# A FASTBUS PROCESSOR INTERFACE USING A 68000 MICROPROCESSOR

Y.Arai, T.Murakami, E.Inoue, Y.Karita
KEK, National Laboratory for High Energy Physics
Oho-Machi, Tsukuba-Gun, Ibaraki 305, JAPAN,

I.Endo
Department of Physics, Hiroshima University
Higashi Senda-Machi, Hiroshima 730, JAPAN
and
F.Shimokoshi
Department of Physics, University of Tokyo
Hongo, Bunkyo-Ku, Tokyo 108, JAPAN

## Abstract

A versatile FASTBUS master utilize a 68000 microprocessor has been developed. The module implements geographical, logical and broadcast addressing and is able to handle the FASTBUS Service Request. The data transfer rate is about 800 kB/sec by using window driver. The logics and test results are described.

## Introduction

A FASTBUS master module using a MC68000 16-bit microprocessor has been developed for use in the TRISTAN $e^+$-$e^-$ colliding beam experiments. First prototypes were built on old size FASTBUS boards and reported in reference [1]. Here we report on the second version module, which has many additional functions and is built on a new size board.

Figure 1 shows a component side view of the module. The module(68K-FPI : 68000 FASTBUS Processor Interface) will play the role of a FASTBUS local master, so it is important not to interfere with the main data stream.† The module starts arbitration in every data transfer cycle and usually ends the cycle within 2 microseconds. The average data transfer rate is about 800 kB/sec. All the FASTBUS addressing types except block transfer are supported.

The 68K-FPI can work not only as a stand-alone master but also as a FASTBUS interface to another host computer. Communication lines to the host computer are either a RS-232C or a 16 bit parallel line. This makes it easy to connect any kind of computer to FASTBUS through the 68K-FPI. The 68K-FPI receives command packets from the host computer and operates FASTBUS cycles. This simplifies the software on the host computer greatly.

† We are thinking to use a VAX-FPI[2] to control the main data stream.

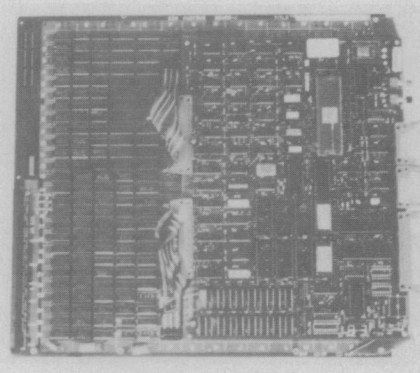


Fig.1 The 68000 FASTBUS Processor Interface(68K-FPI).

## Structural Aspect

Figure 2 shows block diagram of the 68K-FPI. The 68K-FPI consists of a 68000 computer and FASTBUS interface parts.

The interface part can be divided into 3 units; the signal line driver, the window driver and the slave logic. The signal line driver uses two 32-bit registers(FBAD and FBPL). The operation to the FASTBUS is controlled by software through those registers. Although it does not keep up with the speed of FASTBUS, it is very versatile as a debugging tool. The window driver is composed of mapping registers and sequencers. It maps MC68000 address to FASTBUS address, so the MC68000 can handle the FASTBUS modules like memory of the processor. Since the FASTBUS cycle is controlled by hard-wired logic, the speed is high enough. The slave logic has a geographical address decoder, the CSR#0(module ID and Enable bit) register and the CSR#8(arbitration level and protocol type) register.

### 68000 Computer

We are using a commercial 68000 single board computer(Maruei Shoji Co.: M68KSBD). It has a MC68000 microprocessor(12 MHz), 128 k byte dynamic RAM(access time 150 ns), 128 k byte EPROM(access time 200 ns), two serial line ports and dual 16-bit ports.

### Signal Line Driver

The 68K-FPI has two 32-bit I/O registers;

FBAD --- FASTBUS AD line drive register,
FBPL --- FASTBUS protocol line drive register.

The FBAD register corresponds to the FASTBUS AD line. Bit assignment of the FBPL register, which follows the assignment of the FASTBUS Standard Routines[3], is shown in Table 1. There are four additional bits which do not appear in the standard; Enable, Master, Slave and Reset GK bits. The Enable bit indicates the status of the CSR#0<1>(Enable to request Bus Mastership). Master bit indicates that the module is acting as a Master. Slave bit indicates that the module is acting as a Connected Slave. By setting the Reset GK bit, the flip-flop which is currently asserting the GK(Grant Acknowledge) signal is cleared.

