# Core1553BRT-EBR Enhanced Bit Rate 1553 Remote Terminal

DirectCore

## Product Summary

### Intended Use

- 1553 Enhanced Bit Rate Remote Terminal (RT)
- DMA Backend Interface to External Memory
- Direct Backend Interface to Devices
- Space and Avionic Applications

### Key Features

- Supports Enhanced Bit Rate 1553
- 10 Mbps Time-Multiplexed Serial Data Bus
- Interfaces to External RAM or Directly to Backend Device
- Synchronous or Asynchronous Backend Interface
- Encoders and Decoders Operate off 100 MHz Clock
- Protocol Control and Memory Interface Operates off 50 MHz Clock
- Interfaces to Standard RS485 Transceivers
- Programmable Mode Code and Sub-Address Legality for Illegal Command Support
- Memory Address Mapping Allowing Emulation of Legacy Remote Terminals
- Fail-Safe State Machines
- Fully Synchronous Operation

### Supported Families

- ProASIC®3/E
- ProASIC PLUS®
- Axcelerator®
- RTAX-S

### Core Deliverables

- Netlist Version
  - Compiled RTL Simulation Model, Compliant with Actel Libero® Integrated Design Environment (IDE)
  - Netlist Compatible with the Actel Designer Place-and-Route Tool (with and without I/O Pads)
- RTL Version
  - VHDL or Verilog Core Source Code
  - Synthesis Scripts
- Actel-Developed Testbench (VHDL)

### Development System

- Complete 1553BRT-EBR Implementation, Implemented in an AX1000

### Synthesis and Simulation Support

- Synthesis: Exemplar™, Synplicity®, Design Compiler®, FPGA Compiler™
- Simulation: Vital-Compliant VHDL Simulators and OVI-Compliant Verilog Simulators

### Verification and Compliance

- Meets Requirements of Draft SAE AS5682 Standard (2005-10)
- Actel-Developed Simulation Testbench Implements a Subset of the RT Test Plan (MIL-HDBK-1553A) for Protocol Verification
- Protocol Control Derived from Core1553BRT, which Is Certified to MIL-STD-1553B (RT Validation Test Plan MIL-HDBK-1553, Appendix A)

### Contents

# General Description

Core1553BRT-EBR provides a complete, dual-redundant 1553 enhanced bit rate (EBR) remote terminal (RT) apart from the transceivers required to interface to the bus. A typical system implementation using the Core1553BRT-EBR is shown in Figure 1 and Figure 2 on page 3.

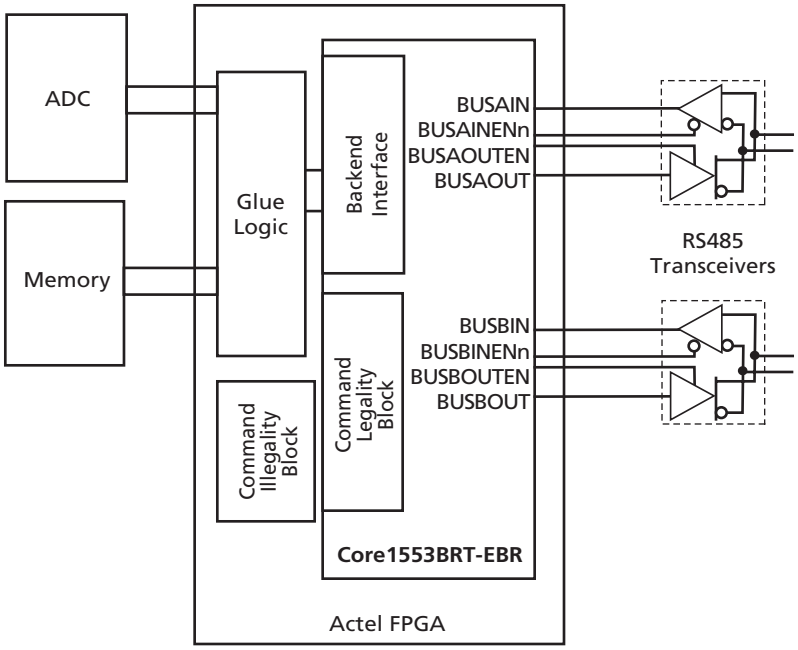



*Figure 1* • **Typical Core1553BRT-EBR System**

At a high level, Core1553BRT-EBR simply provides a set of memory mapped sub-addresses that "receive data written to" or "transmit data read from." The core can be configured to directly connect to synchronous or asynchronous memory devices. Alternately, the core can directly connect to the backend devices, removing the need for the memory buffers. If memory is used, the core requires 2,048 words of memory, which can be shared with the local CPU.

The core supports all 1553EBR mode codes and allows the user to designate as illegal any mode code or any particular sub-address for both transmit and receive operations. The command legalization can be done within the core or in an external command legality block via the command legalization interface.

The core consists of six main blocks: 1553EBR encoders, 1553EBR decoders, backend interface, command decoder, RT controller blocks, and a command legalization block (see Figure 2).

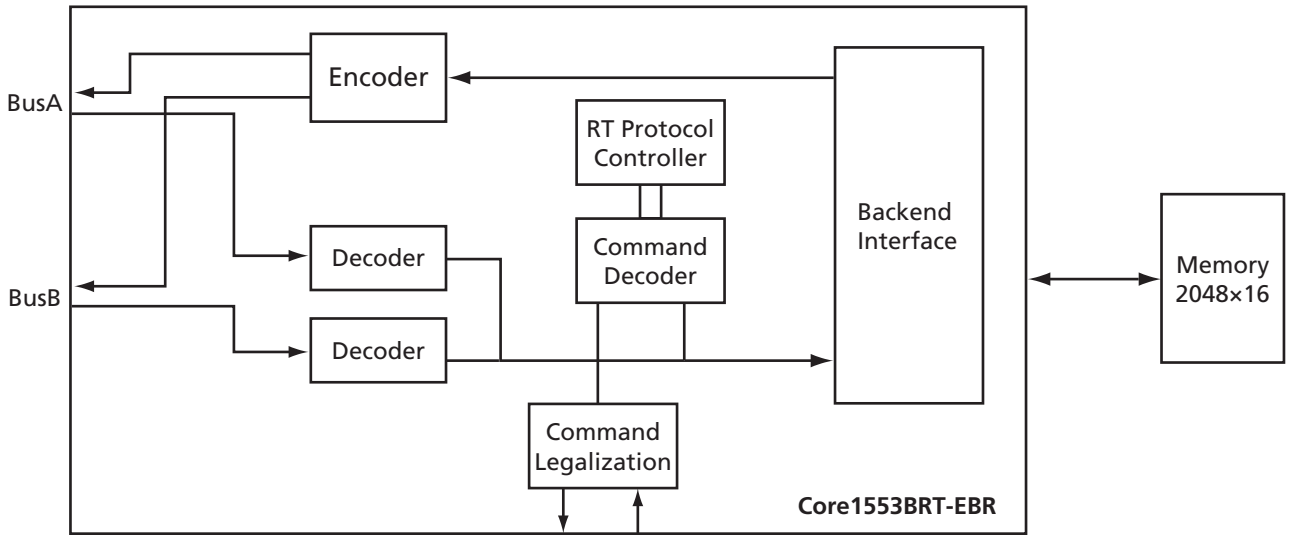



*Figure 2* • **Core1553BRT-EBR RT Block Diagram**

In the Core1553BRT-EBR, a single 1553EBR encoder is used. This takes each word to be transmitted and serializes it, after which the signal is Manchester encoded. The encoder also includes both logic to prevent the RT from transmitting for greater than the allowed period and loopback fail logic. The loopback logic monitors the received data and verifies that the core has correctly received every word that it transmits.

The output of the encoder is gated with the bus enable signals to select which buses the RT should use to transmit.

The core includes two 1553EBR decoders. The decoder takes the serial Manchester data received from the bus and extracts the received data words. The decoder requires a 100 MHz clock to extract the data and the clock from the serial stream.

The decoder contains a digital phased-lock loop (PLL) that generates a recovery clock used to sample the incoming serial data. The data is then deserialized and the 16-bit word decoded. The decoder detects whether a command or data word is received, and also performs Manchester encoding and parity error checking.

The backend interface for the Core1553BRT-EBR allows a simple connection to a memory device or direct connection to other devices, such as analog-to-digital converters. The access rates to this memory are slow, with one read or write every 2 µs. The backend interface operates off the internally derived 50 MHz clock, resulting in a read or write every 100 clock cycles.

The backend interface can be configured to connect to either synchronous or asynchronous memory devices. This allows the core to be connected to synchronous logic, memory within the FPGA, or external asynchronous memory blocks.

The core implements a simple sub-address to the memory address mapping function, allowing the core to be directly connected to a memory block. The core also supports an address mapping function that allows the backend memory map to be modified to emulate legacy 1553EBR remote terminals, therefore minimizing system and software changes when adopting the Core1553BRT-EBR. Associated with this function is the ability to create a user-specific interrupt vector.

The backend interface supports a standard bus request and grant protocol, and provides a WAIT input to allow the core to interface to slow memory devices.

The command decoder and RT controller blocks decode the incoming command words, verifying their legality. The protocol state machine then responds to the command, transmitting or receiving data or processing a mode code.

The Core1553BRT-EBR has an internal command legality block that verifies every 1553EBR command word. A separate interface is provided that, when enabled, allows the command legality decoder to be implemented outside the Core1553BRT-EBR. This external interface is intended for use with netlist versions of the core. For the RTL version of the core, this interface can be used or the source code can be modified easily to implement this function.

