IDPU Data Controller Board - Specification & Functional Description

Revision 0.3
26 February 2002

D. Gordon
ELF Electronics

References.

1. IDPU Data Controller Board Specification, Version D, 2002-Jan-08
2. Impact Harness Specification, Revision A, 2001-Mar-16
3. STEREO IMPACT Intra-Instrument Serial Interface, Version D, 2001-Oct-31
4. 8XC196KC/8XC196KD User's Manual, 1992, Intel
5. UT80CRH196KD Microcontroller, UTMC Microelectronics Systems, June 2000
6. Enhanced SuMMIT Family, Product Handbook, UTMC Microelectronics Systems, October 1999

Revision History

Revision Number	Date	Change Summary
0.1	18 January 2002	Initial Draft
0.2	21 January 2002	Misc. Corrections & Design Modification Telemetry Buffer Max Size Capped at 128KBytes/Subsystem
0.3	26 February 2002	Minor changes/corrections

Table of Contents

1.0 Introduction 3

2.0 Subsystem Descriptions 4

2.1 Clock/Reset Circuitry 4

2.2 CPU 4

2.2.1 Memory 4

2.2.2 Diagnostics Connector 6

2.3 Overall System Control Modules 7

2.3.1 CPU Bus Control 7

2.3.2 Interrupts 10

2.3.3 Timebase Generation 10

2.3.4 SuMMIT Interface 11

2.3.5 Command Interfaces 11

2.3.6 Telemetry Interfaces 13

2.3.7 Bus Arbiter 14

1.0 Introduction

The Data Controller Board is a component of the STEREO/IMPACT IDPU. It handles communication with the Spacecraft via a 1553 Bus, and receives telemetry/sends commands to five instruments via dedicated serial interfaces.

