CONF-891087--13 1--13

DE90 004171

THE SINGLE-CHIP FASTBUS SLAVE INTERFACE*

R O. Nelson, D. R. Machen, and R W. Downing
Scientific Systems International, Ltd. Joint Venture**

Summary

A single-chip implementation of the general-purpose FASTBUS Slave Interface (FSI) has been developed in ECL gate-array technology. The FSI will occupy only 1.6% of the available circuit board space while providing a complete 32-bit interface to the FASTBUS. All mandatory slave-interface requirements of IEEE 960[1] are supported, in addition to several non-mandatory requirements and the optional, extended MS code features.

Geographic, logical, and broadcast addressing are implemented using on-chip registers. An optional multiple-module addressing technique is included that allows participating modules residing on a common crate or cable segment to respond as if individually addressed in sequence.

The user interface provided by the FSI allows control of slave status-response and connection timing for both address and data cycles. The BIT1 ECL array technology[2] used for the FSI allows direct connections to the FASTBUS, eliminating the need for external driver/receiver buffers.

Design Methodology

We felt it vital to the development of the FSI that its logical structure be first detailed in a behavioral modeling language. We partitioned the FSI into several major functional blocks, as shown in Figure 1, to make the design process more tractable. The advantage to the behavioral-modeling approach is that the design can be simulated, modified, and further developed more easily at the functional-block level than can a hardware design. Changes to a design under development are made with more ease in software than in hardware schematics.

The logical operation of the FSI was then verified using our behavioral model prior to conversion to gate-array elements. The Daisy/Cadnetix Systems Corporation DABL modeling language[3] was used for this stage of the design process. We accomplished schematic capture and simulation of the FSI with the aid of our Daisy/Cadnetix CAE equipment. As the conversion progressed, we were able to reuse the simulation test vectors that had been first developed with the behavioral model; thus testing began at a very early stage, as it must with high-density array technology. Functional DABL blocks were then replaced in sequence by their hardware equivalents.

FSI Major Features

We were able to place a complete 32-bit, ECL FASTBUS slave interface into a single gate-array package. The small physical size of the package allows close placement of the chip to the segment connector, resulting in short metal runs from connector to FSI. The low capacitive loading at the array pins and the minimum capacitive loading from the pc trace allows the FSI to be connected directly to the FASTBUS, eliminating the need for ECL receivers and drivers common on FASTBUS modules in the past. Control signals are provided to allow use of buffer tranceivers if desired, however.

We have placed the critical CSR registers required for slave interface into the FSI. The NTA register and CSR#3 are 32-bit registers, as is CSR#1. The latter is a user-defined register that supports a multi-module addressing feature more fully described in Ref. 4. The NTA register is output to the user application on a gated 32-bit bus; gate control is provided to the user as a chip input. The broadcast mode register, CSR#7, is an internal 16-bit register. Finally, we have provided, on chip, the functions defined by IEEE 960 for CSR#0. These include user-application interface-controls for bits 0-5, 14, and 15. We have also provided controls for application-specific bits, like the module ID. The use of the on-chip registers 0, 1, 3, and 7 is controlled by enable pins.

Enable pins also provide for selecting either SS=1 or WT=1 as the module busy response, and for the selection of the width of the internal logical address field.

The FSI can, if enabled for "advanced-response" mode, respond to additional MS codes. The operation of the FSI during "advanced-response" data cycles is defined for each transition of DS over the eight possible MS codes, as shown in Table 1. The advanced response MS codes are fully described in Ref. 4 and elsewhere[5].

Operating modes

The FSI will respond to geographic addressing, logical addressing, and broadcast addressing (cases 1-6, 8). A 5-bit-wide logical width input is used to set the width of the internal address field in a logical address. Address-cycle response by the FSI is controlled by the user application through handshake and status signals. Multi-module interleave addressing is an optional feature of the FSI that uses both logical addressing and the contents of CSR#1 across a suite of identical slave modules.

The FSI implements the standard data cycle modes (random, secondary, block, and pipe) for the defined MS codes and for the advanced MS codes[4]. Data-cycle response by the

* This development was supported by the U. S. Department of Energy under SBIR contract No. DE-AC02-87ER80454.

