BRM1553D
IP Core Hardware Interface Manual

Mil-Std-1553 Bus Controller, Remote Terminal and Bus Monitor

SW Driver Compatible With DDC® ENHANCED MINIACE®

USER'S MANUAL FOR:

BRM1553D IP Cores

Rev 8.01

May 2016

* DDC® and MINI-ACE® are registered trademarks of Data Device Corporation, Bohemia, NY, USA.
There is not any affiliation between Data Device Corporation and Sital Technology, Ltd.

Copyright (C) 2016 by Sital Technology Ltd.

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Sital Technology Ltd.

TABLE OF CONTENTS

1 INTRODUCTION ____ 5
1.1 About this manual ____ 5
1.2 EBR1553 ____ 5
1.3 About BRM1553D – Mil-Std-1553 Core ____ 6
1.4 IP Core Deliverables ____ 7
1.5 Terms used in this document ____ 7
2 hardware Interface ____ 8
2.1 Block Diagram ____ 8
2.1.1 1553 Front-End ____ 8
2.1.1.1 The RT state machine ____ 9
2.1.1.2 The Monitor state machine ____ 9
2.1.1.3 BC State Machine ____ 9
2.1.2 Decoder/Encoders ____ 10
2.1.3 Transceiver Interface ____ 10
2.1.4 DDC® Compatible Interface block ____ 10
2.1.5 Dual-Port RAM ____ 11
2.1.6 Local Bus Host Interface ____ 11
2.1.7 SmartWiring ____ 12
2.2 Interface signals (16 bit data) ____ 13
2.3 Interface signals (32 bit data) ____ 13
2.4 Read and Write Cycle timing ____ 18
2.4.1 Write Cycle ____ 18
2.4.2 Read and Interrupt Cycles ____ 19
2.5 Hardware Connection to Transceiver ____ 20
3 VHDL simulation test bench ____ 22
3.1 Test Bench Top ____ 22
3.2 Bus Tester ____ 23
Appendix A: Changes tracking ____ 25

TABLE OF FIGURES

Figure 1: BRM1553D-RTMT Inputs/Outputs 6
Figure 2: BRM1553D Block Diagram 8
Figure 3: Write Cycle 18
Figure 4: Read and Interrupt Cycles 19
Figure 5: IP Core connection to the 1553 transceiver 20
Figure 6: BRM1553D connection to Sital Discrete Transceiver 21
Figure 7: Simulation Test Bench 22

1 INTRODUCTION

1.1 ABOUT THIS MANUAL

This document is the user's manual for BRM1553D, IP core for Sital's Mil-Std-1553 products. It covers the hardware installation, simulation and testing of Sital's BRM1553D IP cores for FPGAs.

The Hardware-Software interface is covered by the BRM1553D_Hardware-Software-Interface Manual (HSID).

In some configurations of the core, depending on your product, the BC, RT and MT are packed together as a single IP core (BRM1553D-BCRTMT) or product. All reference made for BRM1553D-RTMT is equal to BRM1553D-BCRTMT, when in RT & MT Mode, and all reference made for BRM1553D-BC is equal to BRM1553D-BCRTMT, when in BC Mode.

1.2 EBR1553

The BRM1553D IP core has a version that supports EBR1553 standard AS5652 by SAE Aerospace. The relevant signals that are used in EBR1553 and not in the MIL-STD-1553 IP core are listed in the appropriate paragraph. Please note that the EBR1553 version of BRM1553D only communicates in EBR1553. Please refer to the HSID for more details on EBR1553 capabilities.

Note:
Please note that it is assumed that the user of this manual is knowledgeable about the DDC® Mini-ACE® and Enhanced Mini-ACE® components, and their software interface. It is also assumed that the user of this manual has knowledge of Mil-Std-1553 protocol.

1.3 ABOUT BRM1553D – MIL-STD-1553 CORE

The BRM1553D core for FPGA provides a simple to use link for the MIL-STD-1553B Notice 2 bus interface.

The core supports all modes of Bus Controller (BC), Enhanced Bus Controller (eBC), Remote Terminal (RT), Message Monitor and Word Monitor (MT).

The BRM1553D core is designed to have the same Memory and Registers mapping and contents as DDC's Enhanced mini-ACE has. This compatibility allows the usage of software drivers developed for DDC's device to work with the core without any SW changes. Some of Sital's components, boards and other products are also pin-to-pin compatible to respective DDC® components and boards, making this a true replacement.

The BRM1553D as a stand-alone core was validated to meet the MIL-STD-1553B Notice 2 Remote Terminal Validation test plan, thus reliving the user from mastering the standard.

Note:

BRM1553D does not implement all of DDC's mini-ACE features, but rather the features which are used for the application. In most designs only a very small set of features are used from the DDC's interface. These features are implemented in the core. This document describes the parts that are implemented. If there is a feature that is required by your software, but is not supported by the core, then please contact Sital Technology.

The following diagram shows the inputs and outputs of the IP core: (some pins are not shown)

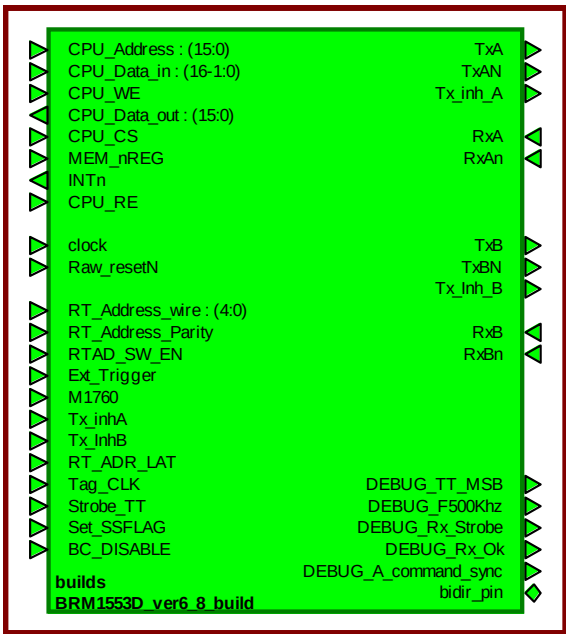

Figure 1: BRM1553D-RTMT Inputs/Outputs

*This Figure might not show all signals.