## Core1553BRT-EBR Device Requirements

The Core1553BRT-EBR can be implemented in several Actel FPGA devices. Table 1 gives the utilization and performance figures for the core implemented in these devices.

The core can operate with a clock of up to 24 MHz. This clock rate is easily met in all Actel silicon families noted in Table 1.

## Core1553BRT-EBR Verification and Compliance

The Core1553BRT-EBR functionality has been verified in simulation and hardware.

To fully verify compliance, the core has been implemented on AX1000 and ProASIC3 parts connected to external transceivers and memory.

## Core1553BRT-EBR Fail-Safe State Machines

The logic design of Core1553BRT-EBR implements fail-safe state machines. All state machines include illegal state detection logic. If a state machine should ever enter an illegal state, the core will assert its FSM_ERROR output and the state machine will reset. If this occurs, Actel recommends that the external system reset the core and also assert the TFLAG input to inform the bus controller that a serious error has occurred within the remote terminal.

The FSM_ERROR output can be left unconnected if the system is not required to detect and report state machines entering illegal states.

## Enhanced Bit Rate 1553 Bus Overview

Enhanced Bit Rate 1553 is a enhanced data rate MIL-STD-1553B bus. The data transmission rate has been increased from 1 MB/Sec to 10 MB/Sec, and the multi-drop bus structure has been replaced with a hub-based point-to-point bus structure. To maintain system compatibility, the data protocol and command, status, and data words are identical to the MIL-STD-1553B specification.

The bus has a single active bus controller (BC) and up to 31 remote terminals (RTs). For 1553EBR, the BC has up to 31 separate transceivers, each one connected directly to an RT. The BC manages all data transfers on the bus using the command and status protocol. The bus controller initiates every transfer by sending a command word and data if required. The selected RT will respond with a status word and data if required.

The 1553EBR command word contains a five-bit RT address, transmit or receive bit, five-bit sub-address, and five-bit word count. This allows for 32 RTs on the bus. However, since RT address 31 is used to indicate a broadcast transfer, only 31 RTs may be connected. Each RT has 30 sub-addresses reserved for data transfers. The other two sub-addresses (0 and 31) are reserved for mode codes used for bus control functions. Data transfers contain up to 32 16-bit data words. Mode code command words are used for bus control functions, such as synchronization.

*Table 1* • **Device Utilization**

| Family | Comb. | Seq. | Total | Device | Utilization | Performance |
|---|---|---|---|---|---|---|
| ProASIC3/E | 970 | 467 | 1437 | A3P250 | 24% | 115/55 MHz |
| ProASIC$^{PLUS}$ | 1298 | 467 | 1765 | APA150 | 29% | 105/55 MHz |
| Axcelerator | 658 | 463 | 1121 | AX500 | 14% | 173/87 MHz |
| RTAX-S | 658 | 463 | 1121 | RTAX250S | 27% | 126/62 MHz |

**Note:** The Performance column shows the maximum clock speed for the 100 MHz and 50 MHz clock domains for each FPGA family.

## Message Types

The 1553EBR bus supports eight message transfer types, allowing basic point-to-point and broadcast BC-to-RT data transfers, as well as mode code messages. Figure 3 shows the message formats.

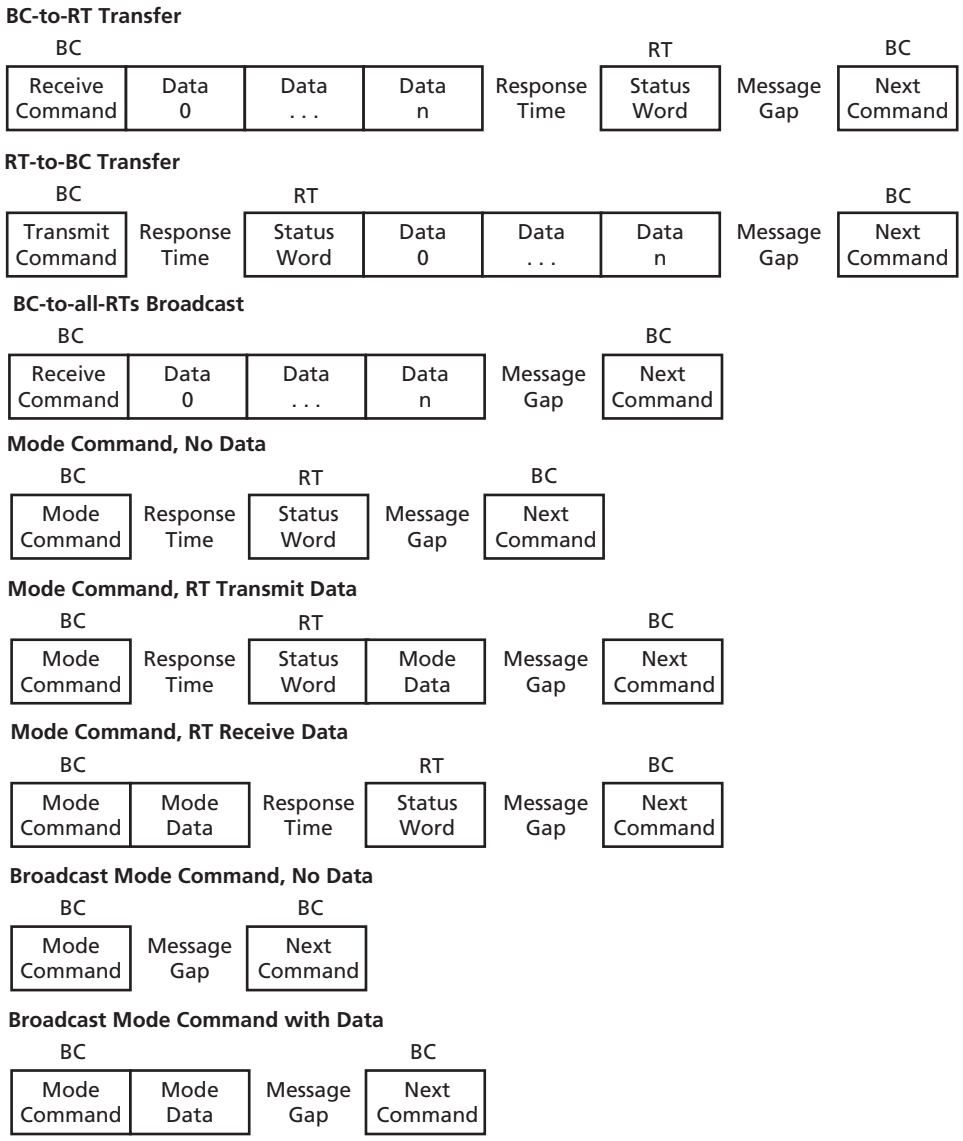



*Figure 3* • **1553EBR Message Formats**

## Word Formats

There are only three types of words in a 1553EBR message: a command word (CW), a data word (DW), and a status word (SW). Each word consists of a 3-bit sync pattern, 16 bits of data, and a parity bit, providing the 20-bit word (see Figure 4).

