SLAC-PUB-2972
September 1982
(I)

An I/O Register to FASTBUS Interface *

C. A. Logg and L. Paffrath
Electronics Department
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

Summary

An input/output register to FASTBUS interface (IORFI) has been designed which provides an inexpensive and simple means to connect a computer to a FASTBUS backplane segment. The FASTBUS backplane interface is built on a single width FASTBUS module. It is connected to a computer by two 16-bit parallel input registers and two 16-bit parallel output registers, which makes the interface computer-non-specific. This paper describes the operational characteristics of this interface, its advantages, limitations, and briefly, the uses to which it has been put.

Introduction

As the development of the FASTBUS specification[1] neared completion, it became obvious that an easily adaptable computer to FASTBUS connection was necessary to facilitate protocol verification as well as initial prototype module testing. In mid 1980 the first IORFI was designed and built. Over the next 2 years, through June 1982, as the FASTBUS specification evolved and the protocol changed, the IORFI design was modified to conform with the latest specifications. Described herein is the IORFI which is currently under development and which will comply with the FASTBUS Specification which is to be published in the Fall 1982.

FASTBUS to Computer Connection

The IORFI is connected to a processor via two 16-bit parallel output registers (OR1,OR2) and two 16-bit parallel input registers (IR1,IR2). One of the output registers (OR1) is used as a control register to specify the interface function that is to be performed when the interface is accessed via the Data-in Register (IR2) or the Data-out Register (OR2). The other input register (IR1) is used to read the direct status of some of the FASTBUS lines independent of OR1.

Control Register Functions

The control register can be loaded with one of eleven function codes to specify the function the FASTBUS interface is to perform when the interface is accessed via the data-in or data-out register. Functions 0-5 are used for reading from and writing to FASTBUS and controlling the IORFI´s WT line control options. Functions 8-12 are for reading back the internal registers of the interface. During the generation of the actual signals on the FASTBUS segment, all timing signals (AS,AK,DS,DK) are delayed with respect to any control lines which are concurrently generated. If WT is asserted, it will inhibit the next timing transition generated by this device. If RB is asserted externally, the IORFI will remove all signals from the bus.

Note the following notational conventions: each FASTBUS signal is labeled with a 2 character mnemonic (AS,AK,PE,MO, etc.). In several of the tables below, a third character is appended to some of the two character mnemonics. The significance is as follows:

character	meaning
+	setting the bit asserts the signal on the line
-	setting the bit clears the assertion of the signal
!	setting the bit enables the feature
\	setting the bit disables the feature
B	the readback is the actual bus signal
I	the readback is the internal interface signal
?	enabled state (=1 if enabled, =0 if not enabled)
W	WT has been generated due to indicated transition
*	enabled option has been triggered

Lines which do not have a third character associated with them assert the signal on the bus according to the state of the bit (if bit=0 then the signal is not asserted).

Function 0

Read (via the data-in register) or write (via the data-out register) the FASTBUS protocol lines.

Data-out Register Bit Assignments

15	14	13	12	11	10	9	8	(high byte)
DK	AK	PA	PE	SR	S2	S1	SO	

7	6	5	4	3	2	1	0	(low byte)
DS	AS	AR	GK-	RD	M2	M1	MO	

Data-in Register Bit Assignments

15	14	13	12	11	10	9	8	(high byte)
DKB	AKB	PAB	PEB	SRB	S2B	S1B	SOB	

7	6	5	4	3	2	1	0	(low byte)
DSB	ASB	ARB	GKI	RDB	M2B	M1B	MOB	

When GK- is written, the act of setting of this bit causes the GK assertion by this device to be turned off. Note that the readback is the internal interface GK state. The state of GK on the bus is available via the Direct Status Register (see below). If this interface is the current master, then the two may be the same.

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.

(Presented at the Nuclear Science Symposium, Washington, D.C., October 20-22, 1982.)

Function 1

Read (via the data-in register) or write (via the data-out register) the FASTBUS AD<16:31> lines.

Function 2

Read (via the data-in register) or write (via the data-out register) the FASTBUS AD<0:15> lines.

For functions 1 and 2, if the IORFI is acting as a master on the bus, the AD lines are gated out only if GK internal or AS is set. If the IORFI is acting as a slave on the bus, the AD lines are gated out only if RD=1 on the bus and AK=1.

Function 3

Read (via the data-in register) or write (via the data-out register) the AL vector:

15	14	13	12	11	10	9	8	(high byte)
0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0	(low byte)
0	0	AL05	AL04	AL03	AL02	AL01	AL00	

The IORFI´s arbitration level is set via this function. The readback (which is the bus AL value) is valid only after the 0->1 transition of AG. The value of AL<00:05> will be the same as the IORFI´s if the IORFI´s arbitration level is the highest, or if the IORFI was the last master and no new arbitration cycle has taken place.

Function 4

This function is used to set and clear various FASTBUS lines. Most of these lines normally are not manipulated by an interface. However, the IORFI allows manipulation here for diagnostic purposes.