1.4 IP CORE DELIVERABLES

The IP core is supplied with the following components:

- EDIF net list for desired FPGA family and clock frequency.
- This user's manual.
- Sample VHDL code that incorporates the core.
- Synthesis script for sample code.

1.5 TERMS USED IN THIS DOCUMENT

- **Remote Terminal (RT)** – The part of the FPGA that manages the 1553 communications and implemented by the core.
- **Bus Controller (BC)** – the system that handles Mil-Std-1553 timing and manages the transmission of RT's on the bus.
- **eBC** – Enhanced Bus controller.
- **SubSystem** – The whole box that connects to the MIL-STD-1553 bus that contains the FPGA – part of it is the Remote Terminal.
- **Host** - the CPU running the SubSystem and managing the IP core interface.
- **FPGA** – Programmable device that contains the BRM1553D - RT & MT MODE core and user logic and is part of the Subsystem.
- **User Logic** – Logic Circuit that resides in the FPGA that is not the BRM1553D - RT & MT MODE core and connects to it.
- **Message** – the group of command data and status words that compose a 1553 message.
- **IP** – Intellectual Property
- **Core** – Supplied logic circuit that interfaces to MIL-STD-1553 bus.
- **ICD** - Interface Control Document.
- **TA** – Terminal Address of the command / status words. Bits 11 to 15.
- **SA** – Sub Address of the command word. Bits 5 to 9.
- **WC** – Word count field of command. Bits 0 to 4.
- **Muxbus** – Time multiplexed bus known as the MIL-STD-1553B Notice 2 bus.
- **SW** – software.
- **BCST** – Broadcast command.
- **TX** – Transmit.
- **RX** – Receive.
- **SACW** – The Sub-address control word.
- **LUT** – Look Up Table.
- When a reference is made to a particular register and a specific bit, the notation is R01B12 for Register address 0x0001 and bit 12.
- When a reference is made to a particular memory location, the notation is M0100 for memory address 0x0100.

2 HARDWARE INTERFACE

2.1 BLOCK DIAGRAM

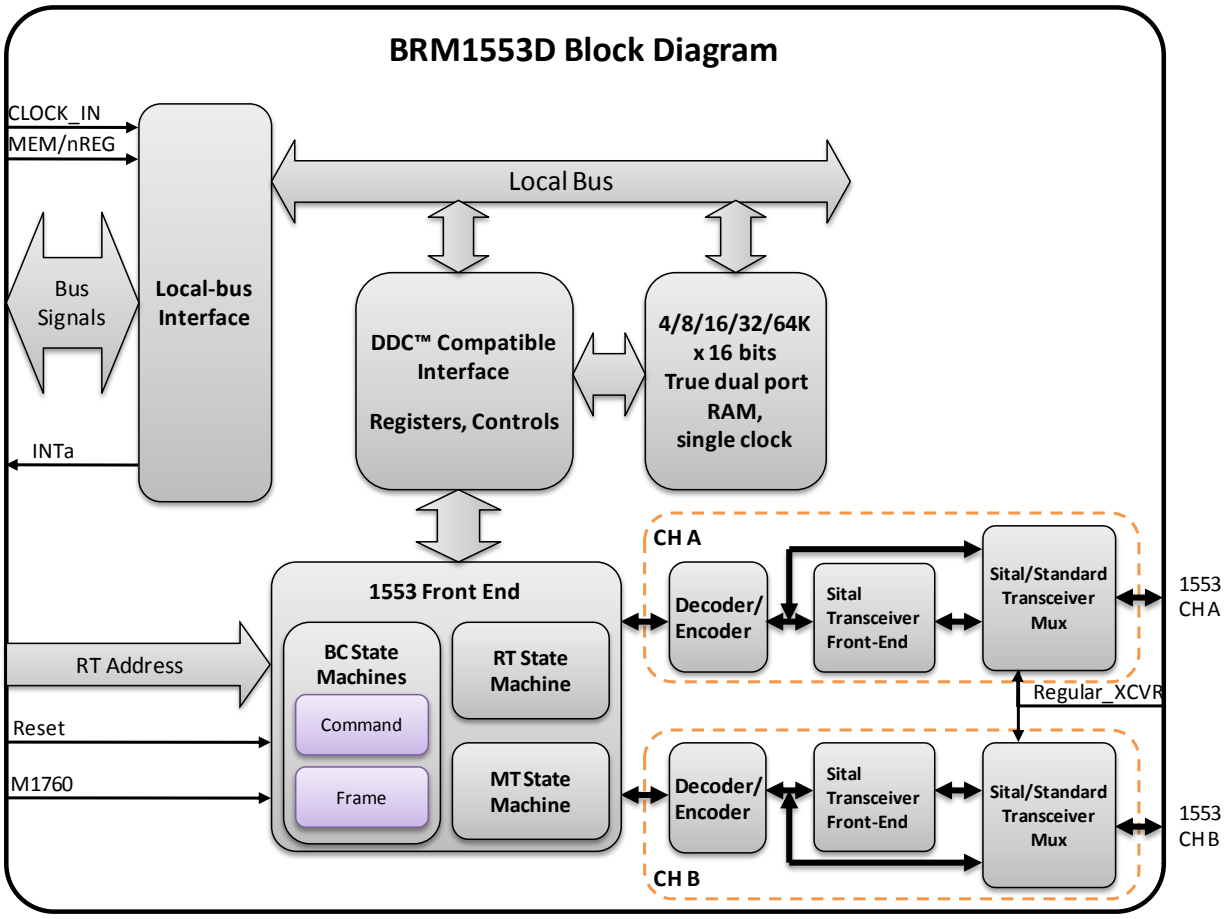

Figure 2: BRM1553D Block Diagram

2.1.1 1553 FRONT-END

Each BRM1553D Core contains a 1553 Front-End core which performs all the functionality of managing the Mil-Std-1553 protocol interface. This core is responsible for producing the messages, commands, responses, and managing the 1553 frames. The core is built of several state machines that follow the Mil-Std-1553B standard. There are separated state machines for Remote Terminal, Monitor and Bus Controller.