** 3491-B Trinity Drive, Los Alamos, NM 87544.

Paper to be presented at the IEEE 1989 Nuclear Science Symposium, San Francisco, CA
Oct. 25-27, 1989

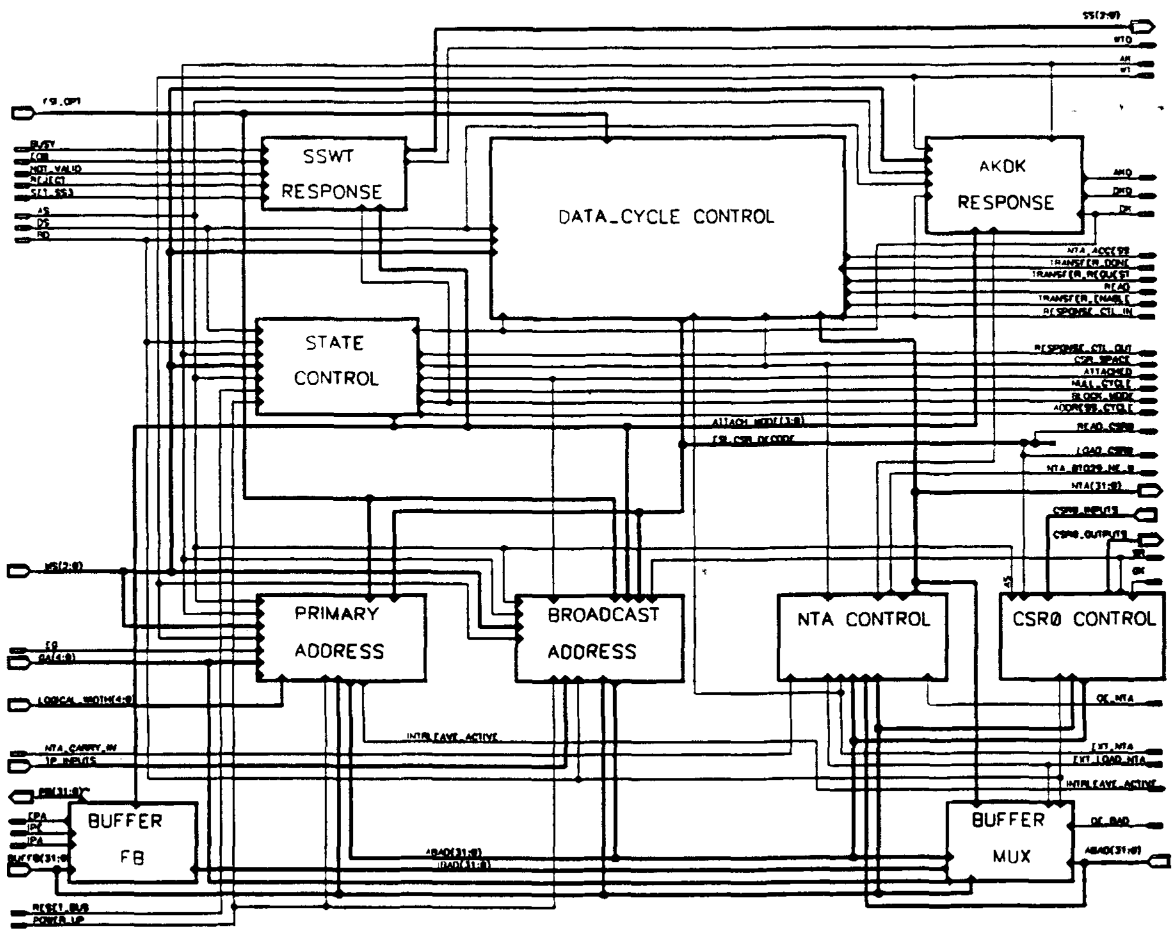

Figure 1 FSI Block Digram (showing only external connections)

FSI, and SS generation, are controlled by the user application through additional handshake and status signals.

FASTBUS Interface

We have configured the FSI to operate on a FASTBUS cable segment in addition to the crate segment. As such, the FSI does not use the segment TP connection. The FSI generates its own TP connection through the use of internal select logic based upon the geographic address pins, in the case of the crate segment, or switches, in the case of the cable segment.

For either segment, an enable signal is provided to operate transceivers between the FSI and the FASTBUS segment connector, if desired.

We placed a protective buffer in the FSI data path to implement the advanced mode requirements. The user can take advantage of this data buffer (as a pipeline register), even though advanced modes may not be used, to overlap internal operations with operations on the FASTBUS. The result can be increased data transfer rates in read operations.

User Interface

The FSI user interface provides a set of 53 output signals, 32 input signals, and 32 bidirectional signals to the user application for control of FSI response and data transfer across the interface. The output signals include 32 NTA register signals and the bidirectional signals are the 32-bit BAD (*i.e.* Buffered AD) bus.