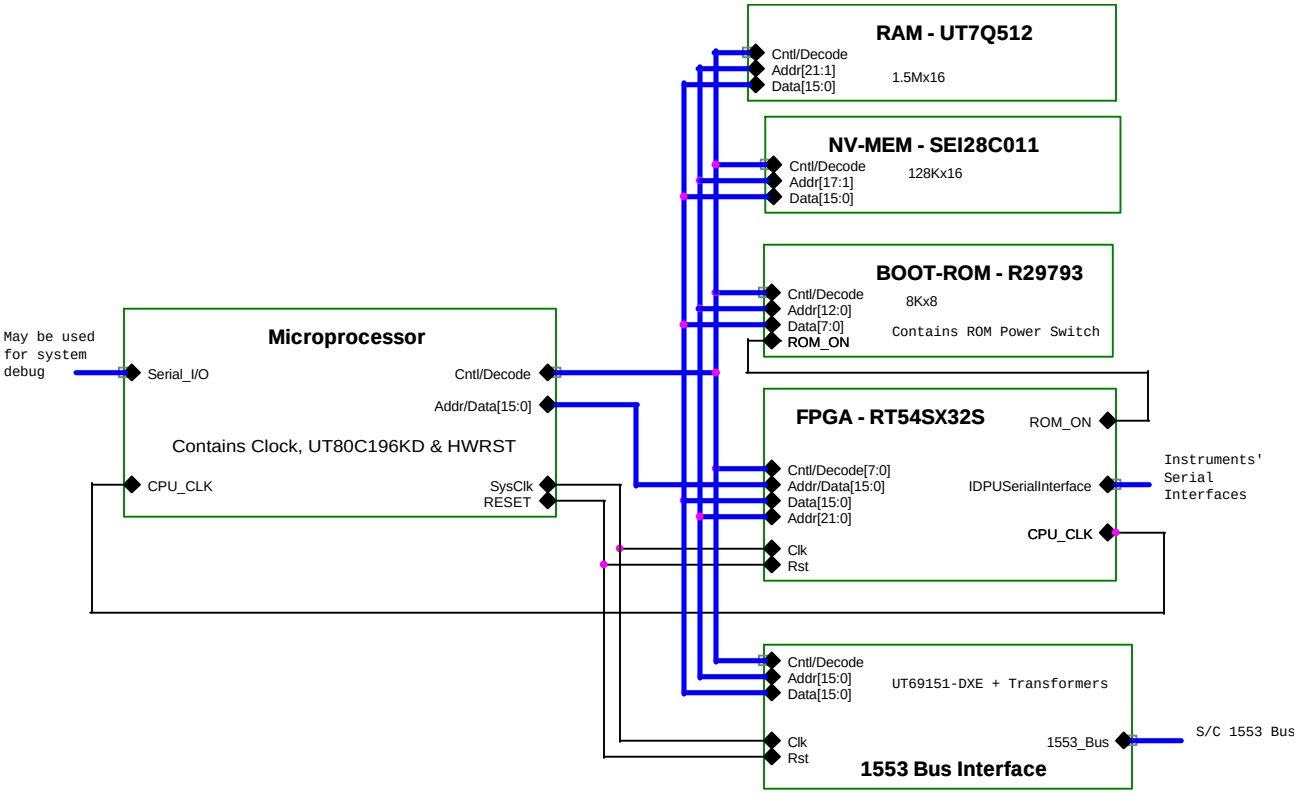

FIGURE 1. DCB: Overall Block Diagram

Figure 1 shows an overview of the main subsystems comprising the DCB. The microprocessor is an UTMC UT80CRH196KD 16-bit microcontroller. The FPGA handles the processor bus control and provides registers for paging support of the various sections of memory. Also included in the FPGA are serial instrument I/Fs, a DMA controller, and a timing generation subsystem.

2.0 Subsystem Descriptions

2.1 Clock/Reset Circuitry

All DCB clocks are derived from a 24.000MHz on-board oscillator. CLK24 is fed directly to the SuMMIT 1553 bus protocol chip and the FPGA. The FPGA divides down the master clock, generating 8MHz for the processor and 1MHz for the Serial I/Fs.

Power on reset is controlled by an RC with a time constant of approximately 0.1 seconds. Push button reset is also provided (which can be via onboard jumper for switch, or routed to the diagnostics connector). Watchdog reset is available in the 80196KD. (NOTE: Watchdog reset just restarts the processor; it does not drive the FPGA or the SuMMIT.)

2.2 CPU

The CPU subsystem consists of the UT80CRH196KD microcontroller. The master clock of 8MHz, supplied via a clock divider resident in the on-board FPGA.

2.2.1 Memory

The CPU memory bus directly connects to an external RAM (1.5Mx16, 6 UT7Q512), a boot ROM (8Kx8, 1 Raytheon R29793), NV Mem (128Kx16, 2 SEI 28C011 EEPROMs) and the peripherals (the Actel RT54SX32S (TBR) FPGA and the UTMC SuMMIT, UT69151DXE-WPC).

NOTE: All address references in the specification are in hexadecimal. However, memory region sizes are often quoted in decimal.

Upon reset, the processor begins 2080; the boot ROM area between 2000 and 2080 is reserved for dedicated processor information (configuration byte and vectors). After initialization, the CPU turns off power to the boot ROM via an on-board switch controlled via an FPGA control register bit.

SRAM, Boot ROM and I/O accesses are all zero-wait state bus cycles. EEPROM requires one wait state during write cycles. This is implemented via the EEPROM Write Enable (bit 6 of the FPGA Control Register). This signal serves two purposes: it allows for write protect during normal operation, and it inserts a wait state for all EEPROM access during write-enabled operation. It defaults to 0 (normal operation, EEPROM writes disabled, at reset.)

The SEI EEPROMs also contain software data protection.

Since the processor only supports a 64Kbyte address space, the FPGA includes a memory paging mechanism. The processor potentially provides for 128Kbyte of address space if the INST signal is decoded, as is done on the DCB Board. Therefore, two unique address spaces are available on the DCB, one allocated to instruction execution only, and the other for data.

Lower memory address (14 bits) are directly driven by the processor; upper memory address and chip selects are determined by four “Page Select” Registers.