The 1553 Front-End core is controlled be the configuration registers at the DDC® Compatible Interface block. RT Address pins are connected directly to the 1553 Front-End so that in case of 1760 protocol requirement the core will respond with a “Busy” bit to any request from the 1553 bus, even before the control registers and memory are configured by the host CPU.

A Reset input is used for resetting all state-machines.

As a configuration option, it is possible to order the BRM1553D core just as a RT and MT core. This configuration can be used in case a design requires Remote Terminal or Monitor operation only. In such case the core will prevent any option of accidently configuring the device as Bus Controller and will also save some FPGA space.

2.1.1.1 THE RT STATE MACHINE

The RT state machine constantly listens to the bus activity and identifies a set of bus words as a valid message, being 'transmit', 'receive', 'mode' or 'broadcast' message, and acts accordingly.

Data messages being received from the bus are saved into the dual-port RAM via the DDC® compatible interface block as they arrive. Messages that need to be transmitted to the bus are fetched from the dual-port RAM via the DDC® compatible interface block, as they are being transmitted to the bus.

Errors, channel status and other information on the 1553 communications status is reported to the DDC® compatible interface block, and reflected to the software at the control registers and memory.

2.1.1.2 THE MONITOR STATE MACHINE

The Monitor state machine searches for valid commands. When a command is found, the state machine checks whether this command defines a message required for monitoring. In such case, the state machine manages the process of storing the words one by one in the Dual-Port memory via the DDC® compatible interface block into a pre-defined location.

2.1.1.3 BC STATE MACHINE

When transmission is initiated the frame state machine manages the sequencing of a whole frame of messages as defined by the configuration registers and fixed memory locations in the memory. The frame is composed of a set of individual messages being transmitted and managed over the 1553 bus one after the other. The host also points the beginning of the first message in memory and defines how many messages to transact through 2 fixed location memory words. When all data has been loaded, and the state machines are idle, the host sends a START command. As a response, the frame state machine starts the frame transmission. The frame state machine fetches the messages from memory and forwards the requested message information to the command state machine which in turn sequences the command data and status words for a complete legal 1553 message. When the message is complete the frame state machine accesses the next command, and so on until all messages have been completed.

The command state machine either transmits words through the encoders or receives RT responses through the decoders. The encoders and decoders interface between the core's 16 bits parallel internal buses and the MuxBus serial bus.

2.1.2 DECODER/ENCODERS

The Front-End core is connected to two separate Decoder/Encoders, one for each dual-redundant 1553 channel. The Decoder/Encoder blocks translate the serial bus messages from the MIL-STD-1553 Manchester coding and format, into a 16 bit parallel data accompanied by status indications for each word. These Decoder/Encoders are carefully designed to overcome noise and other problems related to the 1553 bus.

2.1.3 TRANSCEIVER INTERFACE

The BRM1553D IP core can support two types of 1553 transceivers:

1. A regular COTS MIL-STD-1553 transceiver, using the standard transmit, receive and inhibit signals per channel.

2. A discrete component transceiver developed by Sital Technology. This transceiver requires two additional lines per channel. These four additional lines should be left unconnected if using a COTS transceiver.

If the IP supports both types of transceivers, an input control line 'Regular_XCVR' can select the desired transceiver interface. If set to high, a regular transceiver is connected to the IP, else if low the Sital discrete components transceiver should be used.

The Sital Discrete Transceiver requires some additional control logic, which is managed by the Sital Transceiver Front-End cores. These cores perform signal shaping, filtering and short-circuit protection for the Sital analog Front-End.

In case the IP supports only the COTS transceiver, the 4 additional lines are not relevant and the line 'Regular_XCVR' has no effect.

2.1.4 DDC® COMPATIBLE INTERFACE BLOCK

The operation of the 1553 Front-End state machines is controlled by the DDC® Compatible Interface block. This block contains the control registers and memory interface to the dual-port RAM. This block creates the control signals for the 1553 Front-End core. The registers and interface are explained in details in BRM1553D Hardware - Software Interface Document.

The interface to the device is divided between several control registers and memory access. The registers are used to control the device and its operation, and the memory is used as the 1553 message interface and control.

The software interface of BRM1535D to the host processor consists of 32 internal operational registers for normal operation. These registers determine the device configuration, modes of operation, memory structure, interrupt control and status, etc.

The registers are mapped to address 0x0 to 0x1F (Hexadecimal) and can be written or read (depending on their functionality). Memory can be 4K, 8K, 16K, 32K or 64K (configured during synthesis) – all by 16 bits.

The data reception and transmission is controlled on a message-by-message basis. Each message is stored in the memory or read from the memory based on mapping defined by the host CPU during offline state. The host CPU sets up the mapping and modes of operations in the dual port RAM and in the configuration registers. The mapping options are discussed in the Hardware - Software Interface Document.

Access to the registers or memory is done through the same address and data lines. When accessing the registers, the MEM/REG signal (which is one of the bus signals) should be kept low, and when accessing the memory this signal should be high.

The IP can be synthesized to a variety of data bus widths. The 16 bit data width is the classic mini-ACE compatible mode. 32 bit data bus is very popular and might save a lot of software byte manipulations, and support burst reads and writes. In 32 bit bus, two 16 bit words can be read or written with once clock cycle. For writing and reading, two read and write controls are provided, for each 16 bit word to support single read or write. In 32 bit bus, the address LSB is ignored by the IP. Wider data busses are available upon request.

