ µsecs with multi-channel cards due to memory contention between channels. This is a rare occurrence with heavily loaded channels.

Address Branch/Return Address: W: 0x0020

This is a dual use register depending is Bit 20 or 23 of the BCCB CSR is set.
If bit 20 is set, then this value is the destination BCCB branch address IF the test mask/value comparison is true (non zero after mask/compare).

If bit 23 is set, then this address word is set by the User and is the pointer to BCCB that should be executed if a non-zero value is found after the CDP mask function is checked (please see Section 8 on CDP). Branching and Returns may add up to 10-15 µsecs in gap time.

Starting Frame | Frame Increment Word | Stop Frame:
Test Address, Test Mask, Test Data: W: 0x0024-002C

These registers are is dual use register depending if Bit 20 is set.
If Bit 20 is not set, then these three words are set by the user to designate the starting frame, frame increment and stop frame for the PE to execute sub framing of this BCCB (subframing is BCCB execution at sub-increments of the major frame – the size of the major frame is set by the User in the Root BC Frames Per Major Frame register). The User should not change (write) these values once BC execution has started. Subframing may add up to 10-15 µsecs of gap time per respective BCCB.

If bit 20 is set, then these three words are the Test Address, Data Mask and Data Value for the Address Branch Function. Please see the bit 20 details above and the word explanations below.

Starting Frame: W:
Test Address: 0x0024

If bit 20 is not set, this word is set by the User and directs the PE to not transmit this BCCB until the Start Frame value. The Current Minor Frame count can be initialized to zero by the User, but when the designated Frames Per Major Frame has been met, then PE Major Frame counting resets to one (see Root BC Frames per Major Frame). If this value is set to zero, the PE will bypass the Start Frame Check, i.e. Start Frame condition will always be true. See Subframe discussion after the Frame Stop (0x002C) paragraph.

> **NOTE:** If desired, the user can have a frame zero that could be used as an initialization frame. When the first Major Frame has been completed, frame counting will reset to one.

If Bit 20 is set, this value is the test memory address for the Address Branch function. The data value at this address is used in the mask and comparison of the next two registers.

Frame Increment: W:
Data Mask: 0x0028

If bit 20 is not set, this word is set by the User and directs the PE to execute the BCCB every Nth frame AFTER the start frame. Minimum value is one or greater. A value of zero will cause the BCCB to execute only once because the PE will utilize this value only after the BCCB has been executed, which means a value of zero will direct the PE to not execute this BCCB again until the frame count, also known as a major frame, resets/cycles and the Start Frame condition has been met.

If Bit 20 is set, this value is an AND mask used against the data value pointed to by the previous register Test Address.

Frame Stop: W:
Data Value: 0x002C

If Bit 20 is not set, this word is set by the User and directs the PE to not execute this BCCB after the Nth frame of the minor frame. If this value is set to zero, the PE will bypass the Stop Frame Check, i.e. message will not stop in a minor frame once it has started transmission in that frame.

> **NOTE:** If the User is using frame counting (subframing), then each BCCB must have the proper values programmed for execution. For example, if the User application wants this BCCB to be executed on every frame, then the Starting Frame would be set to one, the Frame Increment would be set to one and the Stop Frame value would be set to N, which is the frames per major frame (N).

If bit 20 is set, this is the final value expected to cause a BCCB Address Branch. The BCCB Address Branch (0x0020) will occur if this value is the same as the Test Address data (0x0024) ANDed with Data Mask (0x0028).

Subframing Discussion

Subframing can greatly simply BC message transmissions that occur in various frequency denominations of the number of frames per major frame. With subframing, the PE must “touch” each BCCB to calculate and store the next frame number for execution, so Subframing cannot be used with branching because the branched BCCBs will not be touched on each frame. Also, the BCCB type NOP will not be touched, but Delay Only will be touched (so can be used with subframing).

The user can achieve the same subframing execution result in BCCBs with branching by setting up/managing the BCCB and CDP pointers. This can be a bit more complicated in setup, but not that difficult (a few more lines of code). There are example programs to show how to manually setup subframing without using this built-in feature.

16 LSB Next Frame Value: 0x0030

The LSB 16 bits of this word is reserved by the PE to track subframing for this BCCB (when this BCCB is executed the PE calculates the next frame for execution and saves the value in this lower LSB 16 bits – the user should NOT change this value of this word once BC execution has started).

API – Message Number: 0x0034

The API uses message numbers rather than memory pointers to identify BC Control Blocks. This word is used by the API to store the message number associated with this BCCB.

API – Number of CDPs: 0x0038

The API allocates CDP buffers for each message (BCCB). This word is used by the API to store the number of CDP buffers allocated for this message.

API – First CDP Pointer: 0x003C

This word provides a pointer to the first CDP buffer for this message. The API uses this pointer to read or write CDP buffer data for the message.

<~~~>

AltaCore-1553 Playback (PB)

Introduction

Playback is a form of BC that consists of a simple list encoding instructions of 1553 words that are transmitted at programmed intermessage gap intervals. Playback transmissions may or may not include RT responses.

Often, playback is associated with the replay of bus recording files, but playback can also be a simplified form of 1553 transmissions where the user wants to generate lists of BC and RT transmissions without the need to program BC frames and schedules – simply defined list of messages with time tags and transmit. Playback can be invaluable for test and simulation applications.

For example, an operator could use ***AltaView*** to record a flight (or simulated) flight segment and then use this recording to stimulate lab/depot level Line Replaceable Units (LRUs) for testing or troubleshooting. When coupled with the Ext Clock input/output capability of ***AltaCore/AltaView*** Analyzer, Playback can be highly accurate to 200 ηsec between channels/cards – the best in the industry.

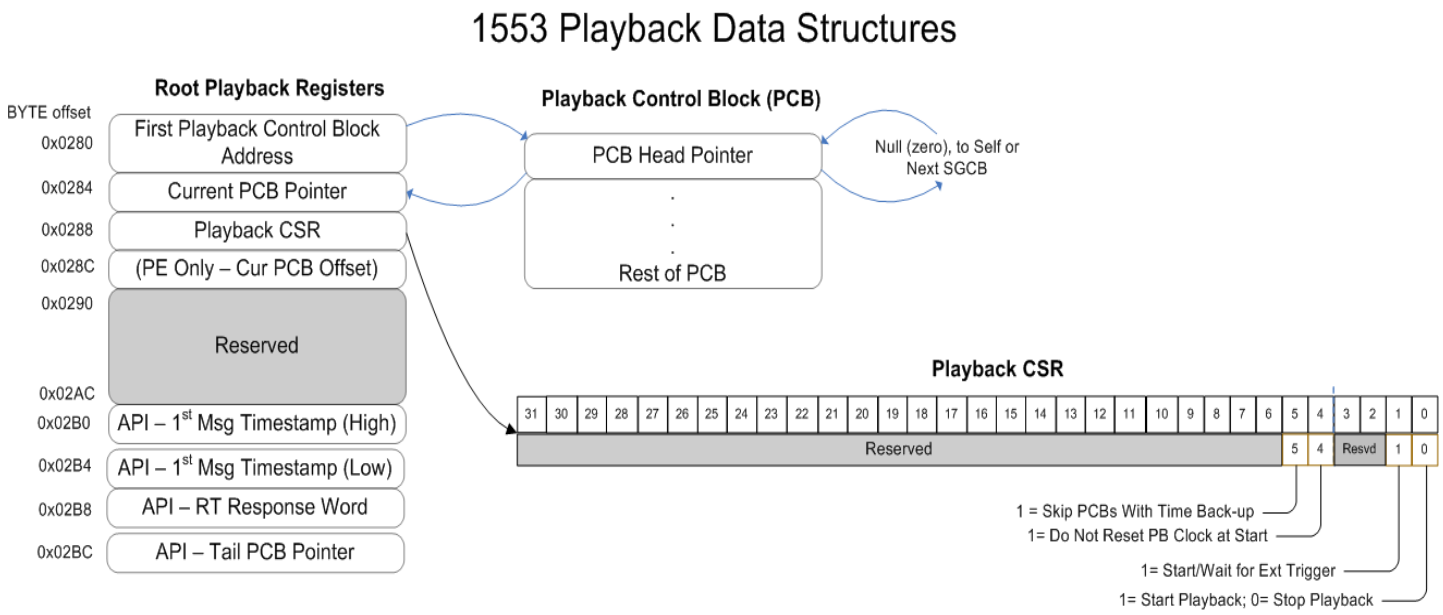


Figure PB-1: Playback Root Registers

The ***AltaView*** and ***AltaAPI***can translate recorded files and massage them for bus Playback and remove RT responses to generate test responses from LRUs. Playback can be a simple BC message generator where the user does not want to program more complicated frames/BC sequences (and just wants to transmit lists of messages with

time gaps). In ***AltaCore*** PE 1553 features, Playback is somewhere between the sophisticated BC framing controls and the Signal Generator bit-level transmissions.

Playback has its' own internal 64-bit, 20 ηsec clock for transmission timing control (for comparisons to PCB time values). This clock may be set, read and synchronized to an external clock (just like the main PE clock) through the Root PE Control Word. Please Root PE Registers Section for more details (Root PE Control Word).

Protocol errors are duplicated only with 3rd bit Manchester (first bit after 1553 sync), Parity, Sync and Word Counts (playback is a digital re-transmission, not a raw A/D re-transmission). Please see individual PCB 1553 word encoding options for error injection options.

Similar to BC and Signal Generator functions, Playback is setup and controlled through a set of Root Playback registers and Playback Control Blocks (PCBs). Root registers control Playback operations and the PCB contains individual message encoding and timing information. PCBs can be time-controlled based on relative time from a zero time value (most popular for general users), or can be time controlled by the actual, absolute time (AT) of their PCB time values (a more desirable method for applications trying to synchronize multiple events in a system to a common clock). The text will discuss the value of both options.

AltaAPI provides a simple CDPWrite() style function call that takes CDPs from the user and performs all the necessary conversions to PCB structures. This should greatly simply the user's application so that detailed knowledge of PCBs may not be necessary.

Playback Transmission Timing

The PE playback logic for time controlled transmission is fairly straight forward. When the PE reads a PCB, the 64-bit time value and first 1553 word(s) are loaded to a lower level PE transmission encoder. The encoder does a tight hardware 20 ηsec comparison of the time value (regardless of possible Root PE Control Word external clock time setup/synchronization), and when the PCB time value is less-than-or-equal-to (<=) to the internal PB clock, then the 1553 words of the PCB are transmitted.

The user has an option to "Skip PCBs with Time Back-up" in the Root PB CSR. If this bit is set, then the PE Playback logic changes slightly. If the bit is set, then the PE will look at each PCB time value and compare if the time is less-than (<) the PB internal clock: if the time is less-than, then it is skipped completely (and all trigger and interrupt settings of the skipped PCB are ignored); if the time is equal-to-or-greater-than (>=), then the PE loads the PCB to the lower-level transmission encoder and the 20 ηsec time comparison logic takes over to very tightly control/compare when the 1553 words

start to be transmitted. Once the PCB time and data information is loaded to the low-level PE encoder, transmission accuracy is within 200 ηsec.

> **NOTE:** Minimum intermessage gap times between PCB (1553 messages) should be 5-6 usec for playback. The user may need to adjust CDP files to ensure this minimum gap time when using Absolute Timing.

PCB Timing Discussion – Relative Timing

For default, relative timing the PE Playback engine has an internal clock that is used to compare against each PCB's user set time. For most applications using relative timing, this PE Playback internal clock is reset to zero when the Playback function is started – so future PCB transmission times must be from a relative value of zero. When Playback starts, the clock resets to zero and then the PE compares each PCB Time High/Low (set by the user) to the internal clock value: If the PCB Time High/Low is <= to the PE Playback internal clock, then the PCB is executed (transmitted). If the PCB Time High/Low is > than the internal clock, then the PE stalls/loops until the PCB time is <= to the internal clock.

AltaAPI CDPWrite() automatically takes a CDP string and makes the offset-zero time calculations, so the user can simple pass CDPs from a file to the API call and not worry about the time offsets to zero. If the user wants to have a start delay, then the user should increase the CDP Time High/Low by the desired start delay time prior to calling the ***AltaAPI***Playback functions.

Relative Timing is the default setting of PB data structures (control bits are set to zero).

PCB Timing Discussion – Absolute Timing (AT)

The ***AltaCore*** PE also has options for users to have PCB transmission timing controlled by the absolute value of the PCB time words (which probably came from a CDP time value). This simply means the PB internal clock is not reset to zero and the user must preset the PB time value through the Root PE Control Word time control bits. Once started, the PB internal clock will count up and look for PCBs with time values <= to the internal PB clock (just like with relative timing).

The user must set the PB Control Word "Do Not Reset PB Clock at Start" bit to have absolute timing. The user can also set the "Skip PCB with Time Back-up" for the PE to skip over PCBs that have a time value < than the internal clock – this is useful only for applications where the user as preload PCBs and does not want transmission started until a certain time has elapsed (most applications will simply want to "load and go").

The following paragraphs detail playback data structures.

Root First Playback Control Block Address: W: 0x0280

This register is set by the User with the address of the first PCB link of the PCB chain. When bit 0 of the Playback CSR (0x0288) is set to one, the PE will use this address to start Playback transmissions.

Root Current PCB Pointer: 0x0284

This register is updated by the PE with the current PCB being executed. The User may read this register to monitor transmission progress. The User should not write to this register. The PE will set this register to zero when transmission has halted by either encountering a null PCB head pointer or by the Playback function being stopped by the User setting bit 0 of the Playback CSR to zero.

Root Playback Control and Status Register (CSR): W: 0x0288

This register has two bits settable by the User to control the starting and stopping of Playback transmission.