The full linear map of external memory is shown below:

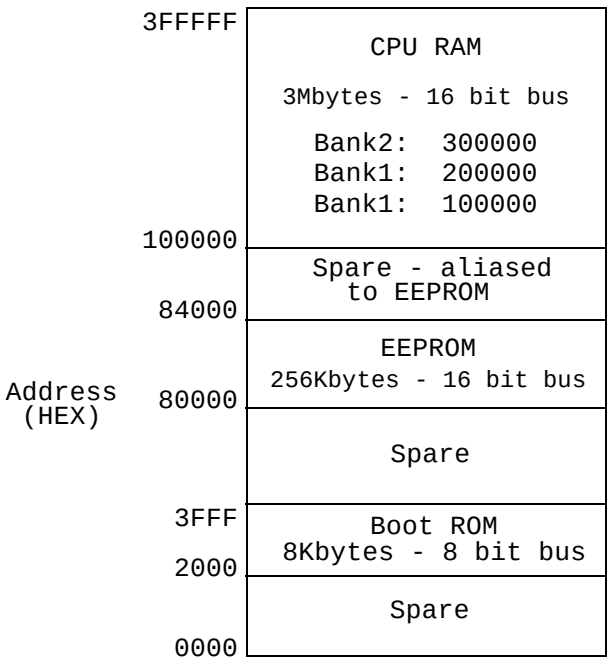

FIGURE 2. DCB External Memory Map

Four Page Select Registers, along with the CPU address, are used to select a region in the linear map. When INST=1 (code fetches), PageRegister 0 is always used; when INST=0 (data accesses) any of the four page registers can be used, with CPU address[15:14] determining the “window”.

For PageRegister 0 accesses, memory address bits [15:14] are directly driven by the processor. For PageRegisters 1,2 & 3, memory address bits [15:14] are included in the Page Register.

At reset, since all page select registers are initialized to 0, the processor will begin execution from the Boot ROM. (Note: Page Select Register 0 is shared by data and instructions accesses, and is the default register used following a system reset.)

The lower region of Window 0 is reserved for external peripherals and the internal processor registers and memory as follows:

2.2.1.1 Internal Processor Registers/SRAM

This region, located at 0000-03FF, overrides any external memory accesses when INST=0. Code fetches generate a chip select to the external memory defined by PageRegister0.

2.2.1.2 External I/O

External I/O (ranging from 1F00-1FFF) chip selects are gated with INST=0. Code fetches to the I/O space generate a chip select for the memory region currently defined by PageRegister0. Data fetches select I/O, overriding external memory.

External I/O is mapped as follows: the SuMMIT Protocol Chip registers at 1F00-1F7F; and the FPGA Registers at 1F80-1FFF (allowing 64 words for each).

2.2.2 Diagnostics Connector

A connector allows for the connection of external diagnostic peripherals and monitoring of the CPU Bus. Signals routed to the Nanonics Diagnostic Connector are:

SIGNAL NAME	DESCRIPTION	# Pins
AD[15:0]	Multiplexed Processor Address/Data Bus	16
Reset	Input to Overall DCB Board Reset - can be driven by a pushbutton	1
ALTROMSEL	Allows for Alternate Boot Device	1
DIAG Select	Allows for Diag LEDs/Switch	1
CPU Clock	4 MHz Clock - output of CPU	1
SelectAltBoot	Pulled up on DCB, can be grounded on “Debug Board” in order to select alternate boot memory	1
TXD, RXD	Serial Port Signals	2
ALE, CPURD, CPUWR, BHE, BUSWIDTH, HOLD, HLDA, EXTINT, EXTINT1	CPU Control Bus - ALE can be used for demuxing the AD bus on the “Debug Board”	9
MEMRD, MEMWR	Memory Read and Write Strobes	2
ADDR[21:14]	Upper Memory Address (Page Select)	8
SYSCLK	24 MHz System Clock	1
YF_INT, MSG_INT	SuMMIT Interrupt Lines	2
SuMMIT_CS, FPGA_CS	I/O chip selects	2
DMAREQ[4:0]	The 5 Serial Telemetry DMA Request Lines (fed out as status from the FPGA)	5
SUMDMAR	The SuMMIT DMA Request Lines	1
CMDDMAREQ	The Command Subsystem DMA Request Line	1
DMASEL[2:0]	Currently selected DMA Subsystem (fed out as status from the FPGA)	3
CSTR, DSTR	Strobes, generated by FPGA, indicating and active CPU or DMA access	2
SERERRS	Serial Subsystem Error Indicator (Or of all error detects asserted for any of the Serial Telemetry Subsystems)	1
CLK8M	Input to CPU	1

VCC (via PCB mounted fuse) and GROUND are also available on the diagnostics connector.