Timing and response control is handled by the Transfer-Request, Transfer-Done, and Transfer-Enable handshake signals and the Response-Control-Out and the Response-Control-In handshake signals. Signals indicating either read or write to CSR or data space are provided. The NTA register in the FSI is available to the user application and is loadable for special tasks. The NTA register is incremented during block or pipe mode transactions. Load and Read of CSR#0 signals are output for use with external CSR#0 features.

Slave status from the user application is controlled by the user application through use of 5 interface signals. Two of these signals are used both in primary address cycles and data cycles, while the remaining three are used only in data cycle operations.

MS Code	DS(u)	DS(d)
0	Single word	Cleanup
1	Block	Block
2	Load NTA(P/A) for RD = 0 Read NTA for RD = 1	Cleanup
3	Pipeline	Pipeline
4	Cleanup	Single word
5	Protective buffer	Protective buffer
6	Cleanup	Load NTA(P/A) for RD = 0 Read NTA for RD = 1
7	Load NTA(P/A~) for RD = 0 Read CSR#0 for RD = 1	Load NTA(P/A~) for RD = 0 Read CSR#0 for RD = 1

Table 1. Advanced MS codes for data cycles in the FSI. References to NTA(P/A) denote space switch to that selected at primary address time coupled with load of the NTA register. Reference to NTA(P/A~) is similar except that space selected is opposite to that selected at primary address time. If the advanced MS codes are disabled, then MS codes 4-7 are reserved and return SS = 6 to the master.