Bit 0: Start/Stop Playback: W

This bit is set to one by the User to Start Playback transmission. The User set this bit to zero if Playback transmission is to stop after the completion of the current PCB. The Root Current PCB Pointer (0x00284) will set to 0x00000000 when transmission has halted.

Bit 1: Wait For Ext Trigger: W

This bit is set to one by the User if Playback transmission is to wait until the external trigger line is set (pull down to ground for at least 1 µsec).
Bits 2-3: Reserved

Bit 4: Do Not Reset PB Clock at Start: W

This bit is set to one by the user/API if the PE should NOT reset the playback internal clock at the start. This bit must be set for absolute timing. Usually, this user/API will have set/preset the PB internal clock with a desired value prior to starting PB (setting bit 0 to one).

Bit 5: Skip PCBs with Time Back-up: W

This bit is set to one by the user/API if the PE should skip PCBs with a time value less than the PB internal clock.

Root Reserved Word: 0x028C

This word is reserved for PE usage to track PB execution (tracks the current offset in a PCB).

Root Reserved Words: 0x0290-0x02AC

These words are reserved for PE processing and should not be written to by the User.

API – 1st Message Time High: 0x02B0

This word is used by the API to store the high 32-bits of the 64-bit time-stamp of the first message in the playback sequence.

The API processes CDP records and converts them to Playback Control Blocks. If Absolute Timing (AT) option bit is not set, the API converts the time-stamps from the CDP records to time relative to the start of playback. The API uses this register and the next to store the 64-bit time-stamp of the first message. The API subtracts this value from the CDP time-stamps to calculate the time relative to start of playback to program into the corresponding Playback Control Blocks.

API – 1st Message Time Low: 0x02B4

This word is used by the API to store the low 32-bits of the 64-bit time-stamp of the first message in the playback sequence.

API – RT Response Word: 0x02BC

This word is used by the API to store the playback RT response settings for the channel. Each of the bits in this word correspond to an RT address (bit 0 corresponds to RT 0, bit 1 corresponds to RT 1, etc.). If the bit is set then the API will include the recorded RT response (status and data words) for that RT address in the playback. If the bit is clear, then the RT response for that RT Address will not be included in the playback.

API – Tail Pointer: 0x02C0

This word is used by the API to store the playback "tail" pointer. This is a pointer to the next PCB to be written with new playback data. The API uses the Root Current PCB Pointer (0x0280) as the "head" pointer, which indicates where the PE is processing in the chain of PCBs. The API periodically checks to see if there is room in the buffer for more PCBs by reading the head and tail pointers – if they are not equal, then the API can write new PCBs to the buffer until the tail pointer equals the head pointer. This is how the API can maintain continuous playback of data from a large number of recorded CDP records.

Playback Control Blocks (PCBs)

PCBs are linked list structures that contain 1553 word values, encoding information and gap time values.

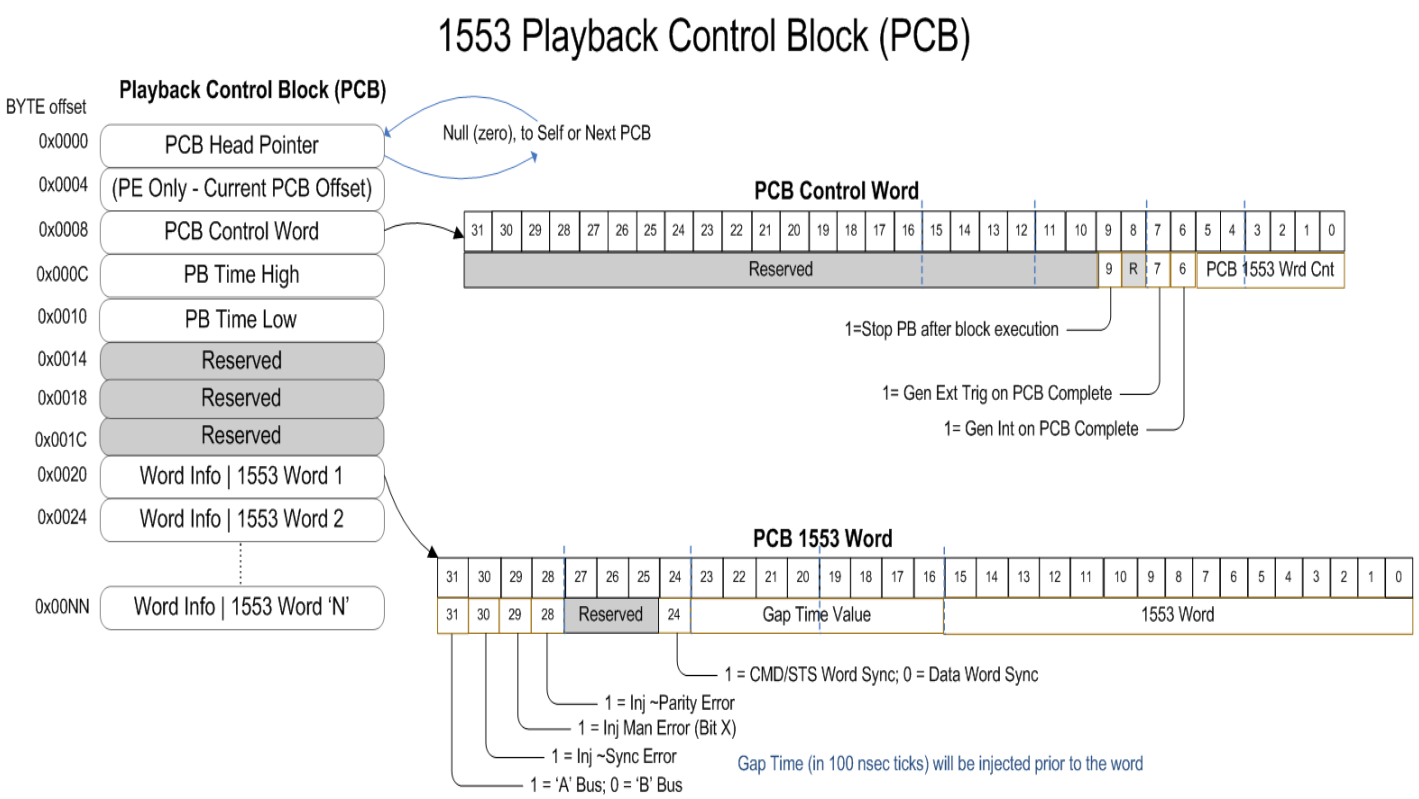


Figure PB-2: Playback Control Block (PCB)

PCB Head Pointer: W: 0x0000

This word is set by the user with the address of the next PCB structure or set to zero for a single or last PCB in the linked- list.

Current PCB Offset – PE Only: 0x0004

This word is reserved for PE process tracking (it holds the offset in PCB for PE processing).

PCB Control Word: W: 0x0008

This word provides the user options for Playback execution and the bit values are described in the ensuing paragraphs.

Bits 0-5: PCB 1553 Word Count: W

These bits are set by the user with the number of 1553 words to be transmitted from the PCB. Minimum is one and maximum is 63.

Bit 6: Gen Int on PCB Complete: W

This bit is set to one by the user to direct the PE to generate and interrupt when the current PCB has completed execution. Please see the Interrupt Section for details.

Bit 7: Gen Ext Trigger on PCB Complete: W

This bit is set to one by the user to direct the PE to generate an external trigger when the current PCB has completed execution.

Bit 8: Flash User LED on PCB Complete: W

This bit is set to one by the user to direct the PE to flash the user LED when the current PCB has completed execution. Please see your Hardware Manual for LED location.

Bit 9: Stop PCB Execution: W

This bit is set to one by the user to direct the PE to stop execution after the current PCB has completed execution.

PCB Time High & Low: W: 0x000C-0x0010

These two words are set by the user to direct the PE to not transmit the PCB 1553 words until this time is "less than or equal to" (<=) the PE Playback internal control clock (if the "Skip PCBs with Time Back-up Bit is set in the Root PB Control Word, then this comparison is tightly latched to equal '=' only). This is the "time of day" value that the user wants this PCB to be executed relative to an offset of zero. With minimum intermessage gaps of 5-6 µsecs, Playback is extremely accurate to within 1-2 µsecs or better (usually within 1-2 clock ticks of the reference clock).

The comparison is done at 20 ηsec resolution, so the user must be careful to have the proper time values.

Reserved: 0x0014-0x001C

1553 Word Encoding: 0x0020-0x005F (Variable to 0x005F)

The next 1-63 words are set by the user with the 1553 word values and transmission encoding information. The first 16 MSBs of each word are the encoding bits and the LSB 16 bits are the actual 1553 word bit values that map one for one with 1553 word formats (bits 4-19).

The upper 16 bits are encoding bits with the following values.

Bits 16-23: Gap Time Value

These 8 bits are set by the user to inject the idle gap (dead bus) period prior to this word. The resolution is 100 ηsec ticks.

Bit 24: 1=Command/Status Word Sync; 0=Data Word Sync

This bit is set to one by the user to direct the PE to transmit a positive-going (high to low) sync as required for Command and Status Words. A value of 0 (zero) will direct a negative-going (low to high) sync as required for Data Words.

Bit 25-27: Reserved

Bit 28: Parity Error

This bit is set to one by the user to direct the PE to transmit “even” Parity for this word.

Bit 29: Manchester Error

This bit is set to one by the user to direct the PE to inject a bad Bi-Phase, Manchester encoding on bit 3 (first bit after sync) of this word.

Bit 30: Sync Error

This bit is set to one by the user to direct the PE transmit an inverted sync pattern as compared to the value programmed in bit 24.

Bit 31: 1=A Bus; 0=B Bus

This bit is set to one by the user if the word is to be transmitted on the A bus. This bit should be set to 0 (zero) if the word is to be transmitted on the B bus.

<~~~>

AltaCore-1553 Signal Generator (SG)

Signal Generator

The ***AltaCore*** PE provides unique capability to transmit signal vectors at 20 ηsec intervals to generate custom digital waveform patterns to the 1553 transceiver. This allows the user to create detailed 1553 word and signal patterns, ideal for error testing, and even exceeding AS15531-4111/4112 RT Validation protocol error injection requirements. Alta uses this feature ourselves to execute detailed error injection for testing of the ***AltaCore.*** This is the most advanced signal generation capability on the market today.

The reader may want to review the ***AltaRTVal*** product. This Windows application greatly simplifies AS4111 and AS4112 RT Testing tasks and makes extensive use of the Signal Generator function.

Signal Generation (SG) is used instead of normal BC transmission. BC cannot be active at the same time as SG.

Overview of Signal Generation

The PE shall provide 1553 signal generation functions through data structures of bi-level state vectors, which represent bus voltage states as fed to standard 1553 transceivers. The vectors are represented as bits in packed words that are linked-listed (called Signal Generator Control Blocks – SGCB) together to provide convenient boundaries for user defined intervals or 1553 messages. There is a 32-bit time gap word in the link to provide intermessage gap (IMG) or response time gaps. The follow paragraphs detail the Signal Generator data structures.

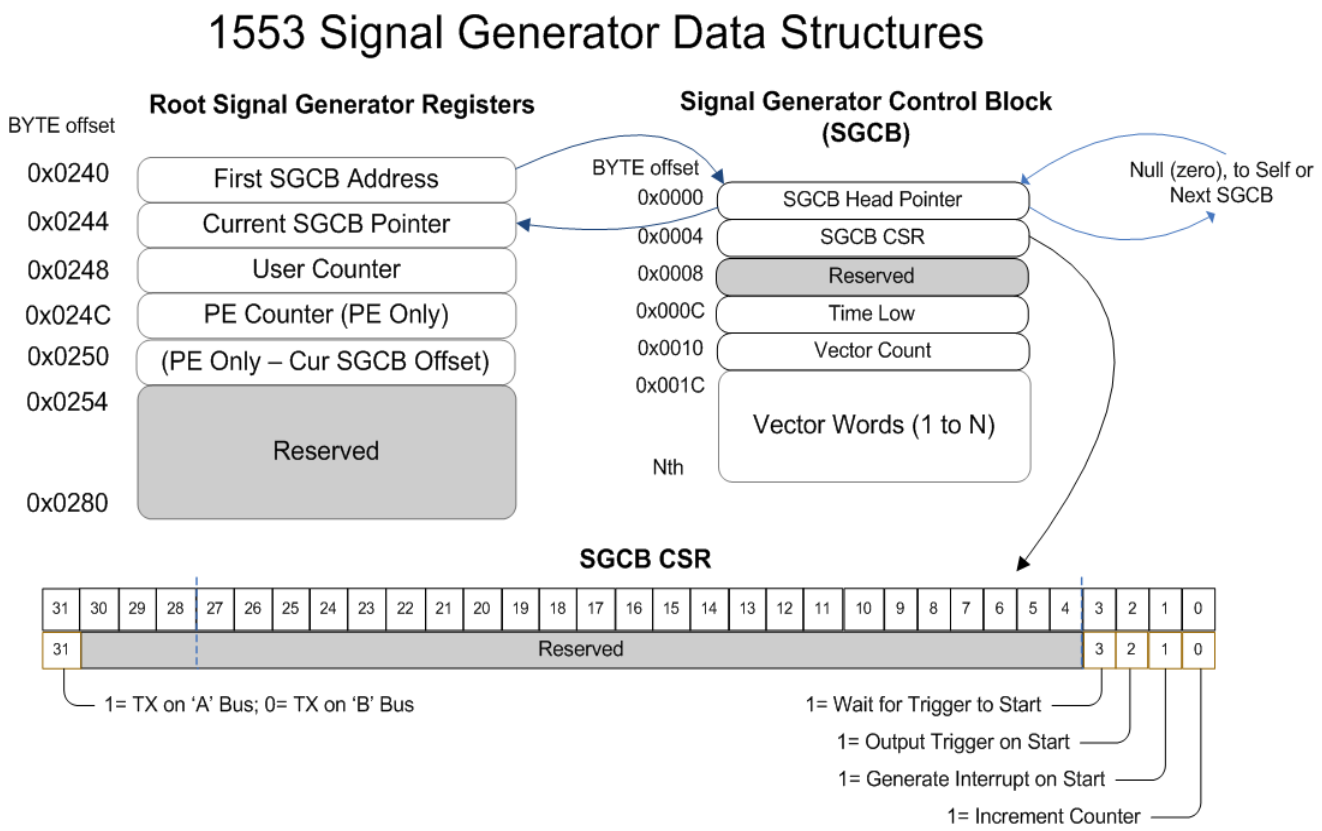


Figure SG-1: Signal Generator Data Structures

Root PE Signal Generator Control Bits

Please review the Root PE Control Word for starting the Signal Generator. Bit 4 of the Root PE Control Word is used to start this function.