0018-9499/84/0002-0188$01.00 © 1984 IEEE

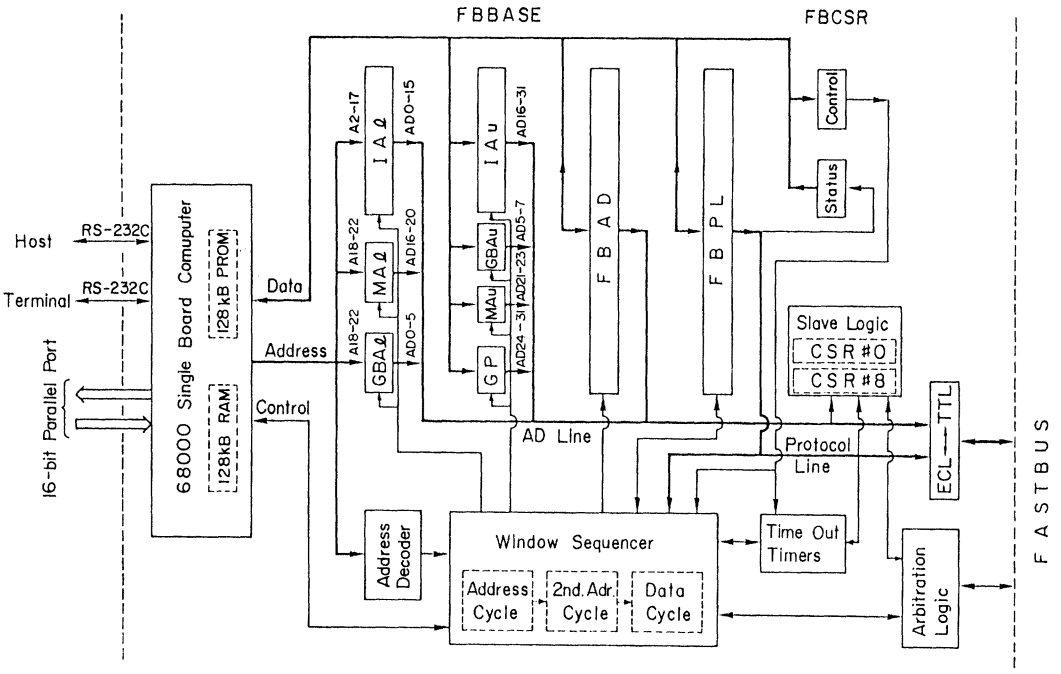



Fig.2 Block diagram of the 68K-FPI.

Table 1. Protocol Line Assignments in the FBPL register.

| Bit | I | Read | I | Write |
|---|---|---|---|---|
| | I | | I | |
| 31 | I | Enable(CSR#0<1>) | I | - |
| 30 | I | - | I | - |
| 29 | I | RB | I | RB |
| 28 | I | BH | I | BH |
| 27 | I | AG | I | AG |
| 26 | I | PE | I | PE |
| 25 | I | PA | I | PA |
| 24 | I | SR | I | SR |
| | I | | I | |
| 23 | I | AR | I | AR |
| 22 | I | AI | I | AI |
| 21 | I | AL05(Bus state) | I | - |
| 20 | I | AL04(Bus state) | I | - |
| 19 | I | AL03(Bus state) | I | - |
| 18 | I | AL02(Bus state) | I | - |
| 17 | I | AL01(Bus state) | I | - |
| 16 | I | AL00(Bus state) | I | - |
| | I | | I | |
| 15 | I | DK | I | DK |
| 14 | I | AK | I | AK |
| 13 | I | GK | I | GK |
| 12 | I | Master | I | Reset GK |
| 11 | I | WT | I | WT |
| 10 | I | SS2 | I | SS2 |
| 9 | I | SS1 | I | SS1 |
| 8 | I | SS0 | I | SS0 |
| | I | | I | |
| 7 | I | DS | I | DS |
| 6 | I | AS | I | AS |
| 5 | I | Slave | I | - |
| 4 | I | EG | I | EG |
| 3 | I | RD | I | RD |
| 2 | I | MS2 | I | MS2 |
| 1 | I | MS1 | I | MS1 |
| 0 | I | MS0 | I | MS0 |

## Window Driver

In the window operation, the FASTBUS address space is mapped to the address space of the 68000 microprocessor. Then FASTBUS modules can be treated like memory of the MC68000. The processor moves data from(to) the FASTBUS module to(from) memory or register in a single machine code. Moreover, the data transfer between FASTBUS modules( FASTBUS -> 68K-FPI -> FASTBUS ) can be done in a single machine code.