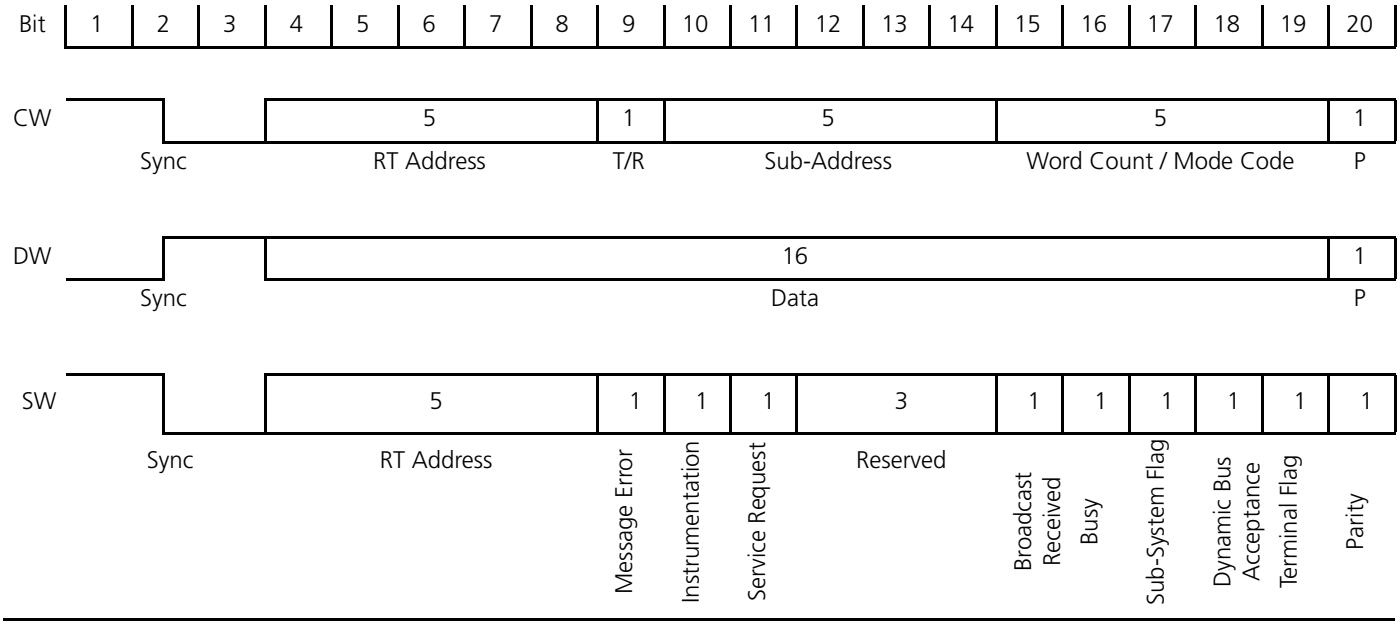



*Figure 4* • **1553EBR Word Formats**

# I/O Signal Descriptions

*Table 2* • **1553EBR Bus Interface**

| Port Name | Type | Description |
|---|---|---|
| RTADDR[4:0] | In | Sets the RT address; must not be set at '11111' |
| RTADDRP | In | RT Address parity input. This input should be set high or low to achieve odd parity on the RTADDR and RTADDRP inputs. If RTADDR is set to '00000', the RTADDRP input should be set to '1'. |
| RTADERR | Out | Indicates that the RTADDR and RTADDRP inputs have incorrect parity, or broadcast is enabled and the RT address is set to 31; when active (high), the RT is disabled and will ignore all 1553EBR traffic. |
| BUSAINENn | Out | Active low enable for the A receiver |
| BUSAIN | In | Data input from the A receiver |
| BUSBINENn | Out | Active low enable for the B receiver |
| BUSBIN | In | Data input from the B receiver |
| BUSAOUTEN | Out | Active high transmitter enable for the A transmitter |
| BUSAOUT | Out | Data output to the Bus A transmitter |
| BUSBOUTEN | Out | Active high transmitter enable for the B transmitter |
| BUSBOUT | Out | Data output to the Bus B transmitter |

*Table 3* • **Control and Status Signals**

| Port Name | Type | Description |
|---|---|---|
| CLK | In | Master 100 MHz clock input |
| CLKOUT100 | Out | 100 MHz clock input routed to an output pin |
| CLKOUT50 | Out | 50 MHz clock used to clock the protocol and memory interface blocks. All core outputs are synchronized to this clock. Will be routed on a global network. |
| RSTn | In | Asynchronous reset input (active low) |
| SREQUEST | In | Directly controls the service request bit in the 1553EBR status word |
| RTBUSY | In | Directly controls the busy bit in the 1553EBR status word |
| SSFLAG | In | Directly controls the sub-system flag bit in the 1553EBR status word |
| TFLAG | In | Controls the sub-system flag bit in the 1553EBR status word. Can be masked by the "inhibit terminal flag bit" mode code. |
| VWORD[15:0] | In | Provides the 16-bit vector value for the "transmit vector word" mode command |
| BUSY | Out | Indicates that the core is either receiving or transmitting data or handling a mode command |
| CMDSYNC | Out | Pulses high for a single clock cycle when the RT detects the start of a 1553EBR command word (or status word) on the bus. Provides an early signal that the RT may be about to receive or transmit data or mode code. |
| MSGSTART | Out | Pulses high for a single cycle when the RT is about to start processing a 1553EBR message whose command has been validated for this RT. |
| SYNCNOW | Out | Pulses high for a single clock cycle when the RT receives a "synchronize" command with or without data mode. The pulse occurs just after the 1553EBR command word (sync with no data) or data word (sync with data mode code) has been received. |
| BUSRESET | Out | Pulses high for a single clock cycle whenever the RT receives a reset mode command. The core logic will also automatically reset itself on receipt of this command. |
| INTOUT | Out | Goes high when data has been received or transmitted or a mode command processed. The reason for the interrupt is provided on INTVECT. Will stay high until INTACK goes high. If INTACK is held high, will pulse high for a single clock cycle. |
| INTVECT[6:0] | Out | A seven-bit value containing the reason for the interrupt. Indicates which sub-address data has been received or transmitted.<br>Bit 6 0: Bad block received 1: Good block received<br>Bit 5 0: RX data 1: TX data<br>Bits 4:0 Sub-address<br>Further information can be found by checking the appropriate transfer status word for the appropriate sub-address. |
| INTACK | In | Interrupt acknowledge input. When high, resets INTOUT to low. If this input is held high, the INTOUT signal will pulse high for one clock cycle every time an interrupt is generated. |
| MEMFAIL | Out | Goes high if the core fails to read or write data to the backend interface within the required time. This can be caused by the backend failing to assert MEMGNTn fast enough or asserting MEMWAITn for too long. |
| CLRERR | In | Used to clear MEMFAIL and other internal error conditions. Must be held high for more than two clock cycles. |

**Note:** All control inputs except RSTn are synchronous and sampled on the rising edge of the internally generated 50 MHz clock (CLKOUT50). All status outputs are synchronized to the rising edge of the same clock.

## Command Legalization Interface

The core checks the validity of all 1553EBR command words. In RTL and netlist versions of the core, the logic may be implemented externally to the core. The command word is provided, and the logic must generate the command valid input. The command legalization interface also provides two strobes that are used to latch the command value to enable it to be used for address mapping and interrupt vector extension functions (Table 4).

## Backend Interface

The backend interface supports both synchronous operation (to the core clock) and asynchronous operation to backend devices (Table 5 on page 9).

*Table 4* • **Command Legalization Interface**

| Port Name | Type | Description |
|---|---|---|
| USEEXTOK | In | When '0', the core uses its own internal command valid logic, enabling all legal supported mode codes and all sub-addresses.<br>When '1', the core disables its internal logic and uses the external CMDOKAY input for command legality. |
| CMDVAL[11:0] | Out | Active Command<br>11 0: Non-broadcast<br>1: Broadcast<br>10 0: Receive<br>1: Transmit<br>[9:5] Sub-address<br>[4:0] Word count / mode code<br>These outputs are valid throughout the complete 1553EBR message. They can also be used to steer data to particular backend devices. In particular, bit 11 allows non-broadcast and broadcast messages to be differentiated. |
| CMDSTB | Out | Single clock cycle pulse that indicates CMDVAL has changed |
| CMDOKAY | In | Command word is okay (active high). The external logic must set this within 2 µs from the CMDVAL output changing. |
| CMDOKOUT | Out | Command word is okay output. When USEEXTOK = '0', the core outputs its internal command word okay validation signal. |
| ADDRLAT | Out | CMDVAL address latch enable output (active high) is used to latch the CMDVAL when it is being used for an address mapping function. ADDRLAT should be connected to the enable of a rising edge clock flip-flop. |
| INTLAT | Out | CMDVAL interrupt vector latch enable output (active high) is used to latch the CMDVAL when it is being used for an extended interrupt vector function. INTLAT should be connected to the enable of a rising edge clock flip-flop. |

*Table 5* • **Backend Signals**

| Port Name | Type | Description |
|---|---|---|
| MEMREQn | Out | Memory Request (active low) output. The backend interface requires memory access completion within 1 µs of MEMREQ going low to avoid data loss or overrun on the 1553EBR interface.* |
| MEMGNTn | In | Memory Grant (active low) input. This input should be synchronous to CLK and needs to meet the internal register setup time. This input may be held low if the core has continuous access to the RAM. |
| MEMWRn | Out | Memory Write (active low)<br>Synchronous mode: This output indicates that data is to be written on the rising clock edge.<br>Asynchronous mode: This output will be low for a minimum of one clock period and can be extended by the MEMWAITn input. The address and data are valid one clock cycle before MEMWRn is active and held for one clock cycle after MEMWRn goes inactive. |
| MEMRDn | Out | Memory Read (active low)<br>Synchronous mode: This output indicates that data will be read on the next rising clock edge. This signal is intended as the read signal for synchronous RAMs.<br>Asynchronous mode: This output will be low for a minimum of one clock period and can be extended by the MEMWAITn input. The address is valid one clock cycle before MEMRDn is active and held for one clock cycle after MEMRDn goes inactive. The data is sampled as MEMRDn goes high. |
| MEMCSn | Out | Memory Chip Select (active low). This output has the same timing as MEMADDR. |
| MEMWAITn | In | Memory Wait (active low)<br>Synchronous mode: This input is not used; it should be tied high.<br>Asynchronous mode: Indicates that the backend is not ready and that the core should extend the read or write strobe period. This input should be synchronized to CLK and needs to meet the internal register setup time. It can be permanently held high. |
| MEMOPER[1:0} | Out | Indicates the type of memory access being performed<br>00: Data transfer for both data and mode code transfers<br>01: TSW<br>10: Command word<br>11: Not used |
| MEMADDR[10:0] | Out | Address (active low). Memory address output (The sub-address mapping is covered in the memory allocation section) |
| MEMDOUT[15:0] | Out | Memory Data output (active low) |
| MEMDIN[15:0] | In | Memory Data input (active low) |
| MEMCEN | Out | Control Signal Enable (active high). This signal is high when the core is requesting the memory bus and has been granted control. It is intended to enable any tristate drivers that may be implemented on the memory control and address lines. |
| MEMDEN | Out | Data Bus Enable (active high). This signal is high when the core is requesting the memory bus, has been granted control, and is waiting to write data. It is intended to enable any bidirectional drivers that may be implemented on the memory data bus. |

**Note:** *The 1 µs refers to the time from MEMREQn being asserted to the core deasserting its MEMREQn signal. The core has an internal overhead of five clock cycles, and any inserted wait cycles will also reduce this time.

## Miscellaneous I/O

Several inputs are used to modify the core functionality to simplify integration in the application. These inputs should be tied to logic '0' or logic '1', as appropriate (Table 6).