The connector has a total of 65 pins. 61 pins are allocated for the signals shown above, allowing 2 pins for power and 2 for ground.

2.3 Overall System Control Modules

The FPGA houses the following system control modules: CPU Bus Interface (Control/Interrupt Logic and FPGA Registers), Timing Generator, SuMMIT Interface, DMA Control and Bus Arbiter, a centralized Command DMA Subsystem and the five Instrument Serial Interfaces.

2.3.1 CPU Bus Control

The FPGA demuxes the CPU's combined address/data bus, and drives ROM and RAM, generating the Chip Selects and ByteWidth in accordance with the DCB External Memory Map, Figure 2 on page 5.

2.3.1.1 FPGA Based Registers

FPGA registers, starting at base address 1F80hex, are described below**:**

Address (hex)	Register Index (dec/hex)	Read Register	Write Register
1F80	0/0	Control - same as write value	Control - All bits default to zero at reset. Bit 6 - EEPROM Write Enable (defaults to 0, disabled) Bit 5 - Enable SuMMIT DMA (defaults to 0, disabled) Bit 4 - Enable Command DMA (defaults to 0, disabled) (does not disable the F0 command generation) Bit[3:2] - sets the Timer Interrupt Period: 3 -> 256 Hz 2 -> 128 Hz 1 -> 64 Hz 0 -> 32 Hz defaults to 0 (32 Hz) at reset Bit 1 - Boot ROM Disable: Controls power switch to Boot ROM; default is 0, ROM enabled Bit 0 - Enable SuMMIT, controls 1553 bus protocol chip reset line, default is 0, Reset asserted.
1F82	1/1	Page Select 0 - same as write value	Page Select 0 - Bit[5:0] - CPU Memory Page Addr[21:16]
1F84	2/2	Page Select 1 - same as write value	Page Select 1 - Bit[7:0] - CPU Memory Page Addr[21:14]
1F86	3/3	Page Select 2 - same as write value	Page Select 2 - Bit[7:0] - CPU Memory Page Addr[21:14]

Address (hex)	Register Index (dec/hex)	Read Register	Write Register
1F88	4/4	Page Select 3 - same as write value	Page Select 3 - Bit[7:0] - CPU Memory Page Addr[21:14]
1F8A	5/5	FPGA Version Number - Bits[7:0]	*Spare*
1F8C	6/6	Diagnostic - just provides external select (to Nanonics Connector)	
1F8E	7/7	Time Register - Internal Clock - same as write value	Time Register - Internal Clock Internal Clock Counter, maintained by the CPU; this is the value sent by the "F0" command. Bits[15:12]: Hours Bits[11:6]: Minutes Bits[5:0]: Seconds
1F90	8/8	Internal Counter Lo - (lowest 16 bits of the internal 1MHz Counter)	Write of any data to this address latches the counter (read at addresses 1F90 and 1F92)
1F92	9/9	Internal Counter Hi - Bits[3:0] - IntCount[19:16]	Note: write to address 1F90 before reading this register in order to latch the count.
1F94	10/A	Interrupt Enables - same as write value	Interrupt Enables - If set to 1, enables the these FPGA Interrupt (defaults to zero at reset) Sources: Bit[5] - Cmd_DMA_Buf_OFlow_Err Bit[4] - Latched MSG_INT Bit[3] - Latched YF_INT Bit[2] - CmdDMADone Bit[1] - TICKTIM Bit[0] - TICK1S
1F96	11/B	Status Register Bit[5] - Cmd_DMA_Buf_Oflow_Err Bit[4] - Latched MSG_INT Big[3] - Latched YF_INT Bit[2] - Latched CmdDMADone Bit[1] - Latched TICKTIM Bit[0] - Latched TICK1S	Pulse Register - Writing to this register with the appropriate bit(s) set causes the following actions: Bit[7] - Reset FPGA - clears all FPGA Based registers, and generates a reset pulse that drives all the FPGA Based Subsystems. Bit[5] - Clear Latched Cmd_DMA_Buf_Oflow_Err Bit[4] - Clear Latched MS_INT Bit[3] - Clear Latched YF_INT Bit[2] - Clear Latched CmdDMADone Bit[1] - Clear Latched TICKTIM Bit[0] - Clear Latched TICK1S
1F98	12/C	Command DMA Buffer Address Bit[15] - Command DMA Active - serves as a "BUSY" indicator. When a Command DMA block has completed, this bit transitions back to zero. Bits[14:8] - same as write value	Command DMA Buffer Address Bit [15] - Start Command DMA Bits[14:8] - CDMASA[21:15] Sets location of the 32Kbyte Command DMA buffer
1F9A	13/D	*Spare*	*Spare*