Window driver uses the FBAD register and two other registers;

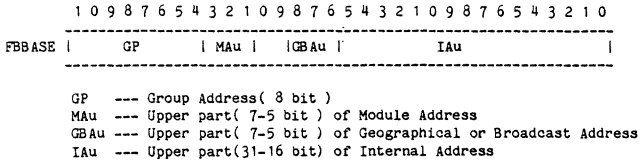
FBBASE --- FASTBUS Base address register,
FBCSR --- FASTBUS Control and Status register.

FBBASE is a 32-bit write-only register. The bit assignment of the FBBASE register is shown in Figure 3. Each field of the register is used to compose FASTBUS address. Detailed description of composing FASTBUS address is given in the following section.

Bit assignment of the FBCSR is shown in Table 2. The register has 8 write-only bits for addressing type and interrupt control. Besides, it has 8 read-only bits for interrupt flags, error code and slave status. The functions of these bits are described later.

The 68K-FPI has two types of the transfer cycle; 2 cycle transfer and 3 cycle transfer. 2 cycle transfer means the primary address cycle plus the data cycle. In the 3 cycle transfer, the secondary address cycle follows the primary address cycle. Figure 4 shows a timing diagram of the 68K-FPI for 3 cycle transfer.

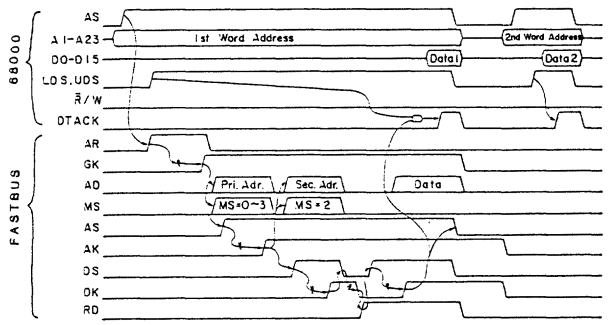
(a) 3 cycle read operation: When the window operation starts, the Arbitration Request(AR) signal is asserted at first. Then the FASTBUS cycle(the primary address, secondary address and data cycle) follows. After the latching of the FASTBUS data in the FBAD, the 68000 processor reads the 1st 16-bit data, and proceed to the 2nd word read operation.




GP --- Group Address( 8 bit )
MAu --- Upper part( 7-5 bit ) of Module Address
GBAu --- Upper part( 7-5 bit ) of Geographical or Broadcast Address
IAu --- Upper part(31-16 bit) of Internal Address

Fig.3 Bit assignment of the FBBASE register.

(a) 3 cycle read operation




(b) 3 cycle write operation

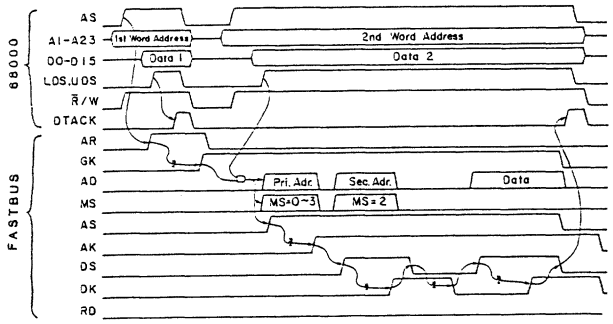



Fig.4 Schematic timing diagram of the 68K-FPI.
(a) 3 cycle read operation.
(b) 3 cycle write operation.
For the 2 cycle operation, secondary address cycle is skipped.

FASTBUS cycle finishs before the completion of the 68000 read cycle.

(b) 3 cycle write operation: The data transfer to the FBAD by the 68000 and FASTBUS arbitration cycle are done in parallel. The FASTBUS address cycle starts, after the long-word write cycle of the 68000 processor and gaining of the the bus mastership. The 68000 processor waits until the FASTBUS cycle ends.

For the 2 cycle operation, the secondary address cycle is skipped.