## Standard Memory Address Map

Core1553BRT-EBR requires an external 2,048×16 memory device. This memory is split into 64 32-word data buffers.

Each of the 30 sub-addresses has a receive and a transmit buffer, as shown in Table 7 on page 11.

The memory allocated to the unused receive sub-addresses 0 and 31 is used to provide status information back to the rest of the system. At the end of every transfer, a transfer status word (TSW) is written to these locations.

*Table 6* • **Miscellaneous I/Os**

| Port Name | Type | Description |
|---|---|---|
| WRTCMD | In | When '1', the core will write the 1553EBR command word to the locations used for the TSW values. If WRTTSW is also enabled, then the command word is written to memory at the start of a message and the TSW value will overwrite the command word at the end of the message, unless an external address mapping function is used. |
| WRTTSW | In | When '1', the core will write the transfer status word to the memory.<br>When '0', the core disables the writing of the transfer status word to memory. This is useful for simple RT applications that do not use memory but have a direct connection to the backend device. |
| EXTMDATA | In | When '1', the core reads and writes mode code data words to and from the external memory (except for the transmit last command and transmit BIT word). The VWORD input is not used when this input is active. |
| INTENBBR | In | When active '1', the core generates interrupts when both good and bad 1553EBR messages are received. When inactive '0', the core only generates interrupts when good messages are received. |
| ASYNCIF | In | When '1', the backend interface is in asynchronous mode.<br>When '0', the backend interface is in synchronous mode. |
| TESTTXTOUT | In | This input is for test use only. It should be tied low.<br>When high, the RT will transmit more than 32 data words if a transmit data command word is received. This will cause the RT to shut down the transmitter and set the TIMEOUT bits in the BIT word. |
| BCASTEN | In | This input enables broadcast operation.<br>When '1', broadcast operations are enabled.<br>When '0', broadcast messages (i.e., RT Address 31) are treated as normal messages. If the RTADDR input is set to 31, then the RT will respond to the message. |
| SA30LOOP | In | This input alters the backend memory mapping so that sub-address 30 provides automatic loopback (Table 7 on page 11).<br>When '0', the RT does not loopback sub-address 30. Separate memory buffers are used for transmit and receive data buffers.<br>When '1', the RT maps the transmit memory buffer for sub-address 30 to the receive memory buffer for sub-address 30, i.e., the upper address line is forced to '0'. |
| FSM_ERROR | Out | This output will go high for a single clock cycle if any of the internal state machines enter an illegal state. This output should not go high in normal operation. Should it go high, it is recommended that the core be reset. |

*Table 7* • **Standard Memory Address Map**

| Address | RAM Contents | Notes |
|---|---|---|
| 000–01F | RX transfer status words | The core only writes to these addresses (except when SA30LOOP is high). |
| 020–03F | Receive sub-address 1 | |
| | … | |
| 3C0–3DF | Receive sub-address 30 | |
| 3E0–3FF | TX transfer status words | |
| 400–41F | Not used | The core only reads from these addresses. |
| 420–43F | TX transfer sub-address 1 | |
| | … | |
| 7C0–7DF | TX transfer sub-address 30 | |
| 7E0–7FF | Not used | |

If the SA30LOOP input is set high, the RT maps transmit sub-address 30 to the receive sub-address 30, i.e., the upper address bit is forced to '0'. This provides a loopback sub-address as per MIL-STD-1553EBR, Notice 2. The TSW is still written to address 03EE. It should be noted that this is not strictly compliant with the specification since the transmit buffer will contain invalid data if the received command fails, e.g., on a parity error. The transmit buffer should only be updated if the receive command had no errors. To implement this function in full compliance with the specification, the SA30LOOP input should be tied low, and the RT backend should copy the receive memory buffer to the transmit memory buffer only after the RT signals that the message was received with no errors.

When the memory buffer is implemented within the FPGA device using dual-port RAMs, separate receive and transmit RAM blocks can be used (each as 1 k words), as shown in Figure 5. In these cases, the RX memory is selected when A10 = 0 and the TX memory when A10 = 1. In this case, the SA30LOOP input must be tied low.

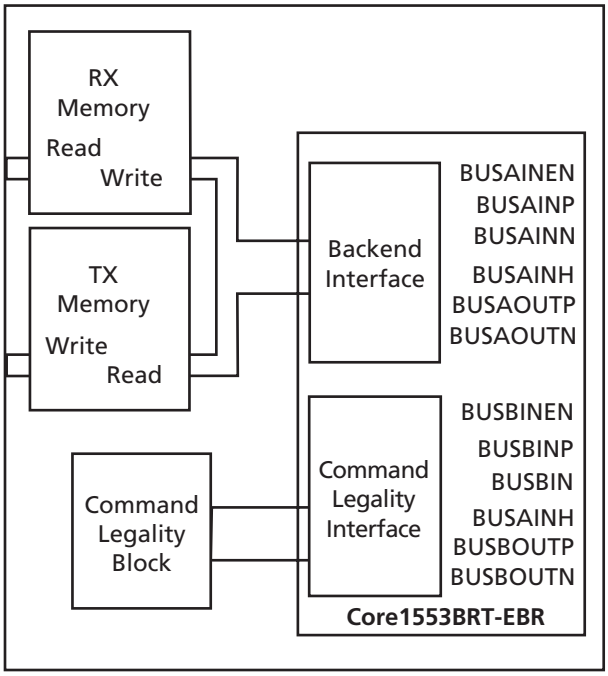



*Figure 5* • **Using Internal FPGA Memory Blocks**

## Memory Address Mapping

The core supports an external memory address mapper that allows the RT memory allocation to be easily customized. To use this function, the CMDVAL output must be latched by the ADDRLAT signal, as shown in Figure 6. Then the address mapper function can map the 1553EBR command words, data words including mode code data, and transfer status words to any memory address.

## Interrupt Vector Extension

The core generates a seven-bit interrupt vector that contains the sub-address and whether it was a transmit or receive message. Some systems may need to include whether the message was broadcast, a mode code, or the actual word count in the interrupt vector. The core supports an interrupt vector extension function, similar to the address mapper function using the INTLAT signal, as shown in Figure 7.

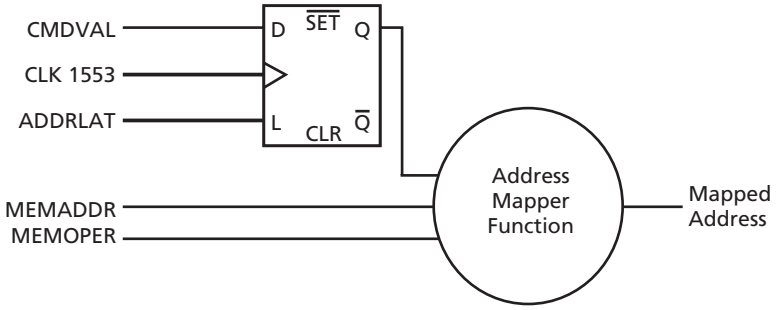



*Figure 6* • **Memory Address Mapping**

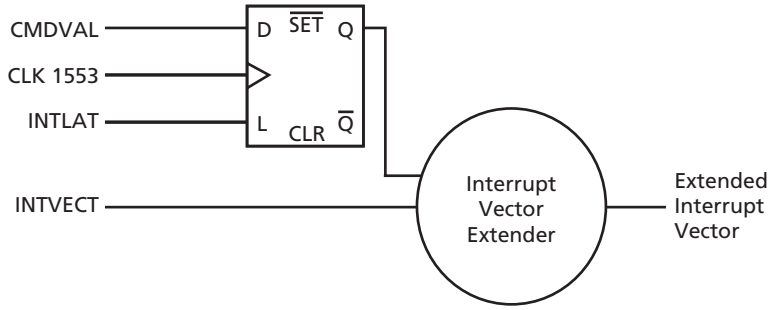



*Figure 7* • **Interrupt Vector Extension**

## Status Word Settings

The Core1553BRT-EBR sets bits in the 1553EBR status word in compliance with MIL-STD-1553B. This is summarized in Table 8.

*Table 8* • **Status Word Bit Settings**

| Bit(s) | Function | Setting |
|---|---|---|
| 15:11 | RT Address | Equals the RTADDR input |
| 10 | Message Error | Set whenever the RT detects a message error |
| 9 | Instrumentation | Always '0' |
| 8 | Service Request | Controlled by the SSFLAG input |
| 7:5 | Reserved | Always '000' |
| 4 | Broadcast Received | Set whenever a broadcast message is received |
| 3 | Busy | Controlled by the RTBUSY input |
| 2 | Sub-System Flag | Controlled by the SSFLAG input |
| 1 | Dynamic Bus Acceptance | Always '0'. The Core1553BRT-EBR does not operate as a bus controller. |
| 0 | Terminal Flag | Controlled by the TFLAG input. If an "inhibit terminal flag" mode code is in effect, will be '0'. |

## Command Word Storage

At the start of every 1553EBR bus transfer, the 1553EBR command word is written to RAM locations 000–01F for receive operations and 3E0–3FF for transmit operations. The addresses are as follows:

CMD location RX commands: '000000' and SA

CMD location TX commands: '011111' and SA

If the RT is implemented without a memory-based backend, the writing of the command word can be disabled (WRTCMD input). This simplifies the design of the backend logic that directly controls the backend function.