Address (hex)	Register Index (dec/hex)	Read Register	Write Register
1F9C	14/E	*Spare*	*Spare*
1F9E	15/F	SuMMIT DMA Status & Page Address Bit[7] - SuMMIT Output TERACT Bit[6] - SuMMIT Output READY Bits[4:0] - SDMAPage[21:17]	SuMMIT Status & DMA Page Address Sets Upper memory bits during SuMMIT DMA transfers Bits[4:0] - SDMAPage[21:17]

Address (hex)	Register Index (dec/hex)	Tlm SubSys #	Read Register	Write Register
1FA0 1FB0 1FC0 1FD0 1FE0	16/10 24/18 32/20 40/28 48/30	0 1 2 3 4	Tlm Buffer Start Page same as write value	Tlm Buffer Start Page Bits[11:0] - StPage[21:10]
1FA2 1FB2 1FC2 1FD2 1FE2	17/11 25/19 33/21 41/29 49/31	0 1 2 3 4	Tlm Read Pointer same as write value	Tlm Read Pointer Bits[15:0] - RdPtr[16:1] (Note: word address, relative to start page)
1FA4 1FB4 1FC4 1FD4 1FE4	18/12 26/1A 34/22 42/2A 50/32	0 1 2 3 4	NextMsg Start Address - Address + 1(wd) of last Complete Telemetry Message written into memory, up-dated at end of each message Bits[15:0] - NMStAdr[16:1] NOTE: upon activation (no telemetry messages in buffer yet), this address is set to StPage+0.	*Spare*
1FA6 1FB6 1FC6 1FD6 1FE6	19/13 27/1B 35/23 43/2B 51/33	0 1 2 3 4	Telemetry SubSys Control/ EndPage/Status Register Bits[14:8] - End Page[16:10] Bit[7] - Enable Outputs Bit[6] - Enable TlmSubsys Bit[2] - FramingErrLat Bit[1] - TimoutErrLat Bit[0] - BufOverrunErrLat	Telemetry Subsystem Control/ EndPage/Pulse Register Bits[14:8] - End Page[16:10] Bit[7] - Enable Outputs (CLK and COMMAND) (default to 0, disabled) Bit[6] - Enable Telemetry Subsystem (default to 0, disabled) Bit[0] - PULSE: Clear all TlmSubLatErrs (bits 2:0 of Status)

2.3.2 Interrupts

There are two external processor interrupts available: EXTINT1 (P2.2, pin 15, at vector location 203A) and EXTINT (P0.7, pin 9 at vector location 200E). EXTINT1, which is the higher priority of the two, is assigned to the FPGA, and EXTINT to the SuMMIT. NOTE: CPU Internal Register IOC1, bit 1, must be set in order to select P0.7 as the source for EXTINT. The processor provides separate masks for each interrupt, as well as a global interrupt enable.

2.3.2.1 FPGA (EXTINT1) Interrupts

The DCB-FPGA ORs the following interrupt sources together to drive the EXTINT1 line:

Interrupt Source	Description
TICK1S	Latched version of the 1 second tick - must be pulsed clear in order to reset.
TICKTIM	Latched version of the timing interrupt - must be pulsed clear in order to reset.
CMD-DMA-DONE	Indicates a Command Block DMA has completed - must be pulsed clear in order to reset.

All FPGA interrupt sources have a corresponding interrupt enable (default is all interrupts disabled at reset) which are readable as Status Bits in the Status Register, whether the interrupts are enabled or not. Upon entry to the interrupt service routine, all interrupts should be masked by clearing the Interrupt Enable Register. Upon exit from the interrupt service routine, any enabled interrupts should be unmasked. (This generates an edge on the EXTINT1 line, required by the processor to guarantee recognition.)