> **NOTE:** The PE will check the BC start bit to make sure it is set to zero before starting Signal Generator. If the BC start bit is set in the Root BC CSR, then Signal Generator will not operate.

> **NOTE:** RT and Sequential Monitor functions are allowed during Signal Generator.

> **NOTE:** The User can stop Signal Generator by setting a loop count or by setting this bit back to zero. This bit is checked prior to starting each PCB. The current PCB will completely execute.

2-Bit, Bi-Level Vectors

Bi-Level vectors directly encode the bi-level (2 signal lines) to the 1553 transceiver; every two bits of a 32-bit word encodes one signal vector clock time. The main use for Bi-Level vectors is for detailed testing of 1553 decoders where the user may want to control the signal generation to test exacting bus conditions. A two bit vector setting of "11" or "00" will encode a 1553 ground state (0 voltage). A vector setting of "10" will

encode a positive voltage state and a “01” will encode negative voltage state. Here are some math facts and example:

- A 32-bit word encodes 16 vector/voltage states and the PE clock frequency is 50 MHz (20 ηsec ticks), so it takes 50 clock ticks/vectors per micro second.
- There are 20 bits in a 1553 word (3 sync bit + 16 word bits+ 1 parity bit)
- There are two bits per vector state
- It takes 2000 vectors to encode one 1553 word (50 states x 20 bits x 2 bits per state)
- A 34 word 1553 message (One Command + One Status Word + 32 Data Words) is 68,000 vector bits (34x2000).
    - We must also encode the Response Time and we’ll assume 6 µsecs response, or 6x50x2 = 600 vector bits. So 68,600 vector bits encoded with 32-bit words is 68600/32 = 2143.75 words – round up to 2144 words.
    - This does not include intermessage gap time if we chain multiple SGCBs together.

So it takes a lot of bits and words of the PE extended 1 Mbyte of memory to encode messages. The user can monitor memory usage and Signal Generator execution process and “page” or reload vector packets on the fly. Most applications, however, will load only a few messages at a time, execute them, and then check testing status. The ***AltaAPI***provides memory management tools to check available memory while building your process, and there are several example programs to help kick start this programming process.

Root Signal Generator Registers

Root First SGCB Address: 0x0240: W

This register is set by the User and provides the PE the starting address of the first SGCB. The Signal Generator On bit in the Root PE CSR controls the starting and stopping of Signal Generator function.

Root Current Signal Generator Address: 0x0244: W

This register is set by the PE and provides the User a monitor value to indicate the current Signal Generator data structure link of the chain that was loaded for PE execution/sequencing. The User should not alter the contents of the current SGCB.

NOTE: The PE will set this register to 0X00000000 when Signal Generator stops.

Root Signal Generator Count: 0x0248: W

This word allows the User to set a loop count for the PE to execute the link-list of SGCBs. This word is defined by the user (0x00000001-0xFFFFFFFF values) and is the total loop count.

NOTE: A value of 0x00000000 will direct the PE to IGNORE the Starting SGCB Bit and will cause an infinite loop of SGCB link-list execution. The User can stop Signal Generator function by writing a zero to the Signal Generator Off bit in the Root PE CSR.

Root PE Count: W: 0x024C

This word is reserved for the PE for tracking the current count. When the PE count in this location equals the Signal Generator Count in the above register, the PE will write 0x00000000 to the current SGCB Address and will clear Signal Generator Mode ON bit in the Root CSR register.

Root PE Current Vector Location: 0x0250

This word is reserved for the PE for tracking the offset to the current vector word.

Root Reserved 0x0250-0x0280

Signal Generator Control Blocks (SGCB) – Extended Memory

This variable length data structure provides vector state and timing for the PE to generate for Signal Generator/signal construction. This is a linked-list structure with a head pointer, a SGCB Control and Status Register, one reserved word, one time gap word, a vector count word and a variable list of words that contain the raw vectors.

Next SGCB Pointer: W: 0x0000

This word is set by the User to point to the next SGCB. A zero value will cause Signal Generator execution to stop (and a value of 0x00000000 will be written to the Root Signal Generator Current Pointer). The user may see up to 10-15 µsecs of delay (dead bus gap) between SGCBs.

SGCB Control and Status Register: W: 0x0004

This register provides control and status options for SGCB execution.

Bit 0: Starting SGCB Bit: W

The User sets this bit to designate the first SGCB link of the chain. This bit is only used with Signal Generator count as described in paragraph above. When this bit is set, the PE will increment the PE count by one when the given SGCB is processed.

Bit 31: Bus Selection Bit: W

The MSB is set by the User and directs the PE to transmit the subsequent vector list on the A bus (set this bit to one) or the B bus (set this bit to zero).

Reserved Word: 0x0008

Gap Time: W: 0x000C

This word is a 32-bit, 20-ηsec time value for inserting a zero level gap. A time of zero will cause immediate execution without gaps. The user may see up to 10-15 µsecs of delay (dead bus gap) between SGCBs.

> **NOTE:** To ensure no signal jitter between SGCBs, it is recommend that the User pack all vectors between gaps (IMGs or response times – dead bus time) in the same SGCB. Switch between busses with no gap may also cause slight jitter (nanoseconds).

Vector Count: W: 0x0010

This 32 bit word provides the raw 2-bit vector count for states to be transmitted in this SGCB (each 2-bit vector is one count). If Vector count is zero, the given SGCB is treated as a NO-OP and execution jumps to the next SGCB.

> **NOTE:** If Vector Counts do not align with the last word of the list, the PE will execute from the MSB (bit 31) downward to the last count (with the remaining LSB bits not clocked-out/transmitted).

SGCB Vector Bit Words: W: 0x0014-Variable

The following one to N words are the packed vector states. The number of words is dependent on the number of states designated in the SGCB Vector Definition Word (described above).

> **NOTE:** The PE is capable of transmitting/clocking-out vectors at the full 50MHz clock rate, but most 1553 transceivers have only a 200 ηsec frequency response. The PE will not error check this condition and the User should ensure the encoding of vectors to follow this guideline.

<~~~>

RT: Active and Map Monitor

The ***AltaCore*** Protocol Engine supports 1-32 Remote Terminal units and offer a wide choice of programming options for data buffering and protocol options. The Root settings allow ***AltaAPI***or user program to control basic RT functions and designate if the RT is an active terminal or is an inactive/passive monitor for the respective RT Subaddresses (RT Map Monitor).

> **NOTE:** ***AltaCore*** supports RT 31 being a Broadcast receiver (for compliant 1553B systems) or as a normal RT address for older 1553A systems (this designation in selected in the Root PE Control Register). For Broadcast messages, the PE will store the respective data in RT 31 indexed/linked data structures for reference by all RTs. All RT options are detailed in the following paragraphs. In most cases RT 31 should always be on.

Overview of RT Functions

The RT data structures consist of Root RT registers (Root RT CSR), RT Control Blocks (RTCBs) a Filter Table array that filters RT Subaddress/Mode Code buffering and an RT Subaddress/Mode Code Control Block (SA/MC CB) that references legalization settings and CDP 1553 buffers. As with other ***AltaCore*** 1553 functions, CDPs can be indexed for one or more RTs/SAs/MCs to allow very flexible data buffering, wrapping and bridging applications.

Root 1553 RT Registers

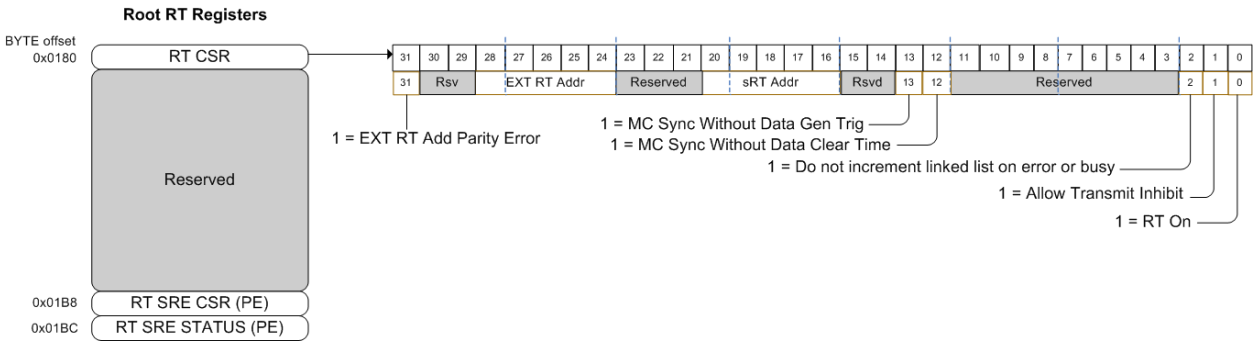


Figure RT-1: Root RT Registers - Root RT CSR

The Root RT Registers begin the RT structures and provide basic setup information of all RTs of the channel. Figure RT-1 below shows the registers.

Single RT and Multi RT(mRT) Modes & 1760 Addressing

Most Alta cards can support MIL-STD-1760 external RT Addressing and auto start-up modes. The user must set Single Ended Discrete lines and Enable lines (see your hardware manual for pin-outs). This feature is an advanced user function and please review the AltaAPI manual PDF bookmark section: **Single RT and Multiple RT Configurations.** Also, please contact Alta support (alta.support@altadt.com) for review and an application note for setting these discrete lines and software setup.

Root RT CSR: W: 0x0180

The Root RT CSR contains the main on/off bit control and active/map monitor selection and other key, Root level values.

Bit 0: RT On/Off: W

1= RT On: 0=RT Off: W/P. This bit is set to one by the User to turn on all RT operations. The User should setup all other data structures (especially RTCBs and SA/MC Filter Tables) prior to setting this bit to one. Setting this bit from zero to one will cause RT operations to halt after the current RT response is finished.

Bit 1: Transmit Inhibit Control: W

1=Allow Transmit Inhibits: W. This bit is set by the User to direct the PE to allow and act on transmit inhibits. This is global setting for all RTs of the PE channel. The PE will use two bits in the RTCB to track inhibits for the respective RT (this is detailed in the RTCB CSR paragraphs later in this section).

> **NOTE:** A zero setting will direct the PE to ignore transmit inhibits, which is against the 1553 standard, but common in test and simulation scenarios. The normal User setting is set this bit to one.

Bit 2: Do Not Increment CDP Link List on Error or Busy: W

1=Do Not Increment CDP Link List with Error or RT Busy. This bit is set to one by the User to direct the PE to ignore completion of the respective SA/MC Filter Table if a 1553 protocol error has been detected or if the RT busy bit is set for the respective SA/MC Filter Table message; this also directs the PE to NOT increment the SA/MC Filter Table CDP link list address. The proper setting for this bit is one since the 1553 standard calls for ignoring BC messages with errors (an invalid message) or when the Busy bit is set in the RT Status Word.

> **NOTE:** If this bit is set to zero, then the PE will complete the SA/MC Filter Table CDP (as best it can) and increment the respective SA/MC Filter Table CDP head pointer.

NOTE: This bit settings has no action on the ME bit being set as defined in the standard for invalid receive commands. If an error is detected in a Data Word of a receive command (the Command Word has decoded without error, but a BC Data Word had an error detected), then PE will set the ME bit for the subsequent command message sent to the respective RT.

NOTE: Regardless of this bit setting, the PE will complete the current CDP with as much 1553 data and error status information as possible (and in the scope for decoding a particular 1553 message).

NOTE: This bit is ignored when the RT is in mapped monitoring mode: CDP pointers will always increment.

Bits 3-11 Reserved

Bit 12: MC Sync Without Data Clear Time Tag

This bit is set to one by the user to direct the PE to clear the PE Time Tag (reset to zero) when a RT has received a Mode Code Sync Without Data (Mode Code 01). The Time Tag will reset for subsequent messages (CDPs) after the Mode Code Sync Without Data (CDP) Message. For mRT (multi RT) modes, this action will occur for any Mode Sync Without Data Command Word to any RT. For single RT configurations, the RT address must match the sRT address in this Root RT CSR.

Bit 13: MC Sync Without Gen Trig

This bit is set to one by the user to direct the PE to generate an external trigger when a RT has received a Mode Code Sync Without Data (Mode Code 01). The trigger will occur within five µsecs of the Command Word's parity bit. For mRT (multi RT) modes, this action will occur for any Mode Sync Without Data Command Word to any RT. For single RT configurations, the RT address must match the sRT address in this Root RT CSR.

Bits 14-15 Reserved

Bits 16-20: Single RT Address: W

These bits are reserved for the PE for configurations of single RT verses the default multiple RT capability. For default, these bits should be set to zero.