There are several dependencies between the configuration of registers and the configuration and operation of the memory. The user must verify that all dependencies configured correctly with accordance to the required operation of the device.

2.1.5 DUAL-PORT RAM

The True Dual-Port RAM stores messages received or to be transmitted to the 1553 bus. The messages are arranged in memory with accordance to the DDC® Mini-ACE® memory structure and user's configuration.

The memory is a true dual port RAM, with both sides independently reading or writing data. When operating as Local Bus, it is preferred that the Host CPU will supply the memory control signals, address, data, chip select and write enable, synchronized with the clock signal supplied to the core. This synchronized approach will ensure robustness of operation. Since BRM1553D core and the Host CPU work with the same clock, the design will have no transient effects. Therefore, BRM1553D IP core enables a wide selection of clock frequencies, between 12MHz and up, in even numbers (12, 14, 16… MHz), that should be specified by the user prior to the synthesis of the core by Sital.

2.1.6 LOCAL BUS HOST INTERFACE

The CPU controls the BRM1553D core by programming the dual port memory for message specific settings, and by programming a set of configuration registers for device wide controls. A special select signal - MEM_nReg – selects between these two sections and should be defined by the CPU for each read and write cycle.

The memory is a true dual port RAM as defined by the target FPGA, with both sides independently reading or writing data. The CPU should supply the memory control signals, address, data, chip select and write enable, synchronized with the clock signal supplied to the core. This synchronized approach will ensure robustness of operation. Since the core and the user logic work with the same clock, the design will have no transient effects.

The BRM1553D core was designed to work with any even number Mhz bigger than 12, or any Odd Mhz bigger than 25. The core was designed to work with the **SAME clock** that is used by the user logic that writes or reads memory and registers data. Thus it is mandatory that any read or write access to the core would be synchronous, i.e, same clock domain, with the clock that is fed into the core, otherwise data consistency would be hurt.

If the system clock is not as stated above – it is important to use a PLL that will multiply the clock to the desired range, and this clock be used with the interfacing CPU logic.

2.1.7 SMARTWIRING

As of version 8.0 all BRM1553D devices are equipped with SmartWiring™ technology.

Please refer to the HSID for a detailed description of SmartWiring.

In order to facilitate SmartWiring, there are a few changes to the IP.

1. SmartWiring indicates a bus wiring fault detection with a pTDR interrupt line added to the IP (It also reports a fault in interrupt status register #1, bit 14).
2. SmartWiring requires more address space for the registers space. Without SmartWiring, only 32 words of 16 bit address space is required, with SmartWiring, 128 register address space is required (256 bytes).
3. In order to produce the relevant resolution for determining a fault, an additional clock is required (SmartWiring_Clock). This clock has to be a PLL multiple of the main clock, that is, be in the same clock domain of the main clock. It should be in the range of 100Mhz to 200Mhz.
4. If the main clock is in that range, use it without multiplying it.

2.2 INTERFACE SIGNALS (16 BIT DATA)

16 bit data interface has been suppressed as of Ver 8.0.

If case 16 bit data interface is required, please refer to test bench example to see how a 16 bit CPU is connected to the 32 bit bus.

2.3 INTERFACE SIGNALS (32 BIT DATA)

Only pins that are different from 16 bit wide are listed:

Signal Name	In/Out	Pins Description
CPU_Data_in	In	(31..0) A 32 bit data bus. 15..0 are targeted to even addresses. 31..16 are targeted to odd addresses.
CPU_WE	In	2 bits. For 32 bit CPU_WE(0) writes bits 15..0, CPU_WE(1) write bits 31..16
CPU_RE	In	bits. For 32 bit CPU_RE(0) read bits 15..0, CPU_RE(1) read bits 31..16 CPU_Data_out reads without the CPU_RE. These are needed for clearing the interrupt status registers in certain configurations.
CPU_Data_out	Out	(31..0) A 32 bit data bus. 15..0 are sourced from even addresses. 31..16 are sourced from odd addresses.

The BRM1553D core requires several controls to properly configure the core for work. The controls are:

The core connects with the CPU and User Logic with the following lines (16 bit data width):

Signal Name	In/ out	Description
RT_Address_wire	In	5 bits that define the address of the Remote Terminal on the bus. These bits should not change their logic state during the operation of the core. If the terminal address is 20 decimal, then that value of these bits should be “10100”.
RT_Address_Parity	In	A single bit that compliments the parity of RT_Address_wire to odd parity. If wrong parity is detected by core, only Broadcast commands would be valid for the core. If not masked, an interrupt will be generated in the event of wrong parity.
RTAD_SW_EN	In	‘1’ - Enables SW change of RT address by configuration register #6 (see below for further details). ‘0’ – No SW change of RT address available.
RT_ADR_LAT	In	‘0’ – RT Address lines & their parity bit are passed as-is to the protocol engine and continuously monitored. Rising – The last RT Address lines & their parity bit are sampled and kept inside the core. Changes in the RT address or parity bit are ignored. ‘1’- RT Address can be latched by means of writing to configuration register #5. RT Address & Parity can be programmed by Software as well, please see configuration registers 4, 5, and 6 for details.