## Slave Logic

The 68K-FPI responds to the geographical addressing. And it has CSR#0 and CSR#8 registers. In the CSR#0 register, Enable<S01>, Disable<C01> and Device ID<16:31> bits are implemented. The CSR#8 register has bits of arbitration level<0:5> and arbitration protocol type<7>. The slave logic returns SS(Slave Status)=6 when a non-existing internal address is asserted.

Table 2. bit assignment of the FBCSR.

| Bit | Read | Write |
|---|---|---|
| 7 | IRQTO flag | Enable IRQTO |
| 6 | IRQSS flag | Enable IRQSS |
| 5 | IRQAT flag | Enable IRQAT |
| 4 | Error code 1 | Enable IRQSR |
| 3 | Error code 0 | B-bit |
| 2 | SS2(at interrupt) | 3cycle-bit |
| 1 | SS1(at interrupt) | CSR-bit |
| 0 | SS0(at interrupt) | G-bit |

Error code = 0 : No error
= 1 : Address cycle error
= 2 : Secondary address cycle error
= 3 : Data cycle error

## Addressing Type

Figure 5 shows an address map of the MC68000 microprocessor. The window address of the 68K-FPI is assigned to $800000-$FFFFFF, that is, A23( bit 23 of the 68000 address ) = 1 indicates window operation. The addressing type is selected by the FBCSR as follows;

| | G-bit | B-bit |
|---|---|---|
| Logical | 0 | 0 |
| Geographical | 1 | 0 |
| Broadcast | X | 1 |

x

| | CSR-bit |
|---|---|
| Data Space | 0 |
| CSR Space | 1 |

x

| | 3cycle-bit |
|---|---|
| 2 Cycle transfer | 0 |
| 3 Cycle Transfer | 1. |

Figure 6 shows how the 68000 address is converted to the FASTBUS address. The lowest 2 bits of the 68000 address are not used in the FASTBUS address, since the 68000 processor has byte address and the FASTBUS has 4 byte address. The Group Address(GP, bit 24-31) is taken from the FBBASE register(see Figure 3). The other fields are composed differently according to the adddressing type as follows;

(a) Logical Addressing: The Module Address(MA, 8 bits) is composed from MAu(3 bits) of the FBBASE and MA1(5 bits) of the 68000 address. Internal Address(IA, bit 0-15) is taken from IA1(16 bits) of the 68000 address for 2 cycle transfer and will be 0 for 3 cycle transfer. The value of Mode Select(MS) is 0 or 1 according to the CSR-bit. In the secondary address cycle, the 32-bit Internal Address(IA) are composed from the IA1(16 bits) of the 68000 address and the IAu(16 bits) of the FBBASE. Since the maximum number of the modules in a crate segment is 26, the 5-bit wide MA1 field is usually enough to cover any operation within a crate. So one does not need to change the contents of the FBBASE register frequently.

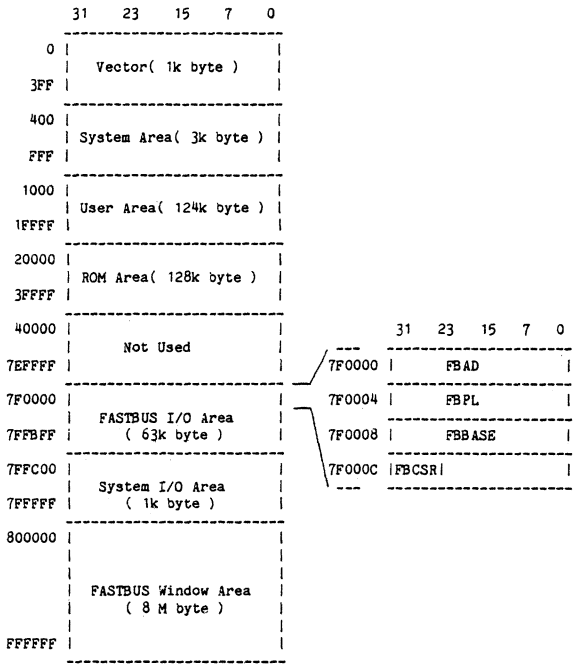



Fig.5 Address map of the 68K-FPI. Window address is assigned to $800000-$FFFFFF.

(b) Geographical Addressing: The Geographical address(GA, 8 bits) is composed from the GBAu(3 bits) of the FBBASE and the GA1(5 bits) of the 68000 address. Usually 5-bit wide GA1 field is enough for accessing the modules in a crate segment.