Data-out Register Bit Assignments

15	14	13	12	11	10	9	8	(high byte)
| |->SEIZE BUS RELEASE	AI\	TP-	AG-	BH-	AI-	RB-	EG-	

7	6	5	4	3	2	1	0	(low byte)
| |->SEIZE BUS	AI!	TP+	AG+	BH+	AI+	RB+	EG+	

The IORFI has an option (the SEIZE BUS option) to hold mastership of the segment regardless of the fact that other options may indicate that mastership should be released. The act of setting the SEIZE BUS bit results in GK being asserted and maintained on the segment. The act of setting the SEIZE BUS RELEASE bit results in the disabling of the SEIZE BUS option.

Data-in Register Bit Assignments

15	14	13	12	11	10	9	8	(high byte)
0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0	(low byte)
GKB	AI?	TPB	AGB	BHB	AIB	RBB	EGB	

Function 5

The IORFI has various timing transition triggered WT line generation capabilities which are monitored and controlled via this function. Writing the bits in this function enables or disables the various facilities. The function readback provides information about the WT state if it has been generated by the IORFI.

Data-out Register Bit Assignments

15	14	13	12	11	10	9	8	(high byte)
0	0	GA\	0	0	TT\	AS\	WT-	

7	6	5	4	3	2	1	0	(low byte)
0	0	GA!	0	0	TT!	AS!	WT+	

Data-in Register Bit Assignments

15	14	13	12	11	10	9	8	(high byte)
DKW	AKW	GA*	0	0	TT*	AS*	WTW	

7	6	5	4	3	2	1	0	(low byte)
DSW	ASW	GA?	0	0	TT?	AS?	WTW	

Note the following:

WT+: The act of setting this bit will set the WT line. The readback indicates that WT is set via this option.

WT-: The act of setting this bit will clear the WT line if it was set by setting WT+.

TT! and TT\: These bits operate together to control the WT line assertion on AS, AK, DS, and DK transitions. The act of setting TT! enables the timing transition WT generation or clears the WT (if it was set as a result of a previous transition), and rearms it. The act of setting bit TT\ clears the WT (if it was set as a result of a previous transition) and disables the timing transition WT generation. If both TT! and TT\ are set simultaneously, then nothing happens. The readback TT? is the enabled state of this feature. The readback TT* is 1 if WT has been generated due to this option.

AS! and AS\: These bits operate together to control the WT line assertion on AS(u). The act of setting AS! enables the interface feature which causes WT to be asserted on the AS(u) transition. This allows the interface to simulate the logical addressing response of a slave. A WT line which has been set by this option, is cleared and rearmed by setting AS!. Setting AS\ will result in clearing and disabling the AS(u) WT generation. The readback AS* is 1 if WT has been generated due to this option.

GA!: The IORFI has an automatic geographic address slave response capability whereby it can generate AK when it is geographically addressed. The act of setting GA! will enable this feature. When the IORFI is geographically addressed it will generate AK and WT, and enable the timing transition WT generation feature controlled by the TT! and TT\ bits. The readback GA? is the enabled state of this option. The readback GA* is 1 if WT has been generated by this option. The act of setting GA\ will disable this feature.

DKW, AKW, DSW, and ASW: These readback bits indicate which signal transition has resulted in WT being generated via the TT! and AS! options.

Any programs handling WT must check to see whether it is being generated internally by this device, or externally by another FASTBUS device.

Function 8

Read (via the data-in register) the internal interface protocol line register. Line assignments are:

15	14	13	12	11	10	9	8	(high byte)
DKI	AKI	PAI	PEI	SRI	S2I	S1I	SOI	

7	6	5	4	3	2	1	0	(low byte)
DSI	ASI	ARI	0	RDI	M2I	M1I	MOI	

Function 9

Read (via the data-in register) the internal interface AD<16:31> data line register.

Function 10

Read (via the data-in register) the internal interface AD<0:15> data line register.

Function 11

Read (via the data-in register) the internal interface AL<0:5>.

15	14	13	12	11	10	9	8	(high byte)
0	0	0	0	0	0	0	0	

7	6	5	4	3	2	1	0	(low byte)
0	0	AL05	AL04	AL03	AL02	AL01	AL00	

Function 12

Read (via the data-in register) the internal interface register which contains the following lines:

15	14	13	12	11	10	9	8	(high byte)
0	0	0	0	0	0	0	0	

```
 7   6   5   4   3   2   1   0   (low byte)
 |  AI? TPI AGI BHI AII RBI EGI
 |-> SEIZE BUS?
```

The Direct Status Register

The Direct Status Register provides direct read access to most of the FASTBUS protocol lines. The bit assignments are:

15	14	13	12	11	10	9	8	(high byte)
DKB	AKB	WTI	WTB	SRB	S2B	S1B	SOB	

7	6	5	4	3	2	1	0	(low byte)
DSB	ASB	GA*	GKB	RDB	RBB	M1B	MOB	

Note that if GA* is set, it indicates that this device is being geographically addressed and has generated AK. The 0->1 transition of GA enables the generation of WT on any subsequent transition of AS,AK,DS, or DK (that is, it automatically enables the WT generation logic controlled by BIT 2 of function 5).