Bits 21-23: Reserved

Bits 24-28: RT Ext Address: W

These bits provide the value the user has set on the external discrete lines for a hardwired RT Address (first channel only). Bit 24 corresponds to bit 8 of the Command Word RT Address value.

Bit 29-30: Reserved

Bit 31: External RT Address Parity: W

This bit holds the value of the parity discrete for external RT Address capability for channel one.

Root RT Reserved Words: W: 0x0181-0x001BC

These registers are reserved for future use. The last two registers are used for tracking values of for PE. These values should be initialized to zero and then not accessed by the user program.

Root RT Control Blocks (RTCBs): W: 0x0400-0x05FC

The RTCBs are the main index and control/status values for individual RT operations. There are 32 RTCBs, which are a 4 word block, one for each RT. The first word of an RTCB is the pointer to the RT's Filter Table and the second word is the RT's CSR. Other words are detail in the following paragraphs. Figure RT-2 shows the RTCB.

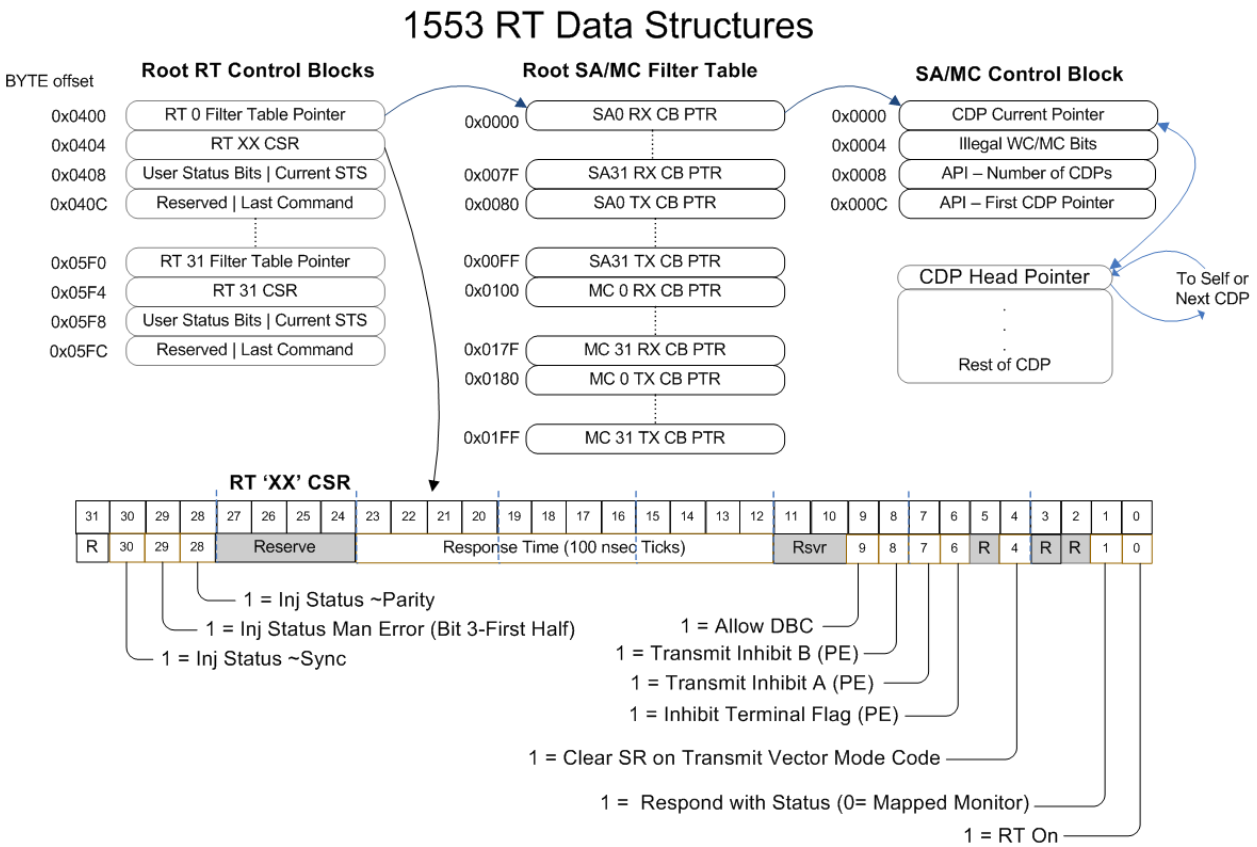


Figure: RT-2: RT Control Blocks, RTCB CSR and Filter Table

RT XX Filter Table Pointer: W: 0x0000

The first word of an RTCB is the Filter Table Pointer for the respective RT address (XX). The Filter Table further indexes RT Control Blocks for Subaddress and Mode Code execution (more on SA/MC Control Blocks later in this section) for data buffering. The pointer value can be any value in external memory (above 0x00000800 for the channel memory relative location).

RTCB CSR: W: 0x0004

This word is a packed bit data structure that provides settings to vary RT responses (for variant 1553 protocols), RT Status Word Response Time and RT Status Word Error Injection settings. The bit structures are detailed in the following paragraphs:

Bit 0: RT On/Off: W

1=RT On: 0= RT Off. W. This bit is set to one by the User to turn on the 1553 responses or Map Monitoring functions for the respective RT. The next bit designates the function.

Bit 1: RT Active/Map Monitor: W

1=RT Response Enabled: 0= RT Map Monitoring: W. This bit is set to one by the User to direct the PE to respond to BC Commands sent to this respective RT.

This bit is set to zero by the User if this respective RT is to be map monitor function (and not respond to BC commands).

Bits2-3: Reserve

Bit 4: Clear Service Request (SR) Bit on Transmit Vector Mode Code: W

1= Clear SR Bit with Transmit Vector Mode Code is Received. This bit is set to one by the User to direct the PE to clear (set to zero) the Service Request (SR) bit of the RT's Status Word when a Transmit Vector Mode Code has been received. The SR Bit is cleared for messages AFTER the Transmit Vector Mode Code Message (the SR bit will not cleared on the Transmit Vector Mode Code Status Word response). This function saves the host application from clearing the SR bit in this common 1553 application scenario. The default setting for the API is zero.

Bit 5: Reserve

Bit 6: Inhibit Terminal Flag

This bit is used by the PE for tracking Inhibit Terminal Flag Mode Code Setting. The API/User may read this bit, but should not change its value.

Bit 7: Transmit Inhibit A: W

1=Inhibit (disable) Bus A. This PE uses this bit for temporary settings for Transmitter Shut Mode Code for bus A. The API/user can also set this bit to one to direct the PE to turn-off bus A transmission. The user must be careful to not step on PE Mode Code functions. With a Transmitter Shutdown Override Mode Code, the PE will set this bit to zero (enabling the bus).

Bit 8: Transmit Inhibit B: W

1=Inhibit (disable) Bus B. This PE uses this bit for temporary settings for Transmitter Shut Mode Code for bus B. The API/user can also set this bit to one to direct the PE to turn-off bus B transmission. The user must be careful to not step on PE Mode Code functions. With a Transmitter Shutdown Override Mode Code, the PE will set this bit to zero (enabling the bus).

Bit 9: Allow Dynamic Bus Control (DBC) Mode Code Control: W

1=Allow Dynamic Bus Control. This bit is set to one by the User to direct the PE to allow Dynamic Bus Control Per the 1553B standard. A setting of zero for this bit (default) will direct the PE to not set the Dynamic Bus Control Bit in the Status Word response.

If this bit is set to one, the PE will respond with the Dynamic Bus Control Accept Status Word Bit set to one on subsequent RT messages AFTER the DBC Mode

Code (so the DBC bit in the Status Word is NOT set in the DBC Mode Code response, but is set in the next RT message response). The PE will not automatically start bus controller operations as this is the application's responsibility. The user's program is responsible for clearing the DBC bit in the response status word for subsequent messages.

Bits 12-23: Response Time Setting: W

RT Response Time Bits (12 Bits): W. These 12 bits are set by the User to direct the PE's Status Response Gap for the respective RT. The value is a 12-bit, 100 ηsecs resolution (40.95 µsecs max). The resolution on the programmed response time is +/- 500 ηsecs. RTs with true 1553A settings should have this set to 40 (4 µsecs dead bus or 6 µsecs per the standard).

> **NOTE:** This value is DEAD BUS TIME and the Official 1553B Status Response is 2 µsecs longer as the official measurement is from mid Parity of the last word of the Command Message (or last word of an RT-RT Transmit message) to mid Sync of the RT's Status Word. This is not an issue: when looking at Intermessage Gaps in CDPs (or Monitor on ***AltaView*** or most other analyzers), the time will show + 2 µsecs.

> **NOTE:** The minimum value for the response time is 4 µsecs (40 ticks), which is 6 µsecs response by the standard. This is necessary due to the fact that RT Validation allows for up to 2 µsecs gaps in data words and required to allow for high-bit/high-word error decoding.

Bit 24-27: Reserved

Bits 28: Inject Status Word Inverted Parity: W

1 = Invert Parity on Status Word. This bit is set to one by the User to direct the PE to invert the Parity (to even) for the Status Word Response. This is used to test BC error handling.

Bits 28: Inject Status Word Manchester Encoding Error on Bit X: W

1 = Invert Parity on Status Word. This bit is set to one by the User to direct the PE to invert the Parity (to even) for the Status Word Response. This is used to test BC error handling.

User Status Word Bits | Last Status Word: W: 0x0008

The third word of an RTCB is a packed word with the 16 LSBs being the most current Status Word transmitted by the RT. The upper 16 bits are available for the user to set Status bits for their application. The 16 MSBs map directly to a 1553 Status Word – one

for one. When the PE is ready to transmit the RT Status Word, it will OR these upper 16 bits into the final value. The user must take great caution in setting these bits as improper settings may cause unpredicted results on the bus. (See RTCB CSR for error injection settings for the Status Word).

Last Command Word: W: 0x000C

The fourth (and last word) of an RTCB is reserved for the PE to place/hold the last Command Word for the RT in the 16 LSBs. The 16 MSBs are reserved for future use. The user can read this register, but not write to it this register. (This register cannot have error injection applied).

RT Filter Table – Extended Memory

The RT Filter table resides in extended PE memory (above 0x00000800 for the channel) and is simple 128-word pointer array.

- The first group of 32 array pointer elements (0-31) corresponds to the RT RECEIVE 0-31 subaddress.
- The second group of 32 array elements (32-63) corresponds to the RT TRANSMIT 0-31 subaddresses.
- The third group of 32 array elements (64-95) corresponds to the RT RECEIVE Mode Code values (0-31). This supports all 1553B standard and reserve Mode Code Receive combinations.
- The third group of 32 array elements (96-127) corresponds to the RT TRANSMIT Mode Code values (0-31). This supports all 1553B standard and reserve Mode Code Transmit combinations.

The RT's Filter Table pointer is addressing a Subaddress/Mode Code Control Block (SA/MC) which further addresses a CDP linked-list and provides settings for legalization of Word Counts and Mode Codes.

> **NOTE:** Unused RT SAs or Mode Code Values must have a common pointer to a garbage RT SA/MCs. Especially in development and test systems, the user will often may have unexpected RT Subaddresses or Mode Codes and these must be trapped. *Null pointers are not allowed for any RT structure.*

Subaddress/Mode Code Control Blocks (SA/MC) – Extended Memory

The SA/MC is a four word structure that contains the head pointer for CDP linked-lists, a legalization setting word for Word Counts and Mode Codes and two reserved words (reserved for future use).

SA/MC-CDP Head Pointer: W: 0x0000

This pointer is set with the address value of the first (or most current) CDP data structure. See CDP Data Structure Section for details on the CDP. The PE will automatically update this value with the next-in-line CDP for processing – so the PE will not update this value until the current RT Response has completed (See the Root RT CSR for settings on CDP address updates for messages received with error or busy – bit 2).

SA/MC Word Count Legalization Settings | Legal Word Count for Mode Code: W: 0x0004

If this SA/MC Filter Table is for a SA, this word is set by the user to designate the legal word counts for the respective subaddress. If this SA/MC Filter Table is for a Mode Code, then this word is provides the word count for the respective Mode Code.

SA/MC Filter Table – SA Legal Word Counts

For Word Count legalization, this word designates legal Command Word – Word Count Values for the RT SA. The bit location directly corresponds to the word count value (so bit 0 is word count zero, and bit 1 is Word Count value one, etc..). The User sets the desired bits to one for the illegal words counts for the SA. If the Word Count is illegal, then the RT will respond with the ME bit set in the Status Word and will not transmit Data Words.

Legalization of Mode Code: W

For Mode Code SA/MC Filter Tables, this allows the user to declare the Mode Code Legal or not. If the Mode Code is illegal, then the RT will respond with the ME bit set in the Status Word. Simply set bits to zero for the Mode Code's that are legal or set bits to 1 if the Mode Code is illegal.

NOTE: The PE will execute legalized Mode Codes per 1553B Notice II. This feature allows for reserve mode codes for 1553A and custom systems (which mean the RT simply responds to the Mode Code with normal Status Word).

Mode Code 0 (Dynamic Bus Control)

If the DBCA control bit is not set in the RT CB Control Register, the PE will send status without DBCA bit set in the status word, but otherwise takes no action (RT rejects dynamic bus control). If the DBCA control bit is set to one in the RT control register, the RT will respond with the DBCA bit set in the status word for messages after the DBC Mode Code. The API will be responsible for switching the PE to BC mode if it detects the DBCA bit set in the status word.

Mode Code 1 (Synchronize)

The PE will respond with status, but otherwise takes no action.

Mode Code 2 (Transmit Status Word)

The PE will respond with status. When this mode code is received, no status bits will change (for example ME, BCR, etc). This MC will be executed regardless of legalization bit settings.