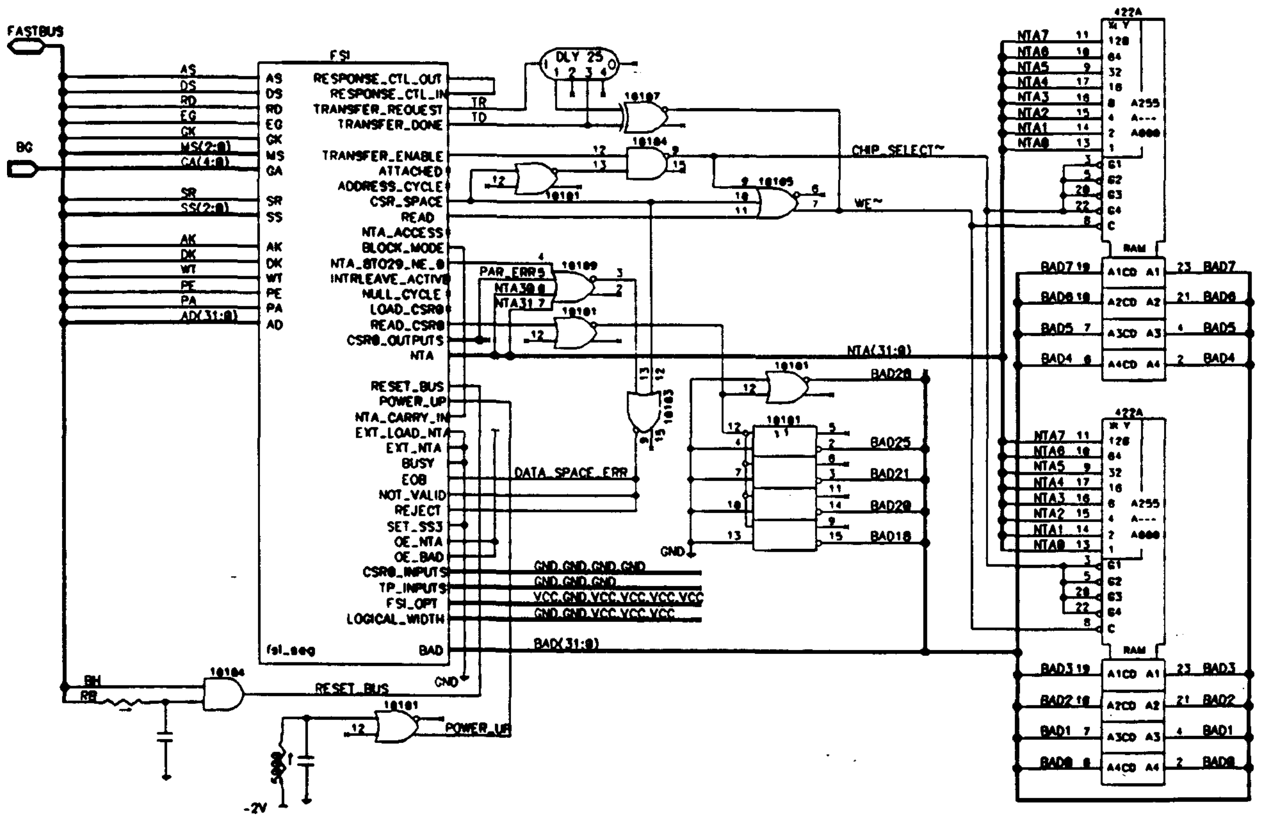

Figure. 2 FSI Test Case Slave Module

Expected Performance

We have assembled a test-case slave module, shown in Figure 2, to illustrate some of the connections between the FSI and typical elements of a FASTBUS slave and to measure typical performance predicted by computer simulation. As illustrated, the module contains a small, ECL static random-access memory (RAM) able to use both random and block mode data cycles in transferring data to and from the ECL-10422 RAMs. The few additional components required over the basic RAMs are commentary on the use of gate-array technology for packaging interface logic. An important result of such a test case is the maximum data transfer rate into and out of the module, which is an indication of the basic performance of the interface logic. For fully handshaked data cycles (using the fastest master allowed by IEEE 960), a sustained rate of 25 MHz is obtained for a slave application that completes data accesses in 15 ns. These figures, shown in Table 2, illustrate the other significant advantage in gate arrays:..that of low overall propagation delay and high internal logic speeds.

Implementation

We have selected the BIT1, 2 micron, ECL gate array technology[2], used by Raytheon Corporation and BIT Inc., for the FSI. We selected the Raytheon CGA70E18 array containing 12540 equivalent gate sites, with an typical gate dissipating only 300 microwatts and operating with a 300 ps typical propagation delay. We expect the completed FSI to dissipate less than 7 watts and use 1.6% of the available space on a FASTBUS printed circuit board. The array interface is 10KH ECL compatible.

The array package is a pin-grid array with 232 pins on .100 in. centers, measuring 1.8 in. x 1.8 in. Although the package requires a heat sink for proper cooling, the height of the array/heat sink combination is within that allowed by IEEE 960[1].

TRANSITION	DELAY
EG -> AK (geographic address)	14 ns
AS -> AK (logical address)	16 ns
AS -> NTA	10 ns
EG -> Response_Control_Out	8 ns
Response_Control_In -> AK	6 ns
BAD -> PA (read)	7 ns
BAD -> AD (read)	4.5 ns
AD -> BAD (write)	2.5 ns
DS -> DK	14 ns
DS -> NTA (write)	9 ns
TRANSFER_DONE -> DK(t)	10 ns
TRANSFER_DONE -> NTA+1	9 ns (block/pipe)
DS(t) -> TRANSFER_ENABLE	10 ns (csr space)
DS(t) -> TRANSFER_ENABLE	6 ns (data space)
EG -> CSR_SPACE	8 ns
DS(t) -> CSR_SPACE	12 ns
TRANSFER_DONE -> SS	6 ns

Table 2 FSI Simulation Performance Results

Conclusions

We have placed a complete 32-bit FASTBUS slave interface in a single-chip gate array and reduced the circuit board space required for a typical interface by nearly 95%. For slaves with 15 ns response timing, the maximum data rate expected with the single-chip interface is 25 MHz for all types of data cycles. For real masters, a data transfer rate as high as 25 MHz is usually obtained only for block mode transfers.

At this writing, the final testing of the array design is near completion, allowing fabrication of the prototype FSI chips to proceed. The first application for the FSI will be an 8 megaword x 32 bit interleaved bulk memory module for FASTBUS.

References

1. FASTBUS-Modular High Speed Data Acquisition and Control Systems for High Energy Physics and Other Applications, ANSI/IEEE Standard 960-1986.

2. BIT1 ECL array technology, Bipolar Integrated Technology, Inc., Beaverton, Oregon 97006. Raytheon Corp. uses the BIT1 process for their ECL gate arrays.

3. Daisy Behavioral Language (DABL), Publication No. 600-315-01, Daisy/Cadnetix Systems Corporation, Sunnyvale, California.

4 SSI/JV Technical Note #2, October 1989, The FASTBUS Slave Interface Users Manual. Scientific Systems International, Ltd. Joint Venture.

5. Skegg, R, private communications-Description of the PCL Gate Array for FASTBUS, Rev. 1.5, Aug. 1988.