Clock	In	The core is synthesized to work with a specific clock frequency as requested by the user. This pin should be connected to that clock. The clock should be one of the global clocks of the FPGA. The logic circuits that interface with the core should also work with this clock.
SmartWiring_Clock	In	SmartWiring_Clock clock derived from the same clock as the “clock” signal. It should be between 100 and 200Mhz.
Raw_ResetN	In	An active low signal that asynchronously resets all of the FFs in the core. This signal is internally sampled to the clock. This signal does not reset the dual port memory of the message data. This signal can be used to inhibit the core from usage. Once the ResetN is logic high, the Remote terminal starts to operate and respond to the bus based on memory and register settings.
TAG_CLK	In	Serves as the clock for the time tag counter in case configuration register #2 enables this mode.
CPU_CS	In	CPU Chip select. When logic high, enables CPU accesses. When low, CPU accesses are ignored.
CPU_Address	In	(15..0) A 16 bit address used to access the memory array and configuration registers. Memory depth is generically defined when the core is synthesized. Up to 64K x 16 bit memory is possible.
MEM_nReg	In	When logic high connects the CPU address and data lines to the memory array. When low connects the configuration registers.
CPU_Data_in	In	(31..0) A 32 bit data bus. 15..0 are targeted to even addresses. 31..16 are targeted to odd addresses.
CPU_WE	In	2 bits. For 32 bit CPU_WE(0) writes bits 15..0, CPU_WE(1) write bits 31..16
CPU_RE	In	2 bits. For 32 bit CPU_RE(0) read bits 15..0, CPU_RE(1) read bits 31..16. For some FPGAs, CPU_Data_out reads without the CPU_RE. These RE pins are required only for read and clear of the interrupt status registers in certain configurations.
CPU_Data_out	Out	(31..0) A 32 bit data bus. 15..0 are sourced from even addresses. 31..16 are sourced from odd addresses.
INTn	Out	An active low interrupt line set by the core based on the message settings in the memory, or the configuration registers. Please see configuration methods as defined by configuration register #2.
pTDR_INT	Out	An active high signal set by the SmartWiring core when wiring disconnection is detected. Note that is a wire toggles between good and bad, the interrupt would also follow in the same sequence, which could impose on the CPU. Masking the bit may reduce further interrupts.
Tx_InhA	In	‘1’ – inhibits 1553 transmission on bus A. ‘0’ – default value. Enables transmission of core on bus A.
Tx_InhB	In	‘1’ – inhibits 1553 transmission on bus B. ‘0’ – default value. Enables transmission of core on bus B.
Ext_Trigger	In	Rising Edge – triggers the BC frame when configured to trigger on external trigger. See BC user manual. Rising Edge – triggers the Word Monitor frame when configured to trigger on external trigger. See Word monitor user manual. In RT mode, has no effect. Sampled on rising edge of clock every time clock enable is high. Clock enable is internal in the range of 12 to 50 Mhz.

Set_SSFlag	In	RT mode only, '1' sets the Sub System flag (bit 2) high in status word.
M1760	In	'1' – Support for MIL-STD-1760. Will power up as RT with Busy bit set. '0' – Powers up as inactive BC or Idle.
Strobe_TT	In	Rising Edge – strobes the Time Tag register to the Latched Time Tag register address 0x15.
BC_Disable	In	'1' – disables the BC mode even if SW enables BC. '0' – BC mode controlled by SW.

The core connects with the transceiver with the following lines for each of the busses (not relevant for BRM1553D-EBR):

Signal Name	**In/ out**	**Description**
Regular_XCVR	In	'1' – Use COTS transceiver. '0' – Use Sital Technology Discrete component transceiver.
RxA, RxAn, RxB, RxBn	In	The 1553 receive signals. The transceiver Rx Enable signal, if exist, should be tied to constant enable level and is not used by the core.
TxA, TxB, TxAn, TxBn	Out	The 1553 transmitter signals. Drive the bus. Connected to the Tx lines of the transceiver. Bus A signals and Bus B signals.
Tx_inh_A, Tx_inh_B	Out	Connected to the tx inhibit line of the transceiver. **This line must be tied to a pull up on the PCB**. This will inhibit transmission for the period of time that the FPGA is not loaded and all of its ports are logic tri-state. They are logic low when transmitting, else high. If Sital transceiver is used (BRM1553D-RTMT-S), these signals are connected to the En_A and En_B of the transceiver circuit. They are logic high when transmitting.
Tx_A_SN, Tx_AN_SN, Tx_B_SN, Tx_BN_SN,	Out	Relevant for BRM1553D-RTMT-S only. Connects to Sital Technology transceiver signals. Should be left unused if COTS transceiver is used.

For BRM1553D-EBR mode the following signals are used in addition, or replacing the listed above signals:

Signal Name	**In/ out**	**Description**
HUB Mode(1..0)	In	"0x" - SPEC mode. BC Tx on all, One RT reply.(Link # =/= RT #) "10" – Switch mode. BC Tx on single, RT reply.(Link # == RT #) "11" – Link Mode. BC Tx on single, RT reply. All RTs with same RT address!
RT_Address_wire and RT_Address_Parity	In	These pins that are defined above are used in EBR1553 BC HUB in Link mode to define the common RT address of all RTs.
Tx(N-1..0)	Out	Up to 31 outputs to the RS485 transmit line. N defined during synthesis.
DE(N-1..0)	Out	Up to 31 outputs to the RS485 Date Enable line. Active high. '0' transmits Z. N defined during synthesis.
Rx(N-1..0)	In	Up to 31 inputs from the RS485 receive line. N defined during synthesis.

The following are debug pins used for the bring-up stage of the core. All are outputs of the core.

Signal Name	Debug Pins Description
DEBUG_TT_MSB	The MSB of the time tag register. If the TT resolution is 64 microseconds, then this signal is a 0.25 Hz clock. 2 seconds on, 2 seconds off. If this signal does not work, the IP core is reset. Check RAW_ResetN and verify it is logic high.
DEBUG_F500Kz	500KHz clock. If this signal is not 500KHz then the clock fed to the BRM1553D is not the correct clock. Each net list is synthesized to a specific clock. Please see the documentation to verify the target clock for this core.
DEBUG_Rx_Strobe	For every 1553 word on bus A OR bus B, this pulse is expected during the end of the parity bit. If this signal does not pulse and 1553 words are on the bus, it means that the plus and minus or Rx and RxN signals are not connected to the 1553 bus. This signal should pulse even if the receive lines are swapped! See DEBUG_A_Command_sync for polarity check.
DEBUG_Rx_OK	This signal is the bus A OR bus B decoder error flag. If there is a sync error (inverted sync is not an error) or bi-phase error, or parity error (inverted rx lines makes a parity error!) this signal goes low. Normally for valid words, this signals will always stay high. If there is a bus coupling issue (missing 78 Ohm termination), this signal will go low about 2 us after the end of the 1553 word.
DEBUG_A_Command_sync	This is the command sync for bus **A only**! Its should be set (or stay) high just after the mid sync of the command and status words, and go low just after the mid sync of the data words on the bus A. If it goes low after command mid sync, you have the two receive lines swapped (bus A).