The IORFI Powerup State

On powerup, the IORFI will not assert any signals on the FASTBUS segment. Furthermore all options (GA!, TT!, and AS!) will be disabled. Note that the IORFI´s internal registers may not be cleared and must be initialized by software.

Advantages

An IORFI can be used to connect any computer which has parallel I/O ports (with associated strobes) to a FASTBUS backplane segment. The IORFI incorporates a universal connection scheme which permits standard mass terminated connectors of any width to be used. The various user pins are jumpered to the proper inputs/outputs on the board.

The IORFI´s features facilitate the development of powerful software tools. The FASTBUS line manipulation capabilities allow software to be written which can slow down FASTBUS operations, and even single step them, thus allowing the operations to be examined at human observation speeds. The WT line generation capabilities facilitate the development of limited SNOOP[2] type software whereby a series of FASTBUS operations can be recorded.

The programmable control of the FASTBUS lines also permits the development of software which can generate protocol errors and check out the response of modules to many error conditions. In addition, the IORFI could be used (with the appropriate software) to generate system errors or trouble conditions (such as a hung segment) and check out the overall system error detection and recovery schemes of a FASTBUS system. Also with the appropriate software, an IORFI can be used to emulate any FASTBUS device, including segment interconnects and FIFO devices.

Disadvantages

The main disadvantage of the IORFI stems from the fact that the speed of its FASTBUS operations is limited by the speed of the connected computer and its software drivers. For example, an IORFI cannot be used to check out module operation characteristics at full FASTBUS speeds. However, it has been used to load a sequencer with a program which then operated at high FASTBUS speeds and performed the desired high speed tests.

Actual Applications

Over the past two years, IORFIs have been used extensively in Canada, Europe, and the United States for the testing and checkout of basic FASTBUS equipment items such as backplanes, Arbitration Timing Controllers, and EG Generators. They have also been used as both master and slave devices in the prototype development and checkout of the segment interconnects, FASTBUS modules for various experiments, and other FASTBUS interfaces such as the UNIBUS Processor Interface[3] and the CERN General Purpose Interface.[4]

Software Drivers

Several different software drivers have been written for the IORFIs.

CERN has implemented a set of FORTRAN routines[5] based on the Standard Subroutines for FASTBUS[6] and a computer connection to the IORFIs via CAMAC. This package utilizes the NIM/ESONE Standard CAMAC Subroutines and is currently running on the NORSK-DATA computers under BASIC/FORTRAN and on PDP-11s under CATY/FORTRAN.

A FASTBUS Diagnostic Operating System (FBDOS)[7] written in FORTH[8] has been developed at SLAC. Currently it is tailored for use on LSI-11s or PDP-11s which are connected to the IORFI via two DRV11-C modules;[9] however it is structured in a layered fashion and the device driver layer could be modified to drive the IORFI via a CAMAC or other type of parallel I/O device.

A PASCAL based system, the FASTBUS Diagnostic Language,[10] has been developed at the University of Illinois for use in their FASTBUS hardware development and checkout. Some of the actual IORFI interface code is in MACRO-11 and has been also transported to FERMILAB.

TRIUMF has also developed IORFI drivers in a version of FORTH which runs under RSX-11M.

All of the above installations have also implemented various kinds of slave and master emulation code for the IORFI.

Future of the IORFI

Although the IORFI is a simple device, it is envisioned that it will continue to be used in FASTBUS module development and checkout and in maintenance and diagnostic situations.

Acknowledgments

We want to thank R. Downing and D. Gustavson for their suggestions, and R.S. Larsen for his support.

References

1. FASTBUS Modular High Speed Data Acquisition System for High Energy Physics and other Applications, Tentative Specification, U.S. NIM Committee, June 1982.

2. FASTBUS SNOOP Diagnostic Module, H. Walz and R. Downing, November 1980, SLAC-PUB-2637.

3. UNIBUS Processor Interface, FERMILAB Document FBN008.

4. The General Purpose Interface, A.W. Booth, C.F. Parkman, P.J. Ponting, E.M. Rimmer, CERN.

5. CERN Software for FASTBUS, E. M. Rimmer, DD Division, CERN, 1211 Geneve 23, Switzerland.

6. Standard Subroutines for FASTBUS, FASTBUS Software Working Group, Ruth Pordes (Editor), FNAL, P. O. Box 500, Batavia, Illinois 60510.

7. FASTBUS Diagnostic Operating System (FBDOS), August 1982, Connie Logg, SLAC, P.O. Box 4349, Stanford, California, 94305.

8. SLAC ELD LSI-11 FORTH User's Guide, August 1982, Connie Logg, SLAC, P.O. Box 4349, Stanford, California 94305.

9. Microcomputer Interfaces Handbook, Digital Equipment Corporation.

10. FASTBUS Diagnostic Language (FDL) Reference Manual for the UNIBUS Processor Interface, Dave Lesny, Seth Abraham, Steve Coffman, Keith Nater, Loomis Laboratory of Physics, University of Illinois, Urbana, Illinois 61801.