When setting the timer period, the TICKTIM interrupt should be disabled via the IENBS Register. Prior to enabling the interrupt, the Latched TICKTIM detect should be pulsed clear

2.3.2.2 SuMMIT (EXTINT) Interrupts

The SuMMIT provides two external interrupt lines: YF_INT (hardware related) and MSG_INT (message related). Both are latched in the FPGA, and can be read back as status. The two latched interrupt lines are ORed and forwarded directly to the CPU EXTINT input. (NOTE: the CPU must pulse the interrupt clear in the FPGA in order to deassert the external interrupt. No mask option is provided in the FPGA because all sources of YF_INT and MSG_INT can be individually masked within the SuMMIT.)

The SuMMIT contains a Pending Interrupt Register, with four bits related to YF_INT and twelve bits to MSG_INT. The SuMMIT also provides a programmable option of keeping a running interrupt log in external memory.

NOTE: A sequence of SuMMIT registers must be accessed in order to clear the SuMMIT based Pending Interrupt Register (see note at the bottom of the Reference 6, Page 11).

2.3.3 Timebase Generation

The 24MHz system clock is divided down to produce an 8MHz (33/67 duty cycle) processor clock and a 1MHz interface clock. The 1MHz clock, used for the serial command and telemetry

interfaces, also generates the TICKTIM and TICK1S. The 20-bit internal counter, cleared only by a hard reset, is readable by the processor at FPGA Register Address 1F90-2. NOTE: Write (any data) to address 1F90 prior to reading the counter in order to latch the current count for a stable readback.

2.3.4 SuMMIT Interface

The SuMMIT is an independent peripheral interface chip which transmits/receives data via the 1553 bus, and directly DMAs this data to/from memory. Most of the SuMMIT programming is done directly via its own register interface, directly mapped to I/O space at 1F00 to 1F7F. All accesses are word-only. Address 0 of the SuMMIT connects to CPU-Address 1 (locating SuMMIT Register 1 at Address 1F02, etc.)

The SuMMIT modes are configured as follows: $\overline{\text{AUTOEN}}$=False (1), $\overline{\text{LOCK}}$=True (0), MSEL=01 (Remote Terminal), and $\overline{\text{A/B\_STD}}$=0, select B (MIL-STD-1553B). The RT Address is set to 17 (decimal) and the RT Parity bit is set to 1 (for odd parity).

SuMMIT serial outputs are forwarded to two on-board transformers which directly drive the 1553 bus.

Status outputs TERAC (terminal active) and READY are readable as status (see Register 1F9E). TBR: JTAG is not used; TRST is tied to GND.

The FPGA drives the SuMMIT reset line, which defaults to 0 (reset asserted). The SuMMIT must therefore be enabled following a hardware reset via the FPGA Control Register.

SuMMIT DMA, handled by the FPGA, must be enabled via the FPGA Control Register in order for the DMAR to be recognized by the arbiter. Prior to enabling DMA, the SuMMIT DMA Page Register (setting Address Bits[21:17]) should be programmed, selecting a region in the SRAM. (Needless to say, the SuMMIT must be programmed as well, see Reference 6.) The SuMMIT chip drives Address Bits[16:1] during DMA cycles. Since all transfers are word-wide, Address 0 is a *don't care*. If the page is incorrectly set (i.e. outside of the SRAM memory region), writes are inhibited, and reads return indeterminate data.

DMA timing and bus allocation estimates are based on using the SuMMIT in non-buffered mode.

2.3.5 Command Interfaces

All five serial interfaces are serviced by a centralized command subsystem. It is responsible for:

- Reading (via DMA) a command list from memory, and transmitting each command to the appropriate recipient(s) via the serial command interfaces.
- Transmitting the "F0" Sample Clock command to all subsystems in accordance with Reference 3.

Command transmission complies with the protocol defined in the Intra-Instrument Serial Interface (Reference 3): 1 start bit followed by the 24 bit command, followed by 1 parity bit (odd parity) followed by at least one stop bit.

Upon power-up, the command interface is enabled and begins sending out F0 commands once per second. NOTE: It is the responsibility of the CPU to update the Time Register (at 1F8E). These F0 (or sample clock) commands are sent out precisely such that the command parity bit coincides with a maximum count on the internal counter (FFFFF) and the stop bit aligns with the minimum count (0). TICK1S (which causes a CPU Interrupt) aligns with the maximum count, and the F0 parity bit. The timing generator defines a "guardband" preceding the 1 second tick and informs the logic when to block DMA initiated commands in order to reserve a window for the Sample Clock Command. (The F0 Command is sent to all 5 serial subsystems simultaneously.)