Mode Code 3 (Initiate Self Test)

The PE will respond with status, but otherwise takes no action.

Mode Code 4 (Transmitter Shutdown)

The PE will send the status word and will set the transmit inhibit bit for the bus opposite from what the command was received on. For example if the command was received on bus A, the transmit inhibit bit will be set for bus B.

Mode Code 5 (Override Transmitter Shutdown)

The PE will respond with status and clear the inhibit bit (if set) for the bus opposite of what the command was received on.

Mode Code 6 (Inhibit Terminal Flag)

The PE will respond with status and set the Inhibit Terminal Flag in the RT control word.

Mode Code 7 (Override Inhibit Terminal Flag)

The PE will respond with status and clear the Inhibit Terminal Flag in the RT control word.

Mode Code 8 (Reset RT)

The PE will respond with status, set status word to default value and clear all “OR’ bits in RT CSR.

Mode Code 16 (Transmit Vector Word)

The PE will respond with status and send the transmit vector word from the respective SA/MC Filter Table CDP. If the Clear SR Bit in the RTCB CSR is set, then the PE will set the Service Request bit of the Status Word to zero for the message after the Transmit Vector Word Mode Code.

Mode Code 17 (Synchronize with Data)

The PE will respond with status and store the data word at the respective SA/MC Filter Table CDP. Otherwise no other action is taken.

Mode Code 18 (Transmit Last Command)

The PE will respond with status followed by the data word containing the last command word the RT received. This mode code will not update the last command word stored. This MC will be executed regardless of legalization bit settings.

Mode Code 19 (Transmit BIT Word)

The PE will respond with status followed by the transmit BIT word (the first Data Word of the respective SA/MC Filter Table CDP).

Mode Code 20 (Selected Transmitter Shutdown)

The PE responds with status, but otherwise takes no action.

Mode Code 21 (Override Selected Transmitter Shutdown)

The PE responds with status, but otherwise takes no action.

Two Reserved Words for Future Use: 0x0008/C

These words should be included in the user data structure for SA/MC Filter Tables, but their definitions are reserved for future RT features (such as function generators, time stamps, EFA and Subaddress 17 designations). The user should set these words to 0x00000000.

<~~~>

Sequential Monitor (SM) – Bus Monitor (BM)

The PE provides a 1553 monitor method that captures selected messages to local memory in a chronological order, known as sequential monitoring (SM). This function is typically used for snapshot or data logging applications. The user can control start/stop operations (including data and external triggers), data location in local memory (pointer to CDP link list), 1553 message filtering (to the RT T/R SA level) and monitor status of the current SM operation. SM 1553 is stored in link-list Common Data Packets (know in these paragraphs as SM-CDPs). Please see the CDP Section for details on the data structure. The following paragraphs detail SM functions.

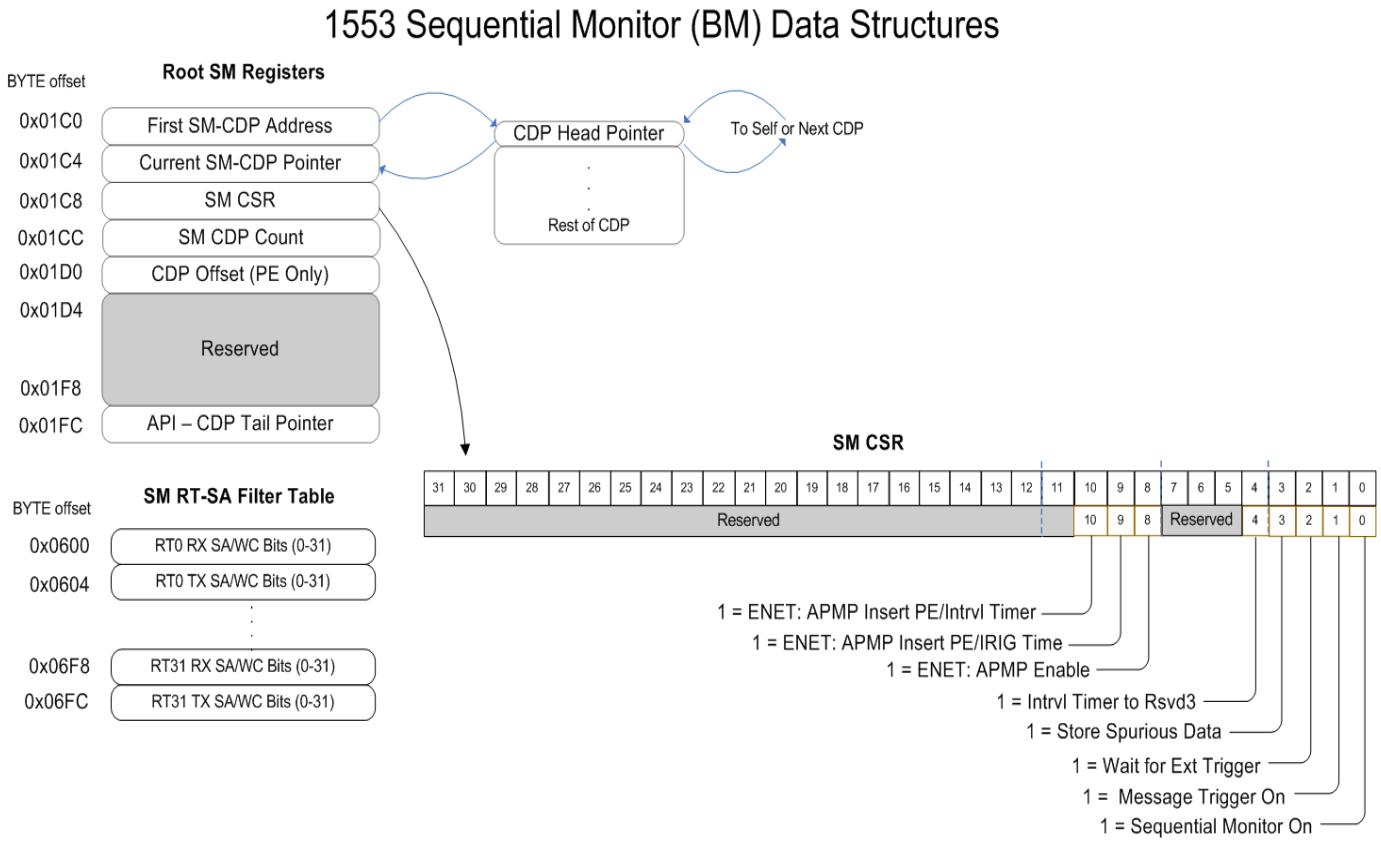


Figure SM-1: Root and Extended SM Data Structures

ENET-1553 APMP

For ENET-1553 products, the SM/BM also controls auto broadcast of CDPs in the Alta Passive Monitor Protocol (APMP) mode (bits 8-10 in the Root SM CSR register).

Root First SM-CDP Address: W: 0x01C0

The User sets this word with the address of the first SM-CDP link data structure for the SM link-list chain. The user may define any number of links for the chain as local memory (and other PE functions being performed and their buffering requirements) will allow.

Root Current SM-CDP Pointer: W: 0x01C4

The PE sets this register with address of the “active” CDP. The active CDP is one currently being processed (or about to be processed) by the PE. This pointer shall be set within 10 µsecs after the last CDP has been completed (filled with the 1553 words, time tags, etc.). This register is valuable in servicing/reading SM-CDP links. The User should not try to process the current CDP.

Root SM CSR: W: 0x01C8

This word is a packed bit data structure and provides the user simple options to enable/disable SM functions and control triggering. The bit controls are described in the following paragraphs.

Bit 0: SM On/Off: W

This bit is set to one by the User to direct the PE to enable (turn on) SM functions. All trigger and filter bit settings should be set prior to setting this bit. The PE will only read this bit at the beginning of a message decode.

Bit 1: Trigger from CDP On/Off: W

This bit is set to one by the User to direct the PE to look for message trigger value (1553 message word values or CDP events such as time bits or status bits) to trigger the storage of message to linked-list SM CDPs. If this function is desired, then the User must properly program CDP Word Offsets 0x0006-7 and the word offset in the Control Register of the first CDP. These locations are discussed in the CDP Requirements Section. Before setting this bit, the First SM-CDP Address must be programmed by the User. The SM ON bit must also be set before the PE will start “looking” for a message trigger.

Bit 2: External Trigger On/Off: W

This bit is set by the User to direct the PE to stall SM capturing until an external trigger has been detected (high to low signal to be defined in the design specification). This bit will delay all 1553 SM capturing and will block the PE from looking for possible Message Trigger (described in the previous paragraph). Before setting this bit, the First SM-CDP Address must be programmed by the user. The SM ON bit must also be set before the PE will start “looking” for an external trigger.

Bit 3: Store Spurious Data On/Off: W

This bit is set by the user to direct the PE to store spurious or garbage data (data that can't be decoded into some type of 1553 message) to a CDP. This is very useful for debugging bus problems.

Bit 4: Store Interval Timer Value in CDP Reserve Word 3: W

This bit is set by the user to direct the PE to the Root PE Interval Timer 24-bit LSB justified value in the CDP Reserve Word 3 (after Intermessage Gap word). See Root Interval Timer (0x004C) register definition in Root PE Register section. Bit 5 of the Interval Timer can be set to allow the user to input an external clock or PPS to track system events, and this BM option allows this value to be copied to the CDP so the user can synchronize CDP 64-bit time and Interval Time (system) events.

Bits 5-7 Reserved

The following bits 8-10 apply only to ENET-1553 Products. They are reserved for other products.

Bit 8: ENET APMP Enable

This bit is set by the user to enable the APMP auto CDP/UDP broadcasting. When set, the ENET-1553 will auto broadcast CDP/UDP packets on Ethernet connection. Please see the AltaAPI manual for more details and usage.

Please see the following bookmarks for IRIG and Interval Timer Formats and Discussion:

Root IRIG Time High & Low: 0x0024/28
Root Interval Timer: 0x004C

Bit 9: ENET APMP Insert PE/IRIG Time

This bit is set by the user to direct the PE to insert the last latched PE and IRIG Time values in to APMP CDP word offsets 0x0C through 0x18. This allows the users Ethernet programs listening to APMP packets to have PE+IRIG absolute time along with the normal CDP Time High/Low relative time (to time sync data capture). This bit must be set with Bit 8. Please see the AltaAPI manual for more details and usage. The time values would be the same as setting the Read IRIG control bit in the Root PE Control Word (0x0000).

Bit 10: ENET APMP Insert PE/Interval Timer

This bit is set by the user to direct the PE to insert the last latched PE and Interval Timer (user PPS or clock) in to APMP CDP word offsets 0x0C through 0x18. Ethernet programs listening to APMP packets to have PE+ Interval/PPS

absolute time along with the normal CDP Time High/Low relative time (to time sync data capture). This bit must be set with Bit 8. Please see the AltaAPI manual for more details and usage. The time values would be the same as setting the Read Time control bit in the Root PE Interval Timer register (0x004C).

Root SM-CDP Count Register (Sequence Number): W: 0x01CC

This register is initialized by the user to provide the starting value for SM-CDP counting (sequence number in CDPs). This would increments with each new CDP and is copied into the respective CDP's 2nd (0x0004 offset) as a sequence number (that can be tracked for skipped messages). It is recommend the user initialize this register to zero prior to starting the SM operations. If desired, the user can read this register to track the SM operations.

Reserved: 0x01D0-0x01F4 Reserved for PE processing. User should not write to these locations.

API – CDP End Pointer: 0x01F8

The API stores the address of the last CDP in this register.

API – CDP Tail Pointer: 0x01FC

The API stores the "tail" pointer value in this word. This is a pointer to the last CDP read from the monitor buffer. The API uses the Current SM-CDP Pointer register (0x01C4) as the "head" pointer to indicate where the PE is in the buffer. The API will periodically read this value to see if new messages have been stored in the buffer. If so, the API will start at the tail pointer and read messages until the tail pointer equals the head pointer.

Root RT/SA Filter Words: W: 0x0600 to 0x06FC

These 64 words are set by the user and direct the PE to filter designated 1553 message by the RT Address (5 bits), T/R Bit (single bit) and Subaddress (5 bits). There are 64 Filter Words, two words for each RT and the bits correspond to the SA T or SA R possibilities. Setting a bit to one will allow the corresponding RT address, T/R, SA address (read from the first command word of a message) to be captured in an SM-CDP. Each of these 32 bit combinations are mapped to a corresponding bit of a 32-bit word where the LSB of each word corresponds to the RT SA for the T or R bit: for example, bit zero of the 32-bit word corresponds to RT SA zero, bit one corresponds to RT SA bit one, etc. Example: for all messages to be capture, then set all bits to one. **Upon power-up/reset the PE sets these filter bits all to one to capture all messages by default.**

> **NOTE:** In the case of an RTRT message, filtering will only be performed on CMD1. If the appropriate filter bit is not set for CMD1, the entire message will be discarded.

<~~~>

AltaCore-1553 Common Data Packet (CDP)

Common Data Packet - Extended Memory

The CDP is a data structure that contains all 1553 bus words (Command, Status and Data Words), timing words (response times and general time stamp), control and status words that pertain to a decoded message. The same CDP data structure is used for BC, RT and Monitor operations (with only minor variations of control/status words within the CDP header words). The following paragraphs detail the CDP data structure.