## Transfer Status Words (TSW)

At the end of every 1553EBR bus transfer, a transfer status word is written to RAM in locations 000–01F for receive operations and 3E0–3FF for transmit operations. The addresses used are as follows:

TSW location RX commands: '000000' and SA

TSW location TX commands: '011111' and SA

As an example, the TSW address for a transmit command with sub-address 24 would be '01111110100' (3F4h). The TSW contains the information in Table 9 on page 14.

If the RT is implemented without a memory-based backend, the writing of the TSW can be disabled. This simplifies the design of the backend logic that directly controls backend functions.

## Backend Access Times

During normal operation, the backend must allow a memory access to complete within 1.0 µs.

While the status word is being transmitted, the core must write the command word to memory and fetch the first data word. Two memory accesses are performed in the 2 µs that the status word takes to transmit.

At the end of a broadcast-receive command, Core1553BRT-EBR writes the last data word and the TSW value before the RT decodes the next command. Two memory accesses occur in the 2 µs during which the command word is being decoded.

The core includes a timer that is set to terminate backend memory access at 1.0 µs.

*Table 9* • **Transfer Status Word**

| Bit(s) | Name | Description | |
|---|---|---|---|
| 15 | USED | This bit is set to '1' at the end of the transmit or receive command. | |
| 14 | OKAY | Indicates that no errors are detected, i.e., bits 11 to 5 are all '0' | |
| 13 | BUSN | Indicates on which bus the command was received<br>'0': BUSA '1': BUSB | |
| 12 | BROADCAST | Indicates a broadcast command | |
| 11 | LPBKERRB | Indicates that the loopback logic detected an error on the transmitted data for bus B | |
| 10 | LPBKERRA | Indicates that the loopback logic detected an error on the transmitted data for bus A | |
| 9 | ILLEGAL CMD | The command was illegal. Either a request to transmit from an illegal sub-address or an illegal mode code was received. | |
| 8 | MEMIFERR | Indicates that the DMA memory access failed to complete quickly enough | |
| 7 | MANERR | Indicates that a Manchester encoding error was detected in the incoming data | |
| 6 | PARERR | Indicates that a parity error was detected in the incoming data | |
| 5 | WCNTERR | Indicates that an incorrect number of words was received | |
| 4:0 | COUNT | SA1 to SA31 | Indicates the number of words received or transmitted for that sub-address. If WCNTERR is '0', '00000' indicates 32 words. Otherwise, '00000' indicates zero words transferred. |
| | | SA0 or SA31 | Indicates which mode code was received or transmitted per the 1553EBR specification |

# 1553BRT-EBR Operation

## Data Transfers – Receive

When a receive data transfer command is detected, the core will decode each incoming word. At the end of each word, the core will assert MEMREQn. When MEMGNTn goes low, it will write the data word to the memory and release the MEMREQn. This process is repeated until the correct number of words has been transferred. The core will then transmit its 1553EBR status word. Finally, the TSW is also written to memory.

## Data Transfers – Transmit

When a transmit data transfer command is detected, the core will transmit its status word and assert MEMREQn. When MEMGNTn goes low, it will read a data word from the memory and release the MEMREQn. Once the word is available, the core will transmit the data word. The core will continue to request data from the memory interface until the required number of words has been transferred. Finally, the TSW is written to memory.

## RT-to-RT Transfer Support

The 1553EBR specification (SAE AS5682) does not support RT-to-RT transfers. Likewise, Core1553BRT-EBR does not support RT-to-RT transfers.

## Mode Codes

When the core receives a mode code, it first checks its command validity. If the command is valid, it is processed in accordance with the specification. Otherwise, the message error bit will be set in the 1553EBR status word. Table 10 on page 15 lists the supported mode codes.

Two mode codes, (1) transmit a vector word and (2) synchronize with data, require external data. When EXTMDATA is inactive, the vector word value is set by the VWORD input and the synchronize with data word is discarded. When EXTMDATA is active, these values are read from and written to memory. The MEMADDR output will be similar to a single-word data transfer message. Bit 10 will reflect the command word TX bit, and bits 9:5 will be 00h or 1Fh, depending on whether the mode code sub-address is set to 0 or 31. Bits 4:0 will be zero. This implies that the vector word will be read from location 400h or 7E0h, and the synchronize with data word is written to location 000h or 3E0h, depending on whether sub-address 0 or 31 is used.

When both WRTCMD and WRTTSW are active for each message, the command word and TSW value will be written to the same location. These writes can be distinguished by the MEMOPER output. This may cause some system problems, but such can be avoided by implementing an external address mapper function to map these accesses to different addresses.

## Loopback Tests

The Core1553BRT-EBR performs loopback testing on all of its transmissions. The transmit data is fed back into the receiver and each transmitted word is compared. If an error is detected, the loopback fail bit is set in the TSW and also in the BIT word.

*Table 10* • **Supported Mode Codes**

| T/R Bit | Mode Code | Function and Effect | Data Word | Core Supports | Broadcast Allowed |
|---|---|---|---|---|---|
| 1 | 00000<br>0 | Dynamic Bus Control<br>The core does not support bus controller functions, so it will set the message error and the dynamic bus control bit in the status word. | No | No | No |
| 1 | 00001<br>1 | Synchronize<br>The core will assert its SYNCNOW output after the command word has been received. | No | Yes | Yes |
| 1 | 00010<br>2 | Transmit Status Word<br>The core retransmits the last status word. | No | Yes | No |
| 1 | 00011<br>3 | Initiate Self Test<br>The core does not support self test. Since the core supports the transmit BIT word mode code, this command is treated as legal and will not set a message error. | No | Yes | Yes |
| 1 | 00100<br>4 | Reserved | No | No | No |
| 1 | 00101<br>5 | Reserved | No | No | No |
| 1 | 00110<br>6 | Inhibit Terminal Flag<br>The core will mask the TFLAG input and the terminal flag bit in the status word will be forced to zero. | No | Yes | Yes |
| 1 | 00111<br>7 | Override Inhibit Terminal Flag<br>The core will re-enable the TFLAG input. | No | Yes | Yes |
| 1 | 01000<br>8 | Reset Remote Terminal<br>The core will assert its BUSRESET output after the command word has been received. It will also reset itself. | No | Yes | Yes |
| 1 | 10000<br>16 | Transmit Vector Word<br>The core will transmit a single data word that contains the value on the VWORD input. | Yes | Yes | No |
| 1 | 10010<br>18 | Transmit Last Command Word<br>The core will transmit a single data word that contains the last command word received. | Yes | Yes | No |
| 1 | 10011<br>19 | Transmit Bit Word<br>The core will transmit a single data word that contains the extended core status information. The value of this word is defined in Table 13 on page 18. | Yes | Yes | No |
| 0 | 10001<br>17 | Synchronize with Data<br>The core will assert its SYNCNOW output after the data word has been received. | Yes | Yes | Yes |
| 0 | 10100<br>20 | Reserved | Yes | No | No |
| 0 | 10101<br>21 | Reserved | Yes | No | No |

## Error Detection

*Table 11* • **Error Detection**

| Error Condition | Action |
|---|---|
| Command Word<br>1. Parity or Manchester encoding errors<br>2. Incorrect SYNC waveform | <br>Command is ignored<br>No interrupt generated |
| Mode Codes<br>1. Illegal mode code or invalid sub-address from internal or external legality block | <br>MSGERR in SW is set, and SW is transmitted. Message Failure interrupt generated |
| Broadcast Data Commands<br>1. TX bit set in command word | <br>Data transfer is aborted. MSGERR in SW is set, and SW is not transmitted. Message Failure interrupt generated |
| Data Word<br>1. Parity or Manchester encoding errors<br>2. Incorrect number of words received<br>3. Data words are continuous<br>4. Incorrect SYNC waveform | <br>Data transfer is aborted<br>MSGERR in SW is set, and SW is not transmitted<br>Message Failure interrupt generated |
| Transmit Data Error<br>1. The RT monitors its transmissions on the bus through its decoder and verifies that the correct data is transmitted with no Manchester or parity errors. | <br>Data transfer is aborted. MSGERR in SW is set, and SW is not transmitted. Message Failure interrupt generated |
| Backend Failure<br>1. The RT makes sure that the backend responds to read and write cycles within the required time. | <br>Data transfer is aborted. MSGERR in SW is set, and SW is not transmitted. Message Failure interrupt generated |
| BUSY<br>1. Backend RTBUSY input is active at any point during the message. | <br>Data transfer is aborted. BUSY in SW is set, and SW is transmitted. Message Failure interrupt generated |
| Transmitter Overrun<br>1. Transmits for greater than 67 µs. The internal state machines prevent this from happening, but the core includes the required timer and functionality. This is implemented separately from the encoder to provide complete protection. | <br>Core shuts down transmissions on the bus |

## Built-In Test Support

The Core1553BRT-EBR provides a BIT word. This is used to communicate fail information back to the bus controller. The BIT word contains the information in Table 12.