The "Enable Command DMA" bit of the Control Register applies to Processor initiated DMA transfers and does not affect F0 Command generation. (NOTE: Command transmission for any subsystem occurs only if the serial outputs for that subsystem are enabled via its Telemetry Subsystem Control Register.)

A 32Kbyte buffer is available for processor initiated DMA commands. The queue must start on an even 32Kbyte buffer, defined by the Command DMA Buffer Address, and can end on any multiple of 4 bytes into the buffer. A terminator byte of 0x80 must be placed at the end of the buffer. If the system reads the last byte of the 32Kbyte buffer, an error condition (Command Buffer Overflow) is flagged and Command DMA terminates.

In order to transmit CPU initiated commands, the processor must:

- Enable DMA via the Command Register. This capability is included primarily for catastrophic error recovery; deassertion of the enable aborts an ongoing Command DMA block. It probably won't be necessary, but is included "just in case".
- Write a Queue of Command Records into SRAM. A command record consists of two 16-bit words as follows:

CommandWord[7:0]	CommandMask[7:0]
CommandWord[23:8]	

  The lower 5 bits of the command mask is a bit mask, which enables subsystems independently such that any or all subsystems can be tagged to receive each command. When the mask is set to all zeros, the system will effectively "mark time", inserting 26μsec of deadtime before issuing the following command. (The command following a zero mask is a *don't care*.) Bit 7 of the command mask serves as an "End of Block" indicator. When the logic detects "End of Block" it halts transmission and returns to an inactive state. Every command queue should terminate with a 0x0080.
- Set the Command DMA Buffer Address. Write the Command DMA Buffer Address (at Register Address 1F98) with bit 15 set. Setting Bit 15 of the DMA Buffer Address Register activates the transfer. Bit 15 also serves as a "Busy" status bit during readback (it stays set until the DMA block has completed transmission), allowing for polling. A Command_DMA_Done Interrupt (maskable) is also provided.

2.3.6 Telemetry Interfaces

There are five independent telemetry interfaces, each with its own enable and set of registers. Telemetry buffers can start and end on arbitrary 1K boundaries, defined by the START_PAGE and END_PAGE registers. The maximum size of any telemetry buffer is 128KBytes. Therefore the END_PAGE Register only requires 7 bits, with the START_PAGE upper bits[21:17] driving a constant value (different for each subsystem). Similarly, the READ_POINTER and NEXT_MESSAGE_START_ADDR Registers require only 16 bits (word addresses),

The Telemetry subsystem can be enabled, and the outputs enabled via the Telemetry Subsystem Control Register. Other registers are:

START_PAGE (set by the CPU): Set Memory Address[21:10] of the first buffer word

END_PAGE (set by the CPU): Set Memory Address[16:10] of the last buffer word

READ_POINTER(set by the CPU): Indicates the last word location that the CPU has read-out of the buffer

NEXT_MESSAGE_START_ADDR: (read-only): indicates the address + 1 (in words) of the last complete message written into memory.

All telemetry subsystem registers are located at the FPGA Register Addresses 1F*n*0 to 1F*n*A, where *n* is a hex nibble corresponding to the subsystem number (A=0, B=1, C=2, D=3 and E=4). See the entries for Tlm Registers in Section 2.3.1.1 for a complete listing of all the register addresses.

Before enabling telemetry, the processor should write the START and END Page Pointers, and initialize the READ Pointer. (Typically the read pointers would be initialized to the last buffer location: END_PAGE + 3FE.)

Each telemetry subsystem page is constrained to a 128KByte memory segment. The upper bits of the START_PAGE register [21:17] are static, and define the location of the segment. The lower bits [16:1] are transferred to the CURRENT_ADDRESS register at startup. The CURRENT_ADDRESS register is incremented after each DMA transfer.