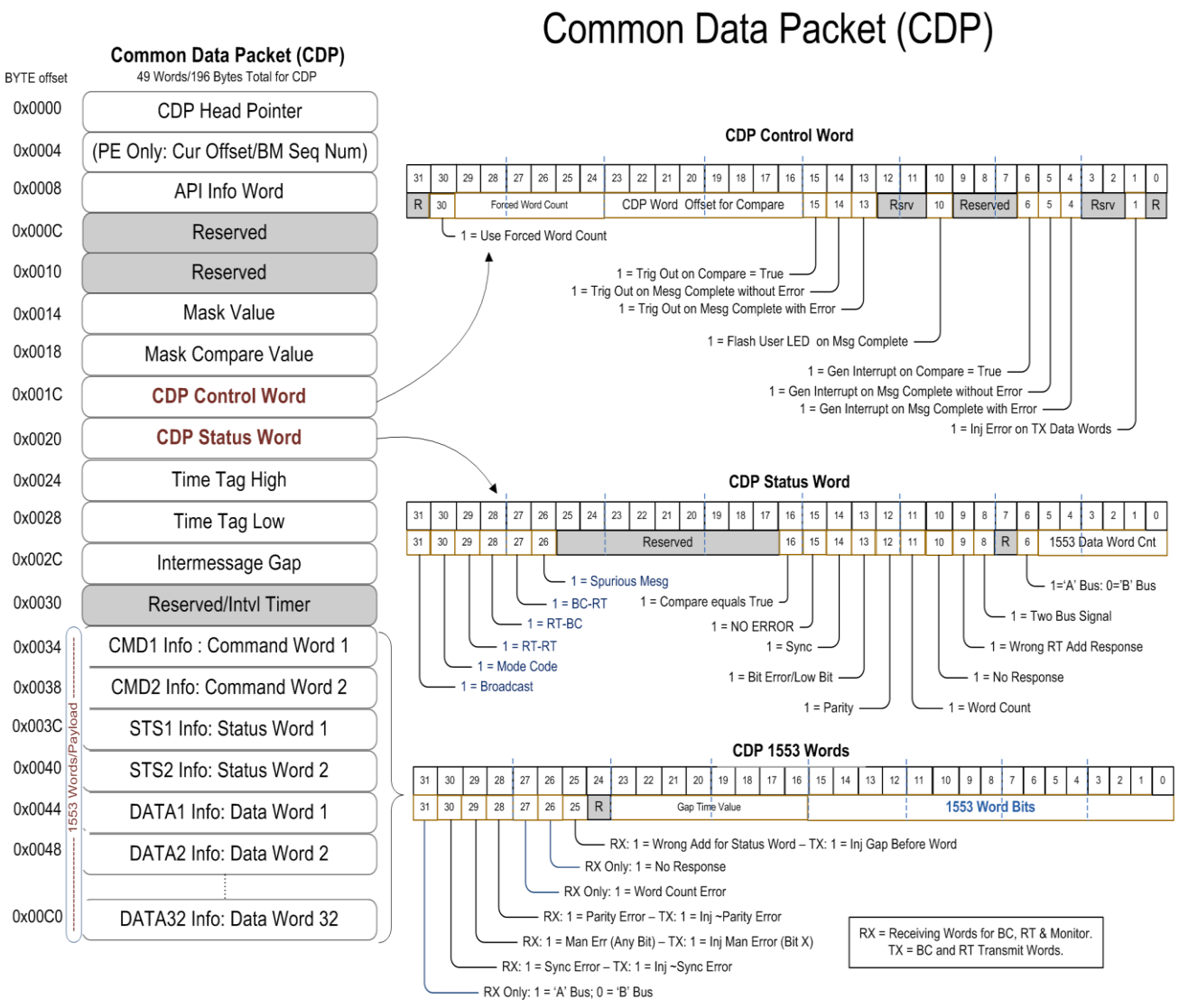


Figure CDP-1: CDP Data Structures

CDP Update Status – How to Sync Your Application to Fresh CDP Data

There are a couple key methods for the user application to make sure they are reading fresh CDP data:

- The easiest method to knowing when a CDP is completed and fresh is to set an Interrupt Bit in the CDP Control Word and then receive a Hardware Interrupt or Poll the Interrupt queue and receive the information that the CDP has been updated or completed. There are many example programs that show this method.
- When the PE starts processing or updating a CDP, the CDP Status Word (CDP Offset 0x0020) is set to 0xFFFFFFFF. The CDP Status Word is the last word updated in the CDP processing sequence. So the user's application can read the CDP Status Word, and if the value is 0xFFFFFFFF, then the data should not be considered valid until the CDP Status Word is NOT 0xFFFFFFFF.
- To know if the CDP data is fresh (different than valid), the user can read the Time Low register and look for a change in value. This is a 20 nSec clock, so the value will always change when a CDP is updated. This is the easiest method for applications where there is only one CDP for each BC, RT or BM buffer.
    - For multiple CDPs in a BC, RT or BM Chain, the user could also read the Current CDP Pointer in the BCCB, RT SA/Mode Control or Root BM register. The user would need to map and know the chain CDP offset values and then determine if the change in Current value matches their desired CDP link. This method is advanced and not recommended for most applications.

Next CDP Pointer : W: 0x00000000

This word is set by the user with the address of Next CDP Pointer of the next CDP (to create a linked-list of CDPs). If a single buffer is desired, then this value should point (address) to itself. For double buffering, the two CDPs point to each other, etc…Users can be creative in having multiple RT SAs or BCCB point to the same CDP link list (or single CDP) to create wrap or bridge conditions (bridging data between RTs or BCs). The user can change this pointer value on the fly/real-time, but great care must be taken not to change the value while the current BC or RT operation is being processed by the PE. This value should not be changed within 10 µsecs of the CDP being processed. The user can read the BCCB or RT SA/MC Control Block, SM or Playback Current Pointer to locate the current position of the CDP linked-list and decide if it is safe to change this pointer on the fly.

Current CDP Offset (PE Only)/BM Counter Sequence Number: 0x0004

This word is read only for the user. For SM/BM operations, this is a 32-bit sequence number copied from the Root SM/BM Sequence Number register at 0x01CC (see above in SM/BM section). This register is reserved in BC and RT operations and should NOT be written to by the user/API (reserved for PE operations).

API Info Words: 0x0008-0x0010

This word is reserved for the ***AltaAPI*** for tracking CPD functions. This read only for the user.

Mask Value: W: 0x0014

This word is set by the user as a logical AND mask for checking a CDP word value. The data word offset of the CDP for masking is set in the CDP Control Word. A value of zero in this location will cause the Compare function to be bypassed.

Mask Compare Value: W: 0x0018

The user sets this word with the bit values to be compared to (after the AND mask operation) to decide if the compare is TRUE or FALSE. The pseudo logic for the compare function is shown below:

```
If ((Data & MASK Value) == Mask Compare Value) Then
        Compare = TRUE
Else
        Compare = FALSE
```

CDP Control Register: W: 0x001C

This word is a packed bit data structure that direct PE operations specific to this CDP. The following bits are defined by the user.

Bit 0: Reserved

Bit 1: Injection Error on Transmit Message On/Off: W

1=Injection Error on Transmit Message. This bit is set to one by the user to direct the PE to allow error injection on the Data Words of a transmit message (BC or RT). If this bit is off, then the PE will not injection errors even if their respective encoding bits are set. The error encoding bits are the 16 MSB of every 32-bit Data Word of a CDP. Subsequent paragraphs of this section detail these bits.

Command and Status Word error injection is encoded by their respective BC or RT Control Blocks. Command and Status Words in CDPs are monitor values only.

Bits 2-3: Reserved

Bit 4: Gen Interrupt on Message Complete with Error: W

1=Interrupt on Message Complete with Error. This bit is set by the user to direct the PE to generate an interrupt with then 1553 message is completely decoded and filled in to this CDP and a 1553 protocol error was detected. The error type can be found in the CDP Status word and in the error status words at the end of the CDP (if this error status word option was selected in the bit described above).

> **NOTE**: The interrupt queue packet will be complete and the interrupt pending bit set in the Global Root Interrupt Pending register (and hardware interrupt generated if selected in Primary Root CSR) within 10-15 µsecs of the CDP being completed.
>
> **NOTE**: This bit may be set in conjunction with Interrupt on Message Complete without Error (the two bits are mutually exclusive). This allows the user application to be signaled for any message complete condition.

Bit 5: Gen Interrupt on Message Complete Without Error: W

1=Interrupt on Message Complete without Error. This bit is set by the user to direct the PE to generate an interrupt with then 1553 message is completely decoded and filled in to this CDP and no 1553 protocol error was detected.

> **NOTE:** The interrupt queue packet will be complete and the interrupt pending bit set in the Global Root Interrupt Pending register (and hardware interrupt generated if selected in Primary Root CSR) within 10-15 µsecs of the CDP being completed.
>
> **NOTE:** This bit may be set in conjunction with Interrupt on Message Complete with Error (the two bits are mutually exclusive). This allows the user application to be signaled for any message complete condition.

Bit 6: Gen Interrupt of Word Mask On/Off: W

1=Interrupt on Word Mask: This bit is set by the user to direct the PE to generate an interrupt when a word mask condition is met. If this function is desired, then the user must properly program CDP offsets 0x0014 & 0x0018 (these words are described later in this section).

Bit 7-9: Reserved

Bit 10: Flash LED on Message Complete: W

1 = Flash LED for 1553 Activity. This bit is set to one by the user to direct the LED to generate a 250 msec flash an LED when the CDP is complete.

Bit 11-12: Reserved

Bit 13: Trigger Out on Message Complete with Error: W

1=Output Trigger on Error Detected. This bit is set to one by the user to direct the PE to generate an output trigger when a 1553 error is detected. This would be a 1 µsec high to low-going pulse.

Bit 14: Trigger Out on Message Complete Without Error: W

1=Output Trigger on Message Complete – No Error: This bit is set to one by the user to direct the PE to generate an output trigger when the respective 1553 is complete with no error detected. This would be a 1 µsec high to low-going pulse. I/O requirements are described in Section 1.

Bit 15: Trigger Out on Mask/Value Positive: W

1=Output Trigger on Word Mask. This bit is set to one by the user to direct the PE generate an output trigger when a word mask condition is met. If this function is desired, then the user must properly program CDP bit offsets 0x0014 and 0x0018 (these words are described later in this section).

Bits 16-23: Word Offset for Compare Function: W

CDP Compare Word Offset (8 Bits). These bits are set by the user for BC Branching, SM Message Triggering, Interrupt on Word Mask or Output Trigger on Word Mask. These eight bits provide the 32-bit offset in the CDP that will have the Mask and Value Words (CDP offsets 0x0014 & 0x0018) masked/compared to. The eight bits may be used to “point” to any word in the CDP, and thus, the user can compare for values of pointers, time bits, 1553 word values, error or status values.

Bit 24-29: Forced Word Count: W

These bits are set by the user to force the PE to transmit the programmed value number of words (0-65). This function is for injecting message-level word count errors for testing. For words over 32 count, the last Data Word value of the CDP will be repeated. Bit 30 and bit 1 must be set to enable this function.

Bit 30: Enable Force Word Count: W

1=Use Forced Word Count value for Data Word count. When this bit is set to one, the PE will transmit the number of Data Words specified in bits 24-29. This bit is ANDed with bit 1 to ensure the user wants to inject errors.

Bit 30: Reserved

CDP Status Word: 0x0020

This read only word is a packed bit structure that provides the user CDP execution status information of the 1553 message (this is NOT the 1553 Status Word). This Status Word is updated for message types, including transmit message. The bit structures are described in the next paragraphs.

Bits 0-5: Actual Data Word Count

These six bits are set by the PE when the CDP has been completed/filled and indicates an integer value of the actual number of 1553 Data Words of the message.

Bit 6: Bus Indicator

1=Bus A; 0=Bus B. This bit is set by the PE to one if the 1553 message was decoded on Bus A and set to zero if the message was decoded on Bus B.

Bit 7: Reserved

Error Indicator Bits (8 Bits)

These eight bits are set by the PE to indicate error conditions (or non error condition) of the 1553 message. These bits are set in addition to CDP Error Code Words (if Error Code Enable has been selected for receive/monitor messages in the CDP Control Word). Generally, embedded BC or RT applications do not need detailed error code information and do not want the overhead of the Error Code words; and thus, these bits are satisfactory for error detection. The error indicator bits are explained in the following paragraphs.

> **NOTE:** Error conditions can vary widely from bus to bus or test run to test run (from impendence/cable changes, timing, loading, etc.). Per the 1553 specification, a message is considered invalid with any error. Often, multiple error indicators are indicative of badly terminated, shorted or poorly constructed (or illegally constructed) bus system.

Bit 8: Two Bus

1 =Two Bus. This bit is set by the PE indicate activity has been detected on both busses in an illegal condition (simultaneously of RT Response on Opposite Bus).

Bit 9: Wrong RT Address Response

1=Wrong RT Address Response. This bit is set by the PE to indicate that the received /monitored Status Word has incorrect RT Address (5 MSB of the 1553 word).

Bit 10: No Response

1=No Response (NORESP). This bit is set by the PE to indicate that an RT No Response Condition has been detected (or BC Response Time-Out).

Bit 11: Word Count

1=Word Count. This bit is set by the PE to indicate the 1553 message has the improper number of data words per the respective Command Word. The user can read the 6 LSB bits of this word to obtain the actual word count.

Bit 12: Parity

1=Parity. This bit is set by the PE to indicate that a parity error was detected on a 1553 word in the message. The individual 1553 word encoding bits would need to be checked to determine the word with the error.

Bit 13: Bit Error/Low Bit

1=Bit Error. This bit is set by the PE to indicate a bit encoding error has been detected (a bad Manchester II encoding), or an improper number of bits has been detected. The individual 1553 word encoding bits would need to be checked to determine the word with the error.

Bit 14: Bad Sync Detected

1=Sync Error. This bit is set by the PE to indicate that a sync error has been detected in a word of the current message. The individual 1553 word encoding bits would need to be checked to determine the word with the error.