*Table 12* • **BIT Word**

| Bit(s) | Function | Description |
|---|---|---|
| 15 | BUSINUSE | Indicates on which bus the transmit BIT word command was received<br>'0': Bus A '1': Bus B |
| 14 | LPBKERRB | Indicates that the loopback logic detected an error on the transmitted data for Bus B. This bit is cleared by the CLRERR input. |
| 13 | LPBKERRA | Indicates that the loopback logic detected an error on the transmitted data for Bus A. This bit is cleared by the CLRERR input. |
| 12 | SHUTDOWNB | Indicates that Bus B is shutdown. This occurs after a transmitter shutdown mode code is received or the hardware timer detected that the core transmitted for greater than 668 µs on Bus B. |
| 11 | SHUTDOWNA | Indicates that Bus A is shutdown. This occurs after a transmitter shutdown mode code is received or the hardware timer detected that the core transmitted for greater than 668 µs on Bus A. |
| 10 | TFLAGINH | Terminal flag inhibit setting |
| 9 | WCNTERR | A word count error has occurred. This bit is cleared by the CLRERR input. |
| 8 | MANERR | A Manchester encoding error has occurred. This bit is cleared by the CLRERR input. |
| 7 | PARERR | A parity error has occurred. This bit is cleared by the CLRERR input. |
| 6 | Reserved | Set to '0' |
| 5 | MEMFAIL | The backend memory interface failed to complete an access within the required time. This bit is cleared in the CLRERR input. |
| 4:0 | VERSION | Indicates the core version<br>'01000': EBR version 1.0 (pre-production)<br>'01001': EBR version 2.0 |

# Command Legalization Interface

1553EBR commands can be legalized in two ways with the Core1553BRT-EBR. For RTL versions, one of the modules in the source code can be edited to legalize or make illegal command words based on the sub-address, mode code, word count, or broadcast fields of the command word. For netlist and RTL versions, external logic may be used to decode the legal/illegal command words (see Figure 8).

The user customization logic block takes in the CMDVAL and simply sets CMDOKAY for all legal command words. The CMDVAL encoding is given in Table 13. The external logic must implement this function within 3 µs.

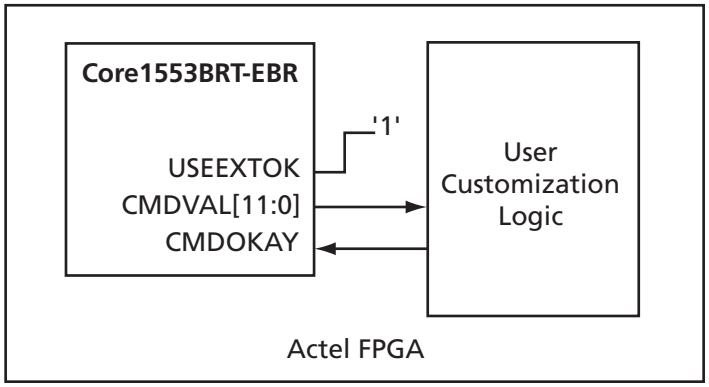



*Figure 8* • **Command Legalization Logic**

*Table 13* • **CMDVAL Encoding**

| Bit(s) | Function | Description |
|---|---|---|
| 11 | Broadcast | '1' indicates broadcast, i.e., the RT address was set to 31 in the 1553EBR command word. |
| 10 | Transmit or Receive | TX/RX field from the 1553EBR command word. '0' indicates receive and '1' transmit. |
| 9:5 | Sub-Address | Sub-address field from the 1553EBR command word |
| 4:0 | Word Count<br>Mode Code | Word count field from the 1553EBR command word. When the sub-address is 0 or 31, this contains the 1553EBR mode code. |

# Bus Transceivers

Core1553BRT-EBR drives the 1553EBR bus through standard RS485 transceivers, such as the Texas Instruments SN65HVD10. Typical connections are shown in Figure 9 on page 19.

It is recommended that the transceiver used support 3.3 V operation to allow direct connection to the 3.3 V I/Os on the FPGA.

# Typical RT Systems

The Core1553BRT-EBR can be used in systems with and without backend memory. Figure 9 on page 19 shows a typical implementation for a system with backend memory and a CPU to process the messages. Figure 10 on page 19 shows a system with direct connection between the Core1553BRT-EBR and external analog-to-digital converters, etc. In this case, any glue logic required between the core and the device being interfaced to can simply be implemented within the FPGA containing the core.