The Page Pointers allow for buffers of any increment of 1KBytes for each telemetry subsystem. Once a subsystem is enabled, it begins writing telemetry words (via DMA), as they arrive from the serial interface, to the first address in the buffer (START_PAGE + 000). After each write, a word-incremented address is transferred to the CURRENT_ADDRESS holding register. When CURRENT_ADDRESS = END_PAGE + 3FE, START_PAGE[16:1] + 000 is reloaded into the CURRENT_ADDRESS register.

If CURRENT_ADDRESS = READ_POINTER, this indicates that the write pointer has caught up with the read pointer (a buffer overrun error). When a buffer overrun error is detected, the system rolls back the CURRENT_ADDRESS to the beginning of the current message (thus erasing any incomplete messages) and stops writing into memory until the start of the following message after the overrun error has cleared.

NOTE: The CPU should update the READ_POINTER regularly as it drains the telemetry buffers.

2.3.6.1 Telemetry Shift Logic

Upon assertion of the enable, the shift-in logic (TSHIFT) seeks 17 zero bits on the data line to achieve inter-message synchronization. Once the zero pattern has been found, a start-bit on the data line causes the next word to be shifted in. The TSHIFT module checks for an improper message due to a stop bit following the first message word (which should never occur because the minimum message size is 2 words). If such a condition happens, the message is discarded and the first word is never written into memory. The framing error latch is set (see below) and the shifter returns to its initialization state (seeking the 17 inter-message zeros).

If TSHIFT successfully starts to receive a message (i.e. one word followed by the second word's start bit), it activates the Master Telemetry control state machine (MTLMCNTL) which begins to write the message words into memory. Each time a word arrives at the interface, the logic requests the bus and writes a word (via the bus arbiter and DMA interface) into memory. TSHIFT determines the END-OF-MESSAGE and signals to MTLMCNTL when the last word has arrived. After receiving the last word, TSHIFT seeks 17 inter-message zeros.

2.3.6.2 Telemetry Error Detection

The following error conditions are detected by the logic:

Framing Errors - mentioned above - message is not written into memory and a latch is set

Timeout Error - if a second telemetry word arrives before the first one has been written into memory, a timeout error is latched. This condition should never occur during normal operation and would indicate a hardware error.

Buffer Overrun Error - if the write pointer collides with the read pointer, a buffer overrun error is latched.

The Message Length Error is not detected by the DPU, but may be put in later if it fits in the FPGA.

All errors cause the write pointer to roll back to the "Start of Message" position, so that the "damaged" message does not appear in the message buffer. Error status is readable in the Telemetry Subsystem Status Register(s). The latched error bits remain set until the CPU writes to the Telemetry Subsystem Pulse Register(s).

2.3.7 Bus Arbiter

The bus arbiter fields requests from seven independent clients: the SuMMIT 1553 IC, the five telemetry subsystems, and the command subsystem.

Upon detection of a request, the arbiter asks the CPU for the bus via the HOLD-HLDA handshake. (It is expected that the maximum CPU hold latency is approximately 650ns.) Upon obtaining the CPU bus, the arbiter assigns it to the highest priority subsystem. If a second request arrives during the HLDA period, the arbiter allows for a second hardware cycle before giving the

bus back to the CPU. (This avoids another lengthy bus latency period, at very little cost to the CPU.)

Priority is assigned as follows:

1. SuMMIT (requests bus every 20µs when performing 1553 block transfer; it also asks for four back-to-back transfers at the beginning of a block, and six back-to-back transfers at the end of a 1553 bus block.)
2. Telemetry Subsystems (in this order 0, 1, 2, 3, 4) (request bus every 17µs when transmitting telemetry messages) Telemetry Subsystems are mapped as follows: 0: Plastic, 1: SWEA, 2: MAG, 3: SEP, 4: STED.
3. Command Subsystem (requests bus every 26µs when a command queue is being processed).

The bus arbiter handshakes with the client subsystems informing them when they have the bus, and then allowing them to relinquish the bus when the cycle has completed. Since the 24MHz system clock is used to generate the strobe timing, it is expected that the maximum time a client will hold the bus is on the order of 200ns. The SuMMIT generates its own strobes and addresses once it obtains the bus.

As a common service to the FPGA client subsystems, the Bus Arbiter includes an address incrementer. Each time a client is granted the bus, it outputs its current address, which the FPGA selects and drives out to memory. This address is fed into an incrementer, with the result fed-back to the client subsystems, who latch it as required into their running address registers.