Bit 15: No Error Detected/Message Complete

1=No Error/Message Complete. This bit is set by the PE to indicate that the current 1553 message has completed without error.

Bit 16: Compare Equals True

1=Compare Equals True. This bit is set by the PE to indicate that the compare function returned a TRUE result.

Bits 17-25: Reserved

Message Type Indicator Bits (6 Bits)

The CDP Status Word provides 6 bits to indicate the 1553 message type for this CDP. These bits are detailed in the following paragraphs.

Bit 26: Spurious Data

1=Spurious Data. This bit is set by the PE to indicate that the 1553 message was too badly garbled for decoding (meaning a 1553 bit encoding or message structure was invalid and could not be decoded). The main reason for this condition is badly terminated, shorted or poorly constructed (or illegally

constructed) bus system, two computers transmitting at the same time or bit/sync/gap error injection in the middle of a message. Only the first 32 words of spurious data will be stored in the CDP (1553 payload area) and other non-message words (words that cannot be decoded into a valid message) will be dropped.

Bit 27: BC-RT

1=BC-RT. This bit is set by the PE to indicate that the CDP 1553 message type if a BC-RT Message. The user will need to read the Command Word of the CDP for more information about the 1553 message.

Bit 28: RT-BC

1=RT-BC. This bit is set by the PE to indicate that the CDP 1553 message type if a RT-BC Message. The user will need to read the Command Word of the CDP for more information about the 1553 message.

Bit 29: RT-RT

1=RT-RT. This bit is set by the PE to indicate that the CDP 1553 message type if a RT-RT Message. The user will need to read the Command Words of the CDP for more information about the 1553 message. This bit is also set for the Broadcast RT-RT messages.

Bit 30: Mode Code

1=Mode Code. This bit is set by the PE to indicate that the CDP 1553 message type if a Mode Code Message. The user will need to read the Command Word of the CDP for more information about the 1553 message.

Bit 31: Broadcast

1=Broadcast. This bit is set by the PE to indicate that the CDP 1553 message type is any type of Broadcast Message (including BC-RT, Mode Code or RT-RT). The user will need to read the Command Word(s) of the CDP for more information about the 1553 message.

Time Tag High/Time Tag Low: 0x0024 & 0x0028

These two words make-up a 64-bit, 50 MHz (20 ηsec) time stamp for the CDP. The time stamp is latched at mid parity of the 1st Command Word of the message (accuracy is within 100 ηsec). These time words are updated for all message types, including transmit CDPs.

Intermessage Gap: 0x002C

This word is a 32-bit, 100 ηsec value of the intermessage dead bus or idle gap time from the previous message. This time represents “dead bus” time the end of the previous message’s last word (previous CDP) to the beginning of the first Command

Word of this CDPs message. This is dead bus/idle time (not mid parity to mid sync timing). This is accurate to within 500 ηsecs.

Reserved/Interval Timer: 0x0030

This word is usually reserved (zero) for most applications. The user does have an option to have the Root PE Interval Timer (0x004C) stored in this location for Sequential Monitor operations only (by setting the appropriate bit in the BM Control Word). Please see Root PE Interval Timer and Sequential Monitor BM Control Word settings for more details.

CDP 1553 Words (Message Payload): 0x0034-0x00C0

The following 36 words of the CDP are the 1553 message words (payload). Each word is a 32-bit structure where the 16 MSBs are an error encoding or decoding (depending if the message was a transmitted or received word) and the lower 16 LSBs are the actual bits of the 1553 word (bits 4-19 of a 1553 word format). Parity and sync bits are not provided (stripped off), but a parity or sync error indicator bit would be set if the values are not proper. Figure CDP-1 shows the make-up of these words (regardless of 1553 word type).

The first two words are the two possible Command Words (CMD1/CMD2). The next two words are the two possible Status Words (STS1/STS2). These words are monitor values ONLY and are not transmitted in BC or RT functions.

The subsequent 32 of the 1553 payload are the 1-32 1553 Data Words of the 1553 message. These words can be receive/monitor words or user/API set transmit words depending on the PE's RT or BC function of operation. Here are the rules for the 32 1553 Data Words (See Figure CDP-1 for details):

- If the PE (card device) is set to be an RT or BC, and the message type requires transmitting data words, then the 1-32 1553 Data Words are set by the user/API for transmission.
    - With the PE set to be an active RT, this would include RT-BC (including Transmit Mode Codes with Data Words) and RT-RT Transmit Message Types.
    - With the PE set to be a BC, this would be for BC-RT messages, including Broadcast and Transmit Mode Code with Data Words.
- If the PE is set to be a Bus Monitor, RT Map Monitor, or an RT or BC message where the PE would receive 1-32 Data Words, then these words are set by the PE with the receive/monitor values.
    - The PE set as a Bus Monitor or RT Map Monitor: These words are always monitor values.

- The PE Set as a BC: These are monitor/receive values for RT-BC messages including Receive Mode Code with Data Word, and RT-RT messages.
- The PE set as an RT: These words are monitor/receive values for BC-RT messages including Transmit Mode Codes with the receiving part of an RT-RT messages.
    - Broadcast BC-RT (including Transmit Mode Codes) or Broadcast RT-RT Messages have the 1553 Data Words received in only RT 31 Subaddress/Mode Code CDPs.

The following paragraphs provide further details on the CDP 1553 Payload Words. The next couple sections detail the upper 16 MSB of each 32-bit word depending if the payload word was a receive/monitor or transmit word (only 1553 Data Words could be a transmit word). The lower 16 LSB are always the 1553 data bit values.

Info RX Bit Values (16 MSB) for 1553 Words

For receive data words and for CMD and STS words, the upper 16 bits are status bits with the following bit values.

Bits 16-23: Gap Time Value

These 8 bits are set by the PE with the 100 ηsec dead bus time that preceded the word.

Bit 24: Reserved

Bit 25: Wrong Status Address

This bit is to one by the PE is the word is a decoded as a Status Word with the wrong RT Address value.

Bit 26: No Response

This bit is set to one by the PE is a No Response condition (the absence of the Status Word Response from an RT) has been detected.

Bit 27: Word Count Error

This bit is set by the PE is an improper word count as been detected for the decoded message. For word counts greater than 32, only the Sequential Monitor function will capture the extra data words (typically flagged a spurious packet). Please see the Sequential Monitor Section for details.

Bit 28: Parity Error

This bit is set to one by the PE is a Parity Error was decoded for this word.

Bit 29: Manchester Error

This bit is set to one by the PE if the decoded word had a bad Bi-Phase, Manchester bit detected in the word.

Bit 30: Sync Error

This bit is set to one by the PE if a bad/improper/inverted sync was decoded for the word.

Bit 31: 1=Bus A, 0=Bus B

This bit is set to one by the PE is the word was decoded on Bus A, and this bit is set to zero if the decoded word was from Bus B.

Info TX Bit Values (16 MSB) for 1553 Words

For transmit data words, the upper 16 bits are encoding bits (used for optional error injection) with the following values. (Word Counts errors can be injected by the CDP Control Word – please see that paragraph for details.) If errors are not to be injected, then set all these bit values to zero.

Bits 16-23: Gap Time Value

These 8 bits are set by the user to inject the idle gap (dead bus) period prior to this word. This value must be used in conjunction with Bit 25. The resolution is 100 ηsec ticks.

Bit 24: Reserved

Bit 25: Inj Gap Before Word

This bit is set to one by the user to turn on the gap period programmed in Bits 16-23.

Bit 28: Parity Error

This bit is set to one by the user to direct the PE to transmit even Parity for this word.

Bit 29: Manchester Error

This bit is set to one by the user to direct the PE to inject a bad Bi-Phase, Manchester encoding on bit 3 (first bit after sync) of this word.

> **NOTE:** If bit 29 and bit 28 are set simultaneously, then a Manchester error will be injected in the parity bit. This provides an additional bit position for Bi-Phase error injection.

Bit 30: Sync Error

This bit is set to one by the user to direct the PE transmit a positive-going (high then low) sync pattern for this word. Sync Errors for Command and Status Words are encoded in their respective BCCB or RT Control Block structures.

Bit 31: Reserved

CMD1 Info | CMD1 Value : 0x0034

This word is set by the PE with first decoded Command Word for the CDP message. The 1553 word bits are in the 16 LSBs and the decoding information (as explained in the previous paragraphs) are in the 16 MSBs. For a valid message, this word should always have a value not equal to 0xFFFFFFFF. A value of 0xFFFFFFFF signifies that this word was not decoded and may signify a spurious packet of noise (or a group of words that was not decode-able as a valid 1553 message).

CMD2 Info | CMD2 Value : 0x0038

This word is set by the PE with the second decoded Command Word (usually from an RT-RT message) for the CDP message. The 1553 word bits are in the 16 LSBs and the decoding information (as explained in the previous paragraphs) are in the 16 MSBs. A value of 0xFFFFFFFF signifies that this word was not decoded, which is typical/normal for non RT-RT messages, or may signify a spurious packet of noise (or a group of words that was not decode-able as a valid 1553 message).

STS1 Info | STS1 Value: 0x003C

This word is set by the PE with first decoded Status Word for the CDP message. For RT-RT Messages, this is the TX Status Word. The 1553 word bits are in the 16 LSBs and the decoding information (as explained in the previous paragraphs) are in the 16 MSBs. For a valid message, this word should always have a value not equal to 0xFFFFFFFF. A value of 0xFFFFFFFF signifies that this word was not decoded and may signify a spurious packet of noise (or a group of words that could not be decoded as a valid 1553 message).

STS2 Info | STS2 Value: 0x0040

This word is set by the PE with decoded Command Word for the CDP message. The 1553 word bits are in the 16 LSBs and the decoding information (as explained in the previous paragraphs) are in the 16 MSBs. A value of 0xFFFFFFFF signifies that this word was not decoded, which is typical/normal for non RT-RT messages, or may signify a spurious packet of noise (or a group of words that could not be decoded as a valid 1553 message).

Data Info N | Data Word N: 0x0044/BC

These words hold the word values and encoding/decoding information for the 1553 Data Words of the CDP. As mentioned in previous paragraphs, if the Data Word is part of a receive CDP, then the 16 MSBs have status decode bits, and if the Data Word is part of a transmit CDP, then the 16 MSBs are encoding bits (for optional error injection). Please see the discussion in previous paragraphs for the definitions of the upper 16 MSBs.

The lower 16 LSBs contain the Data Word bit values and correspond one for one with the 1553 word format (bit times 4-19).

ENET APMP CDP Format: PE + IRIG or Interval Timer Inserts

For ENET-1553 product only, the CDP has a different format if the user selects the APMP Insert PE/Intrvl or PE/IRIG Time options in the Root SM CSR registers. These time formats are the SAME as if IRIG time is read from the Root PE Control register (for PE + IRIG Time) or the Read Timer in the Root Interval Timer Control Register. All other CDP values are the same. Please see the following bookmarks:

Root IRIG Time High & Low: 0x0024/28

Root Interval Timer: 0x004C

The following diagram shows the CDP with the optional Time Values in CDP Offsets **0x0008 to 0x001C** (again these time values match the description in the bookmarks above).

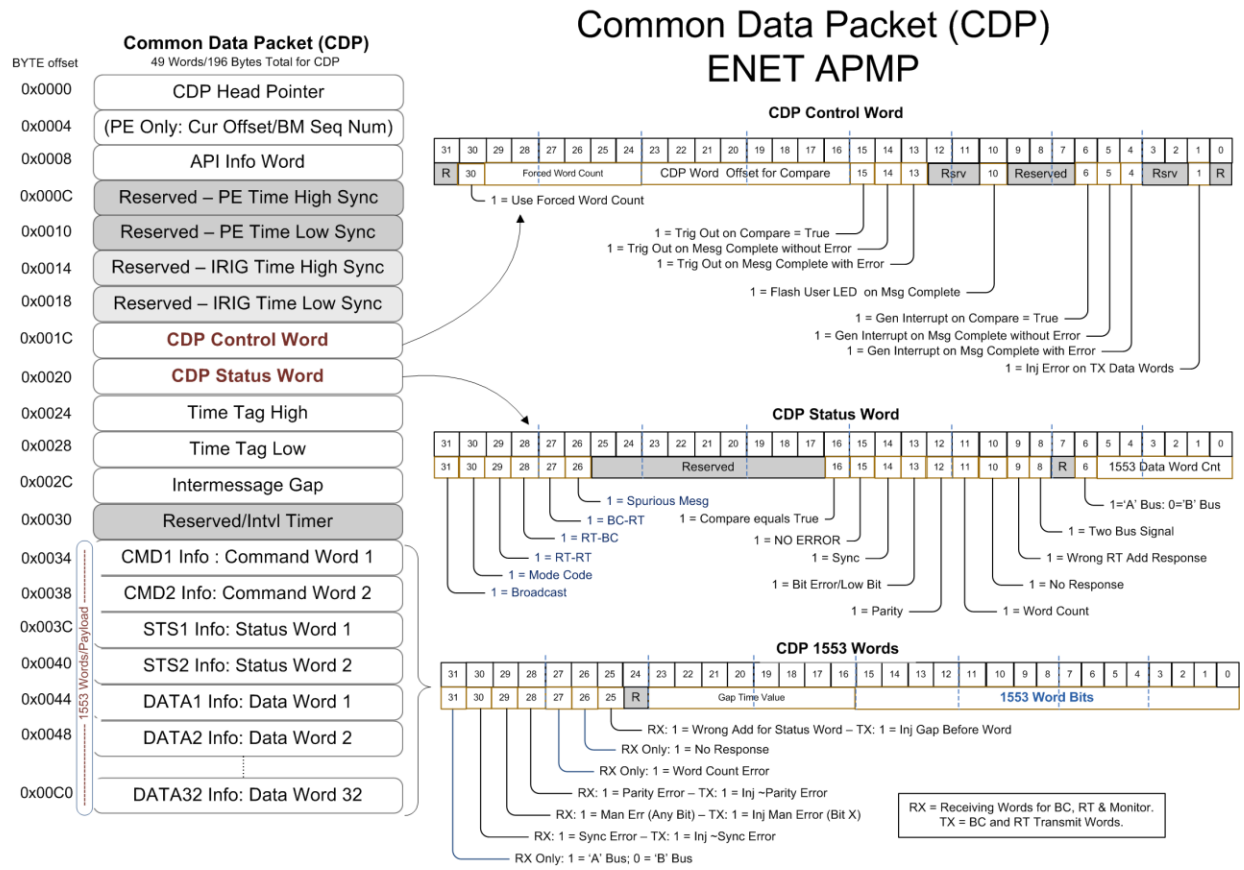


Figure CDP-2: ENET APMP CDP Structure

<~~~>

Interrupt Functions

The PE provides a linked-list interrupt queue and status registers for logging interrupt events set by the user in the various BC, RT, SM, PB and SG data structures. These flag events are trapped in an interrupt queue linked-list data structure to store their occurrence. The user will either poll or receive a hardware interrupt to trigger the reading of this queue to retrieve information as to what event was capture.