The following is a licensing pin used for enabling the IP in case it exists in the pin out.

Signal Name	Pins Description
bidir_pin	This is a bi-directional pin used by the IP core to connect with the licensing device, if it exists. Please refer to the license key user manual for details.

2.4 READ AND WRITE CYCLE TIMING

All cycles are synchronous to the rising edge of the clock. **The user code interfacing with the IP core must write and read with the same clock that is being fed into the core**; otherwise the core may not accept the read and write operations. Burst read and write are supported, every clock new Address.

2.4.1 WRITE CYCLE

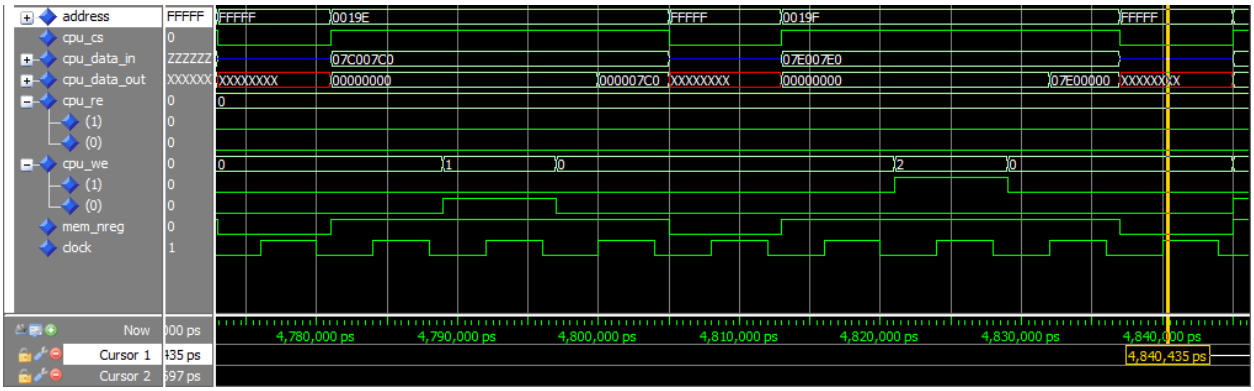

Figure 3: Write Cycle

As seen above, the signal CPU_CS must be high when CPU_WE is high, and the data that is present on the CPU_Data_in bus will be written into the core on the rising edge of the clock. The point of write is marked with the yellow cursor.

CPU_Data_out can be ignored in the write cycle. Nevertheless, after the write edge of the data, this data is available on the CPU_Data_out. This is the nature of the internal RAM inside the core.

Write burst is supported. You may burst any length of write, change address and data every clock.

2.4.2 READ AND INTERRUPT CYCLES

The INTn signal is set to 500 ns pulse in Figure 4: below. It could also be set to level mode. In this mode, the signal will stay low until the read from Register 6 is finished.

The CPU_Data_out signal is updated with the addressed location, one clock after the address is presented to the lines. This one clock delay is identical to both the RAM access, and the Register's access.

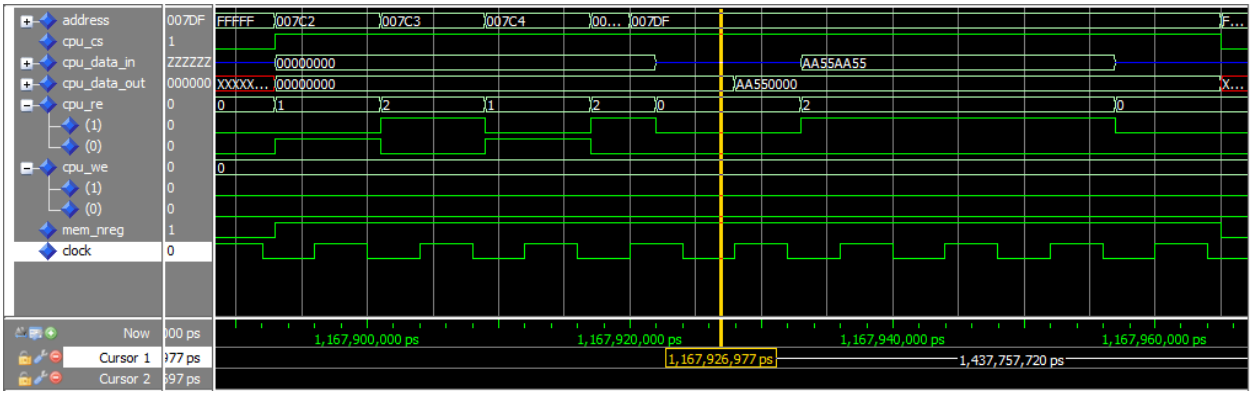

Figure 4: Read and Interrupt Cycles

2.5 HARDWARE CONNECTION TO TRANSCEIVER

The user can select any existing 1553 transceiver, or use the Sital Discrete transceiver. In case a standard transceiver is used, then each channel should be connected directly from/to the FPGA in the following manner:

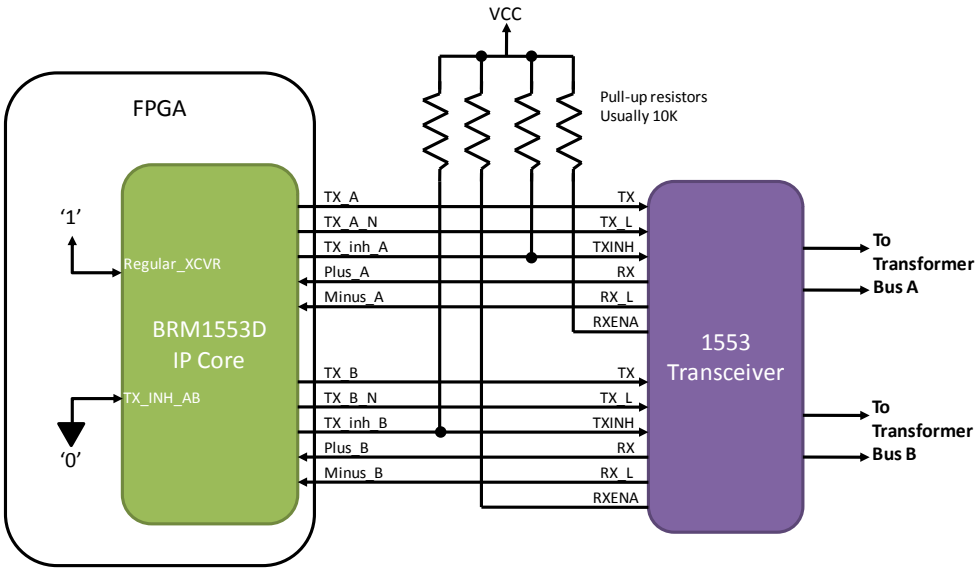

Figure 5: IP Core connection to the 1553 transceiver

In case a Sital Discrete Transceiver is used, then use the following connection:

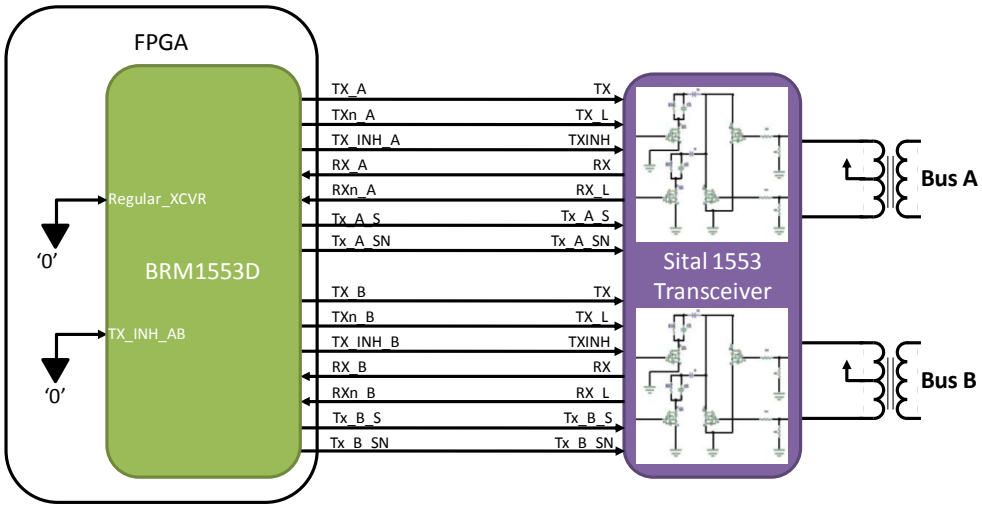

Figure 6: BRM1553D connection to Sital Discrete Transceiver

Note that VCC and logic levels must be compatible. In most cases, if the FPGA uses 3.3V, then it is best to use a 3.3V transceiver as well.

Note that the Tx_Inh outputs from the IP have different active polarity between a COTS transceiver and a Sital transceiver. For the COTS transceiver, when Regular_XCVR is high, these signal inhibit transmission when high. For Sital transceiver mode, these signals act as transmit enable signals and are logic '1' during the transition period.

3 VHDL SIMULATION TEST BENCH

3.1 TEST BENCH TOP

The core is provided with a fully functional test bench. The Test Bench block diagram is as follows:

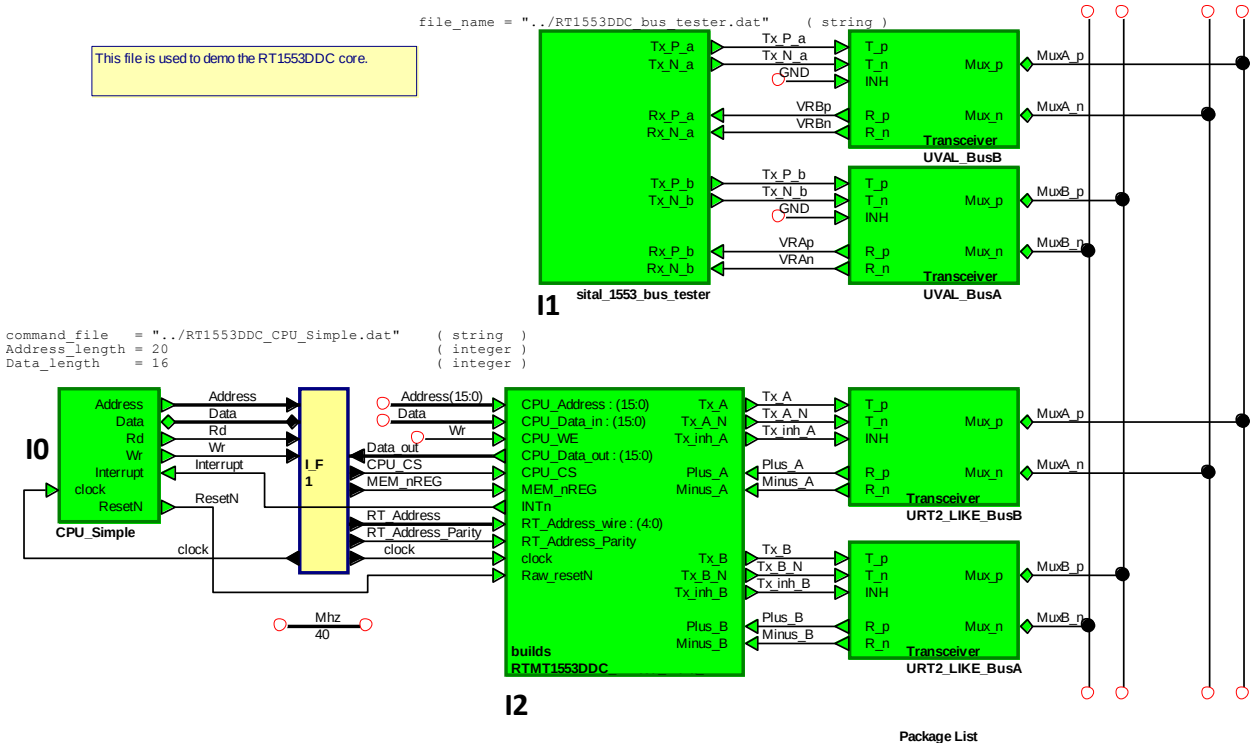

Figure 7: Simulation Test Bench

I0 is a CPU simulation model. It can performs read and write cycles based on commands defined in the "RT1553DDC_CPU_simple.dat" file which is located in the root directory. Please see available commands and examples in that file. VHDL RTL source is supplied.

I2 is the core supplied in EDIF and VHDL net list.

The four transceivers simulate the actual transceivers on the PCB, and allow many units to be connected to the bus. You may add more RTs for simulation by connecting them to the bus with these transceiver models. VHDL RTL source is supplied.

I1 is a 1553 signal generator capable of generating 1553 messages for simulation. It acts as a Bus Controller / Tester. Notice that both the transmitted words and the expected received words must be written in the "RT1553DDC_bus_tester.dat" file, else things would not work properly. Please start from the examples in the supplied file. VHDL RTL source is supplied.

The yellow block between I0 and I2 is the glue logic for generating the clocks and managing the 3-states of the data bus for the CPU.

A script for Modelsim is supplied. This script compiles everything, invokes simulation, adds signals to the wave window runs simulation and stops automatically. Please see installation root for more details on how to run it.

Note:

EBR1553 note: The same test bench is used for EBR1553 IP core with the difference that the transceivers and busses are adjusted to RS485 transmissions, as well as the bus tester being EBR1553 tester. All other commands are the same.

3.2 BUS TESTER

The bus tester is a complex VHDL design, which is able to parse and read commands from a regular text file.

The set of commands allows the composer to simulate valid and invalid MIL-STD-1553 traffic.

The Bus Tester can be instructed to transmit words onto the bus, or receive words from the bus and verify their value.

The lexical elements of the language are as follows:

#	A commented line starts with the # character.
<word>	is a 4 nibble element such as AB01.
<1553 *word*>	can be described in one of the following options: C <TA T/R SA WC> Command sync with command data built from Terminal Address Transmit or Receive, Sub Address, and Word Count all in Hex. S <word> Status Sync (same as command sync) with a 4 nibble word. D <word> Data sync with a 4 nibble word.
<*message*>	Could be one or more <1553 word> elements. Message example: C 18 R 01 03 D 2222 D 3333 D 1234 S C000
<status>	Could be of the form： CS Clear Status NR No Response

Language supported commands:

TxA <*message*>	Transmit *message* on bus A
TxB <*message*>	Transmit *message* on bus B
ReceiveA <*message*>	receives a message on bus A.
ReceiveB <*message*>	receives a message on bus B.
checkA <*status*>	Checks Either Clear Status or No Response
CheckB <*status*>	Checks Either Clear Status or No Response
WAIT_UNTIL	RxA A word is received fully on bus A RxB A word is received fully on bus B TxA_Rdy The A transmitter double buffer is ready for another transmit TxB_Rdy The B transmitter double buffer is ready for another transmit example : WAIT_UNTIL RxA will wait until a word was received from bus A
WAIT_FOR <*time*>	Wait for time delay. Time delay could be 1 ns or 23 us or 3456 ns no decimal point in time unit
echo	40 characters echoed to screen
break	stops the program.

The commands should be located in a file called **bus_teser.dat** located in the hds2004 sub directory.

A sample bus_tester.dat file is supplied with the core, please read through and edit to your desired commands.

A simple program would look like this:

```
WAIT_FOR 2 us

ECHO transmit a receive command with two data words to the UUT.
TxA  C 18 R 01 02  D 0000  D 0001

 # wait until the data word is transmitted + 2 us.
WAIT_FOR 22 us

ECHO  wait for a reply
WAIT_UNTIL RxA

# check that the 10 LSBs of the received status is Clean Status.
CHECKA CS

# allow for an intermessage gap to the next command.

WAIT_FOR 6 us
```

APPENDIX A: CHANGES TRACKING

Date	Revision	Change	By
January '11	6.7	United manuals for BC/RT/MT Split manuals between HW and HW/SW interface	Duli
April '11	6.71	Made format updates on the document	Duli
January '12	6.8	Added Extended Bit Rate Support	Ofer H.
July 2015	7.1	Added 32 bit bus signaling	Ofer H.

17 Atir Yeda St., Kfar-Saba, ISRAEL 44643

Email: info@sitaltech.com

Website: http://www.sitaltech.com

The information provided in this User's Guide is believed to be accurate; however, no responsibility is assumed by Sital Technology for its use, and no license or rights are granted by implication or otherwise in connection therewith. Specifications are subject to change without notice.

Please visit our Web site at www.sitaltech.com for the latest information.

© All rights reserved. No part of this User's Guide may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission by Sital Technology.