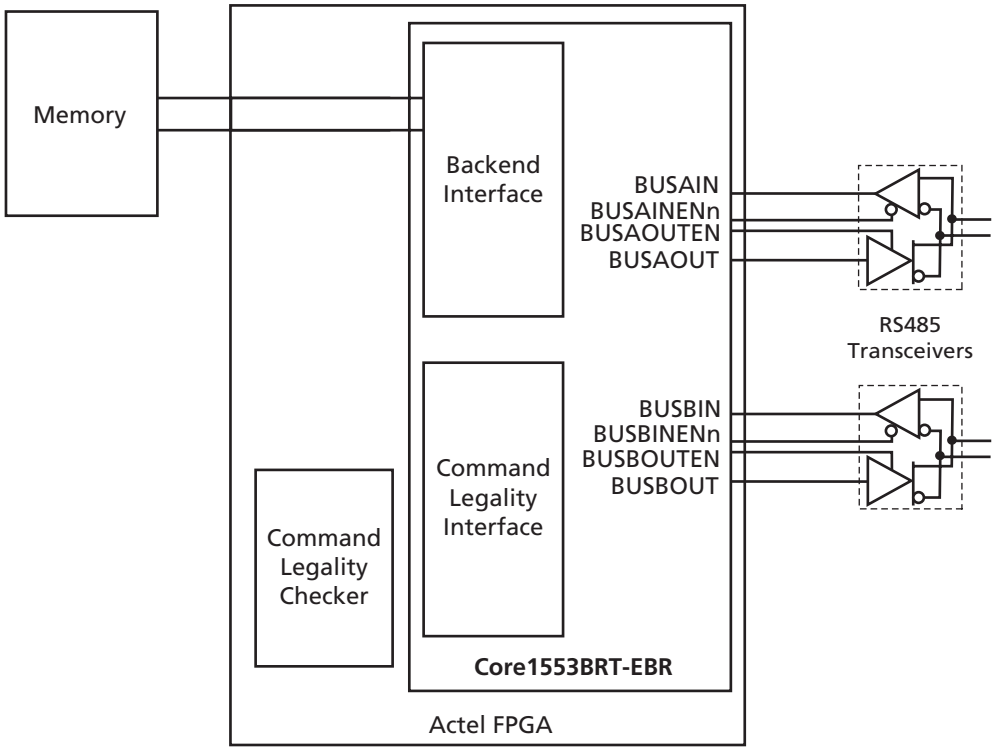



*Figure 9* • **Typical CPU and Memory-Based RT System**

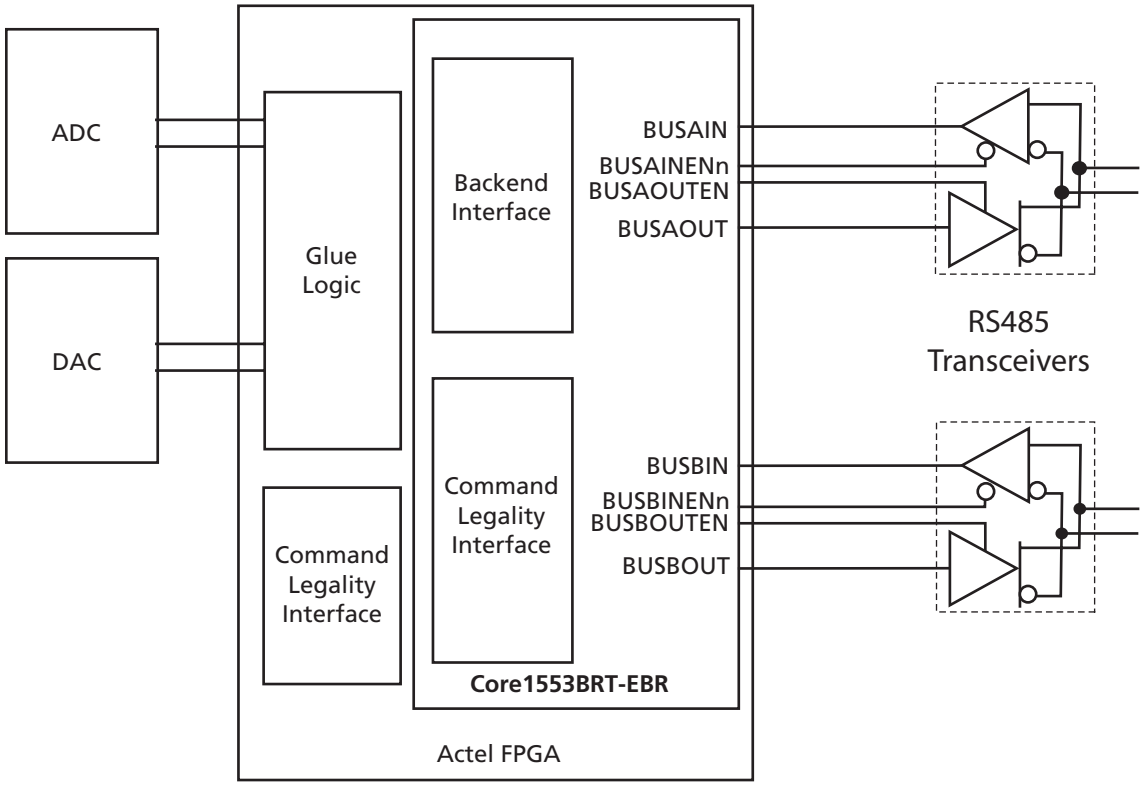



*Figure 10* • **Typical Non-Memory-Based RT System**

# Specifications

## Memory Write Timing – Asynchronous Mode

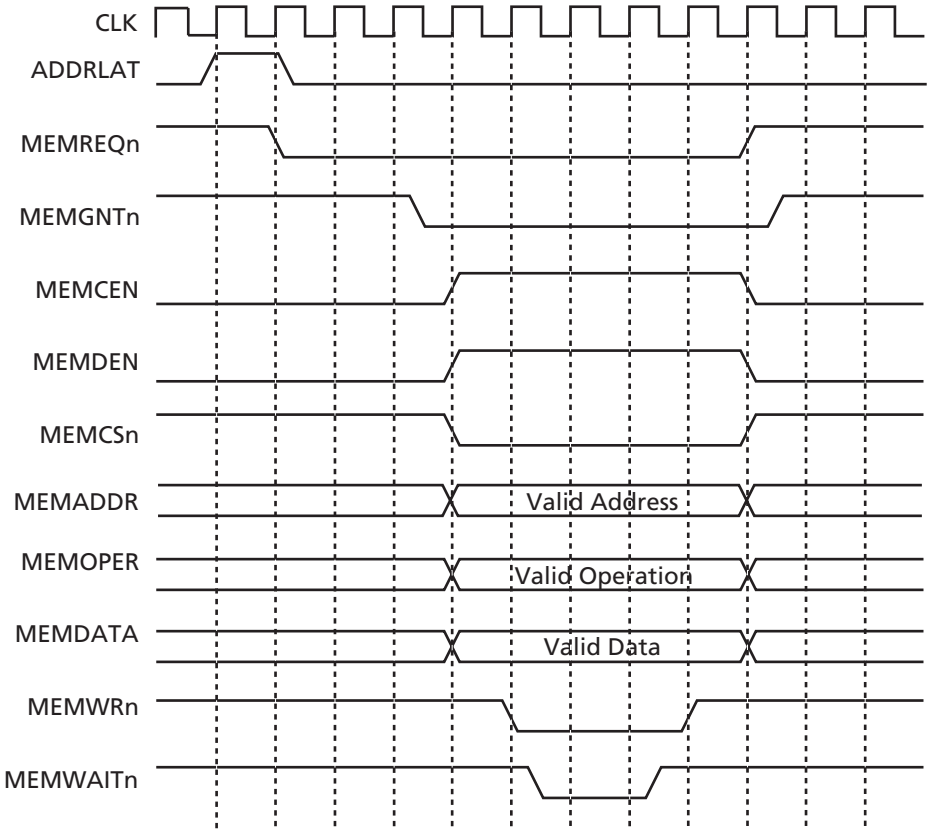



*Figure 11* • **Memory Write Timing – Asynchronous Mode**

## Memory Write Timing

*Table 14* • **Memory Write Timing**

| Sync Mode | Description | Time |
|---|---|---|
| $T_{pwWR}$ | Write pulse width (no wait states) | 1 clock cycle |
| $T_{pdGNT}$ | Maximum delay from MEMREQn to MEMGNTn active | 1.2 µs |
| $T_{suDATA}$ | Data setup time to MEMWRn low | 1 clock cycle |
| $T_{suADDR}$ | Address setup time to MEMWRn low | 1 clock cycle |
| $T_{hdDATA}$ | Data hold time from MEMWRn high | 1 clock cycle |
| $T_{hdADDR}$ | Address hold time from MEMWRn high | 1 clock cycle |
| $T_{suWAIT}$ | Wait setup to rising clock edge | 1 clock cycle |

## Memory Read Timing – Asynchronous Mode

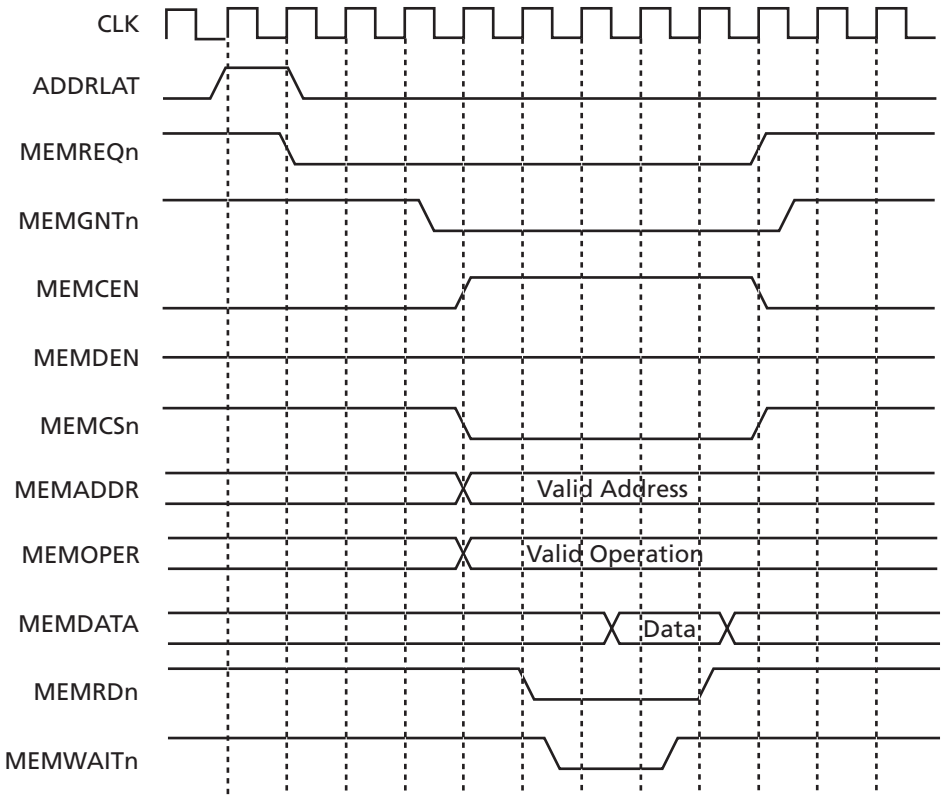



*Figure 12* • **Memory Read Timing – Asynchronous Mode**

## Memory Read Timing

*Table 15* • **Memory Read Timing**

| Async Mode | Description | Time |
|---|---|---|
| $T_{pwRD}$ | Read pulse width (no wait states) | 1 clock cycle |
| $T_{pdGNT}$ | Maximum delay from MEMREQn to MEMGNTn active | 1.2 µs |
| $T_{suADDR}$ | Address setup time to MEMRDn low | 1 clock cycle |
| $T_{hdADDR}$ | Address hold time from MEMRDn high | 1 clock cycle |
| $T_{suWAIT}$ | Wait setup to rising clock edge | |
| $T_{suDATA}$ | Data setup time to MEMRDn high | |