The following paragraphs and Figure Int-1 detail these interrupt queue data structures.

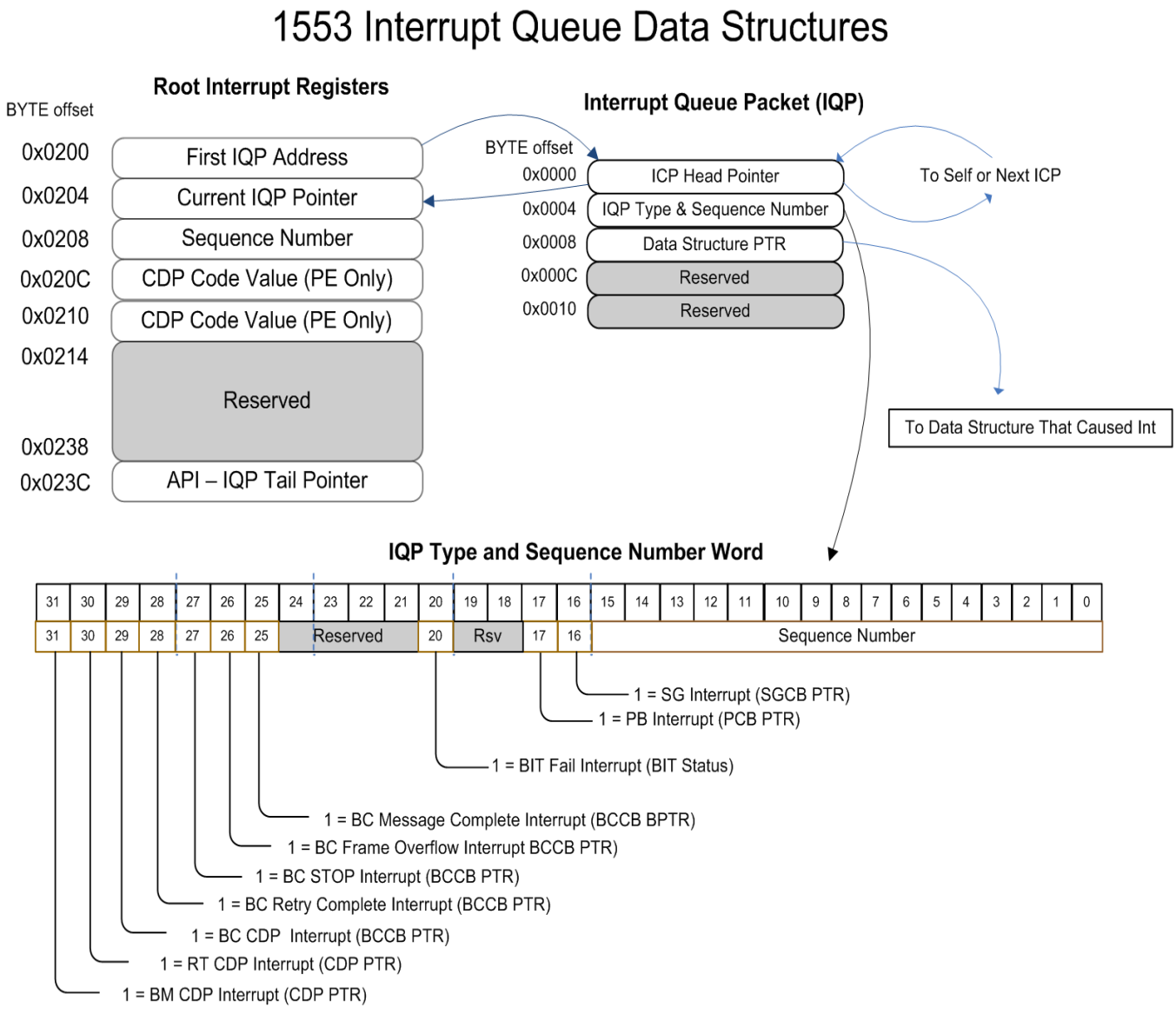


Figure Int-1: Interrupt Data Structures

Interrupt events are serviced by the user's application in two methods: a hardware interrupt signal to the system with an OS Interrupt Service Routine (ISR in a driver or user level setup) or software polling through some type of host application process timer (or aperiodically by the application). For either method, the user should read about the interrupt control/pending/latch bits in the Global Registers and Root PE Registers Sections. These bits control hardware signaling/clearing of the interrupt line to the host backplane/system and provide pending information for channel/event identification.

Root Starting Interrupt Queue Packet (IQP) Address: W: 0x0200

The user sets this value with the address of the first IQP link of the chain. If this value is zero, then the PE will not process interrupts.

> **NOTE**: The user can stop interrupt processing by setting this value to zero. Interrupt conditions will not be processed if this value is zero. The user should initialize all interrupt structures prior to setting this value (turning on) to the first IQP address to start/re-start interrupt processing.

Root Current IQP Address: 0x0204

The user can read this value to see the current location of the active IQP. The user should not process this IQP, but the previous unread IQP in the link-list.

Root Interrupt Sequence Number (16 LSB Bits): W: 0x0208

This register is set by the user with the starting sequence count, which is probably zero. Only the 16 LSBs are used. The PE will store the value of next incremented value back at this location.

Root CDP Code Value (PE Only) – Reserved: 0x020C

Root CDP Code Value (PE Only) – Reserved: 0x0210

The user should not write to these registers. They are reserved for PE interrupt temporary processing.

Reserved: 0x0214-0x0238

API – IQP Tail Pointer: 0x023C

The API uses this word to store the Interrupt Queue "tail" pointer, which points to the next Interrupt Queue entry to be processed by the API. The API reads the Current IQP Address (0x0204) to get the "head" pointer. The API then reads Interrupt Queue entries until the tail pointer equals the head pointer.

Interrupt Queue Packets (IQP) – Extended Memory

PE process events that the user set for detection various are queued for user application access. The queue is a link-list of packed word data structures (Interrupt Queue Packets) that provides status to what event caused the interrupt, a running

counter of interrupt events (sequence number), the address (pointer) of the data structure that triggered the event and two reserved words. The IQP is detailed in the following paragraphs.

> **NOTE:** Interrupts will be posted to this queue structure within 10-15 µsecs with Full Function, Multi-Channel RT-RT messages as a worst case condition.

IQP Head Pointer: W: 0x0000

The first word of the IQP data structure is the address of the next IQP of the chain. The user determines how many lQP links are required to allow interrupts events to synchronize user application requirements. Since this is a circular chain, then length should be at least 2 links and it is recommended the user default to 100 links.

Interrupt Type (16 Bits) | Interrupt Sequence Number (16 Bits): W: 0x0001

This register is a packed data structure where the MSB 16 bits are the Interrupt Code derived from the respective PE function (BC, RT, SM, PB, SG) that caused the interrupt (this allows the user application to prepare the appropriate data structure). The bit codes are detailed in the following paragraphs. The LSB 16 bits are a Sequence Number that simply increments to allow the user to error-trap dropped interrupt events. The user should initialization this word to 0XFFFFFFFF.

Bit 16: SG Interrupt (SGCB PTR)

This bit is set by the PE to signify that the Data Structure Pointer is form a Signal Generator Control Block. This option is set with bit 1 of the SGCB CSR.

Bit 17: PB Interrupt (PBCB PTR)

This bit is set by the PE to signify that the Data Structure Pointer is for a Playback Control Block. This bit is set with bit 6 of a PCB Control Word.

Bits 18-19: Reserved

Bit 20: BIT Fail Interrupt (BIT Status)

This bit is set by the PE to signify that the Data Structure Pointer is for the BIT Status Word. No Data Structure Pointer value is given/needed.

Bit 21: Interval Timer

This bit is set by the PE to signify that the Interval Timer has reached is max time value. This option is set with bit 5 of the Interval Timer register (0x004C). No Data Structure Pointer value is given/needed.

Bits 21-24: Reserved

Bit 25: BC Message Complete Interrupt (BCCB Pointer)

This bit is set by the PE to signify that the Data Structure Pointer is for a BC Control Block that had a message complete (bit 8 of BCCB CSR).

Bit 26: BC Frame Overflow Interrupt (BCCB Pointer)

This bit is set by the PE to signify that the Data Structure Pointer is for the BC Control Block that had a Frame Overflow (Frame Time Exceeded – set by bit 13 of the Root BC CSR).

Bit 27: BC Stop Interrupt (BCCB Pointer)

This bit is set by the PE to signify that the Data Structure Pointer is for the last BC Control Block executed (bit 14 of the Root BC CSR).

Bit 28: BC Retry Complete Interrupt (BCCB Pointer)

This bit is set by the PE to signify that the Data Structure Pointer is for the BC Control Block that completed a retry sequence (bit 15 of the Root BC CSR).

Bit 29: BC CDP Interrupt (CDP Pointer)

This bit is set by the PE to signify that the Data Structure Pointer is for the CDP that caused the interrupt (the user would need to read the CDP Control Word and check values to determine the cause of the interrupt).

Bit 29: RT CDP Interrupt (CDP Pointer)

This bit is set by the PE to signify that the Data Structure Pointer is for the CDP that caused the interrupt (the user would need to read the CDP Control Word and check CDP values to determine the cause of the interrupt).

Bit 29: BM CDP Interrupt (CDP Pointer)

This bit is set by the PE to signify that the Data Structure Pointer is for the CDP that caused the interrupt (the user would need to read the CDP Control Word and check CDP values to determine the cause of the interrupt).

Data Structure Pointer: W: 0x0002

This word is set by the PE with the address of the data structure that caused the interrupt event. The bits described above provide the coding of this word.

Reserved Words: 0x0003/04

<~~~~~>

Revision Information

Date	Rev	Description
9/24/08	E	Minor Typos and Offset Corrections
12/09/08	F	Corrected BCCB Control Word, Bit Zero Stop on Error Description. Corrected CDP Status Bit Error Definitions. Added Absolute Timing Playback and Timing Control.
2/9/09	F1	Clarified PB IMG Timing and Corrected Minor Grammatical Errors
3/12/09	F2	Minor Typos Corrected. Added BCCB Message Type "Delay Only", Signal Capture Triggers, DDISC2 RS-485 Trigger Enabled
8/20/09	F3	1. Corrected Description on "Delay Only" for BCCB: removed "(no subframing)." 2. Added BCCB CDP-Branch. Added Global Time Trigger. 3. Changed Address. 4. Added verbiage to better explain Frame Scheduling for first message of a frame. 5. Added verbiage to better explain BC External Trigger. 6. Corrected Verbiage to not allow branching in Subframes. 7. Correct BM/BC RT Response Time out to maximum of 32 µsecs. 8. Added Interval Timer Information.
10/4/09	F4	Added to RT Root CSR Mode Code Sync Without Data Int and Trigger Options
11/19/09	F5	Fixed RT-2 Figure – Filter Table Memory Offsets Where Wrong
2/18/2011	F6	Added BCCB Address Branch. Added IRIG Clarification that requires +1 Second User/API adjustment and listed IRIG Decode Formats Supported. Added Interval Timer Bits 2 & 5 for using an external clock or PPS to sync user time events with PE 64-bit clock. Added Bit 4 to BM Control Word to Copy Interval Timer to CDP Rsvd3.
3/14/2012	F7	Updated Interval Timer Read Control Description. Added SM CSR APMP Control Bit Bits Descriptions Added description to CDP section to explain when a CDP has been updated or refreshed. Added Bit 15 to BCCB for PE/IRIG Auto Insert option to Data Words for BC-RT Messages
7/10/2012	F8	Fixed Typos in SM CSR Bit 9 (Interval Timer Setting) and in BC Subframing discussion. Fixed NAICS Code to 334119.

2/6/13	F9	Changed NAICS Code to 34118 on Cover. Updated Single Ended Discrete and RT Section to review more about external RT Addressing and Enable lines.
6/24/13	G0	Updated Interval Timer Bit 6 Re-Arm, BC Retry IMG Note, CDP Drawing. Corrected Sequence Number in CDP and SM/BM Section.
10/15/13	G1	Fixed Typos in address offsets and IRIG specification.
11/11/13	G2	Added PTP IEEE-1588 References
3/1/14	G3	Clarified/Fixed Signal Generator Word/Bytes and Trigger Position
6/16/14	G4	Added additional IRIG support options. Added Flash Write Protect bit to Global CSR Register.
9/9/14	G5	Correct Interrupt API Pointer Address. Added IRIG Code Types.