## Memory Write Timing – Synchronous Mode

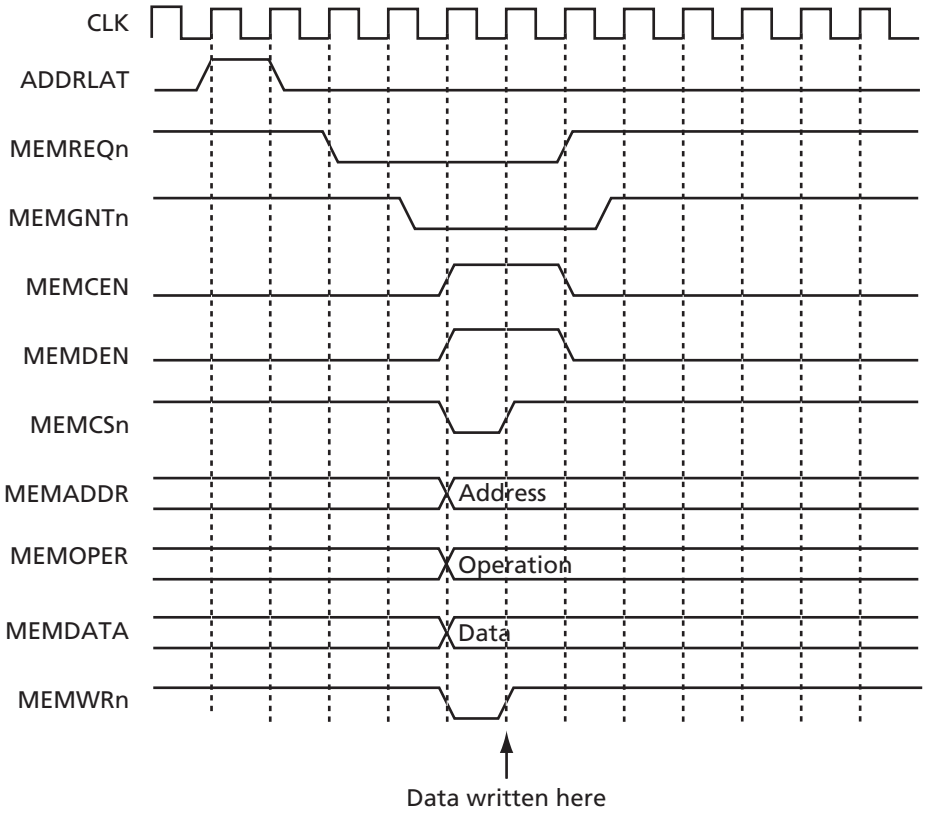



*Figure 13* • **Memory Write Timing – Synchronous Mode**

## Memory Read Timing – Synchronous Mode

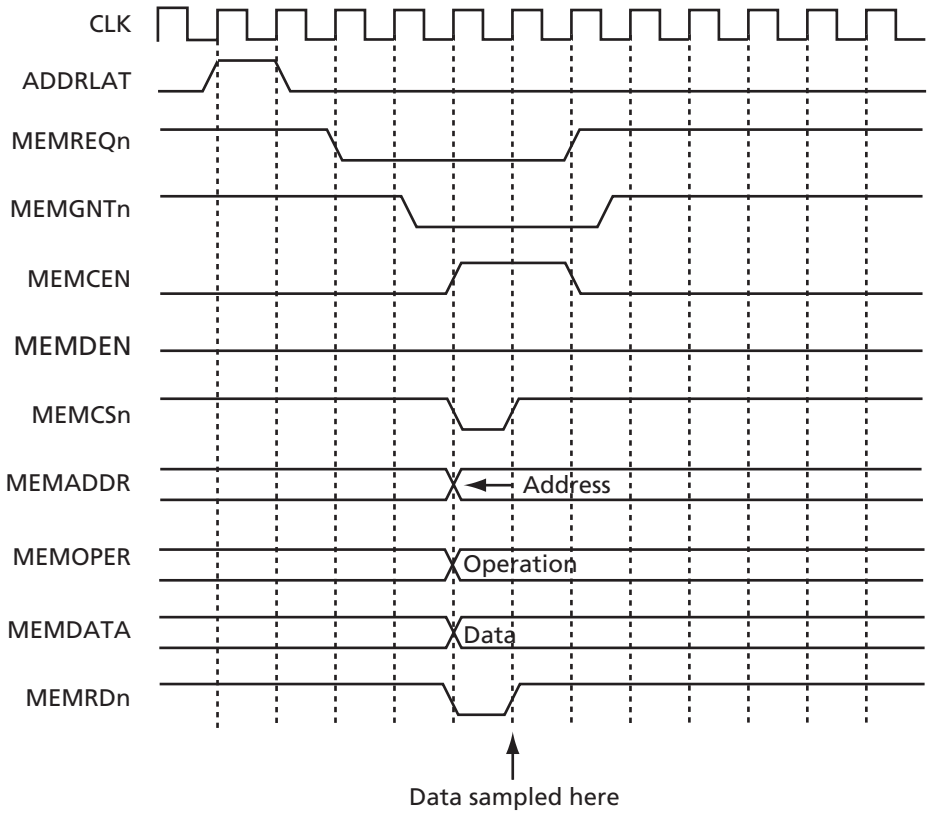



*Figure 14* • **Memory Read Timing – Synchronous Mode**

## Command Word Legality Interface Timing

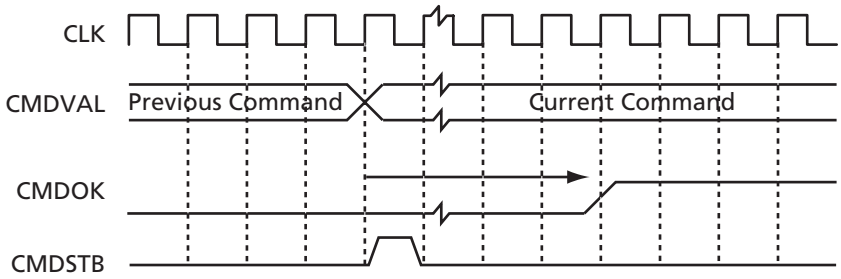
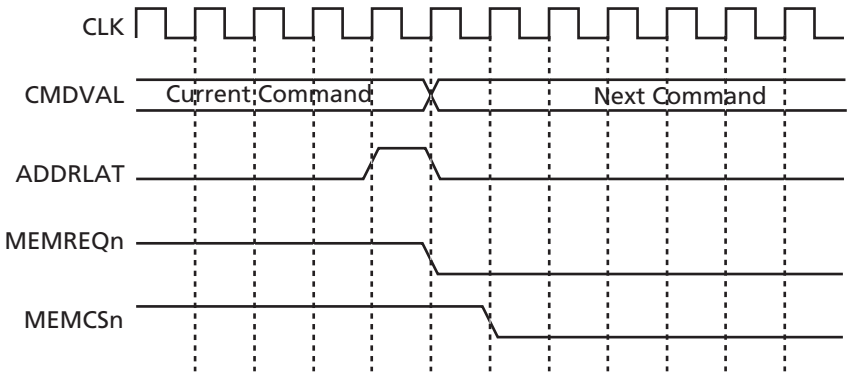



*Figure 15* • **Command Word Legality Interface Timing**

*Table 16* • **Command Word Legality Interface Timing**

| Name | Description | Time |
|---|---|---|
| $T_{pdCMDOK}$ | Maximum external command word legality decode delay | 3 µs |

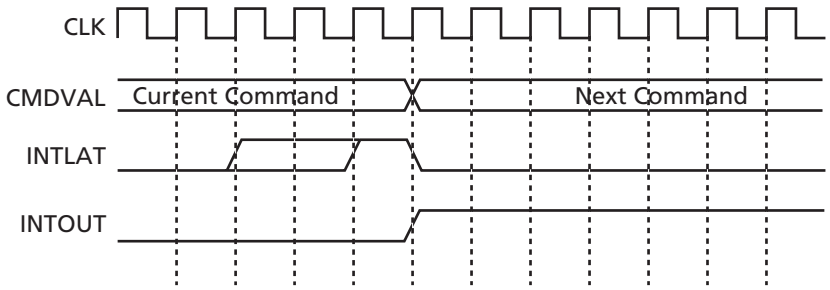
## Address Mapper Timing




**Note:** This figure shows worst-case timing when a second 1553EBR command arrives as the core starts a backend transfer and MEMGNTn is held low.

*Figure 16* • **Address Mapper Timing**

## Interrupt Vector Extender Timing




**Note:** This figure shows worst-case timing when a second 1553EBR command arrives as the core asserts an interrupt request. Also, INTLAT may be active for several clock cycles prior to INTOUT.

*Figure 17* • **Interrupt Vector Extender Timing**

## RT Response Times

RT response time is measured from the midpoint of the parity bit in the command word to the midpoint of the status word sync (Table 17).

*Table 17* • **RT Response Times**

| Spec | Description | Time |
|---|---|---|
| $T_{rtresp}$ | RT response time | 0.4 to 0.8 µs |
| $T_{xxto}$ | Transmitter timeout | 71 µs |

The RT-to-RT timeout is measured from the first command word parity bit to the expected sync of the first data word.

# Transceiver Loopback Delays

Core1553BRT-EBR verifies that all transmitted data words are correctly transmitted. As data is transmitted by the transceiver on the 1553EBR bus, it is monitored by the transceiver and decoded by Core1553BRT-EBR. The core requires that the loopback delay, i.e., the time from BUSAOUT to BUSAIN, be less than 180 ns.

The loopback delay is a function of the internal FPGA delay, PCB routing delays, internal transceiver delay, and transmission effects from the 1553EBR bus. Additional register stages may be inserted on either the 1553EBR data input or output within the FPGA, providing the required loopback delay is not violated.

## Clock Requirements

To meet 1553EBR transmission bit rate requirements, the Core1553BRT-EBR clock input must be 100 MHz ±0.01%.

# Ordering Information

Core1553BRT-EBR can be ordered through your local Actel sales representative. It should be ordered using the following number scheme: Core1553BRT-EBR-XX, where XX is listed in Table 18.

*Table 18* • **Ordering Codes**

| XX | Description |
|---|---|
| EV | Evaluation version |
| SN | Netlist for single-use on Actel devices |
| AN | Netlist for unlimited use on Actel devices |
| SR | RTL for single-use on Actel devices |
| AR | RTL for unlimited use on Actel devices |
| UR | RTL for unlimited use and not restricted to Actel devices |

# List of Changes

The following table lists critical changes that were made in the current version of the document.

| Previous Version | Changes in Current Version (Advanced v1.1) | Page |
|---|---|---|
| Advanced v1.0 | The product name was changed from Core1553EBRRT to Core1553BRT-EBR. | N/A |
| | Changed "MIL-STD-1553EBR" to "MIL-STD-1553B" under "Verification and Compliance" | 1 |
| | First bullet added under "Verification and Compliance" | 1 |
| | Changed "SAE AIR5610" to "SAE AS5682" under "RT-to-RT Transfer Support" | 14 |
| | Changed Time values in Table 17 | 24 |
| | Changed maximum loopback delay under "Transceiver Loopback Delays" | 25 |

# Datasheet Categories

In order to provide the latest information to designers, some datasheets are published before data has been fully characterized. Datasheets are designated as "Product Brief," "Advanced," and "Production." The definition of these categories are as follows:

## Product Brief

The product brief is a summarized version of an advanced or production datasheet containing general product information. This brief summarizes specific device and family information for unreleased products.

## Advanced

This datasheet version contains initial estimated information based on simulation, other products, devices, or speed grades. This information can be used as estimates, but not for production.

## Unmarked (production)

This datasheet version contains information that is considered to be final.

www.actel.com

**Actel Corporation**

2061 Stierlin Court
Mountain View, CA
94043-4655 USA
**Phone** 650.318.4200
**Fax** 650.318.4600

**Actel Europe Ltd.**

Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
**Phone** +44 (0) 1276 401 450
**Fax** +44 (0) 1276 401 490

**Actel Japan**
www.jp.actel.com

EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
**Phone** +81.03.3445.7671
**Fax** +81.03.3445.7668

**Actel Hong Kong**
www.actel.com.cn

Suite 2114, Two Pacific Place
88 Queensway, Admiralty
Hong Kong
**Phone** +852 2185 6460
**Fax** +852 2185 6488

51700061-1/2.06