(c) Broadcast Addressing: The Broadcast address(BA, 8 bits) is composed from the GBAu(3 bits) and the BA1(5 bits) of 68000 address.

## Interrupt and Error Handling

Four interrupt sources are implemented in the 68K-FPI; AK/DK response time out, Slave Status(SS) response error, Arbitration Request(AR) time out and Service Request(SR).

The AK/DK response time out occurs when a AK or DK response is not received within a certain time(IRQTO time). The IRQTO flag of the FBCSR register(see Table 2) will be set when the time out occurs. Interrupt to the 68000 CPU is enabled/disabled by the Enable IRQTO bit of the register. The IRQTO time can be selected from 833 ns to 6.7 μs by a jumper pin. The timer is stoped while the Wait(WT) signal is asserted.

When a SS response other than zero is detected, the IRQSS flag of the FBCSR register will be set, and the SS value and the error code related to the FASTBUS cycle is latched in the register. Interrupt to the 68000 CPU is enabled/disabled by the Enable IRQSS bit of the register.

The arbitration time out occurs when bus mastership can not be gained in a certain time(IRQAT time). The IRQAT flag of the FBCSR register is set when the time out occures. Interrupt to the 68000 CPU is enabled/disabled by the Enable IRQAT bit of the register. The IRQAT time can be selected between 3.3 μs and 1790 sec by a jumper pin. This timer can not be stopped by the WT signal.

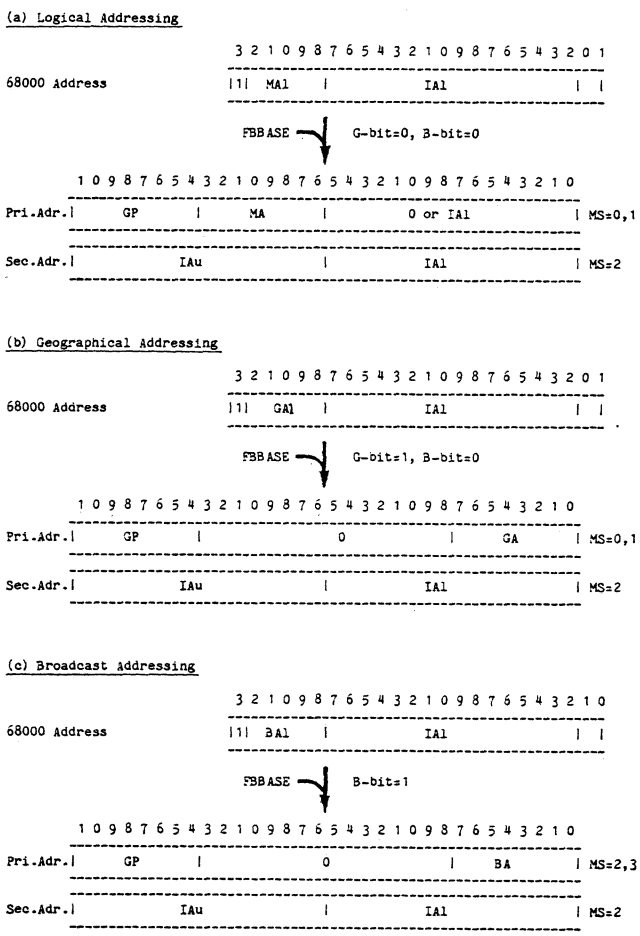



Fig.6 Schematic representation of 68000 address to FASTBUS address conversion.

The 68K-FPI can handle the Service Request(SR). When Enable IRQSR bit is set, interrupt to the 68000 CPU occurs if SR is asserted.

The interrupt flags and error status of the FBCSR are cleared as a result of the 68000 read operation to the address FBCSR+1.

## Test Results

We used our switch and display module as a slave for the test. The module takes about 300 ns for the response of acknowledge. For an example, we show the program of the block data transfer below.

```
        MOVE.L  #Buffer_Size-1,D0
        MOVE.L  #Buffer_Adr,A0
        MOVE.L  #Siave_Adr,A1
Loop    MOVE.L  (A0)+,(A1)+
        DBRA    D0,Loop
```

This program moves buffer data to slave's memory for Buffer_Size times. The FASTBUS timing signals recorded by a logic analyzer are shown in Figure 7. The addressing type is geographical, and the secondary address is used. The figure shows that the data transfer cycle is repeated in 5 micro seconds, that is, the data transfer rate is 200 k x 32-bit = 800 kB/sec.

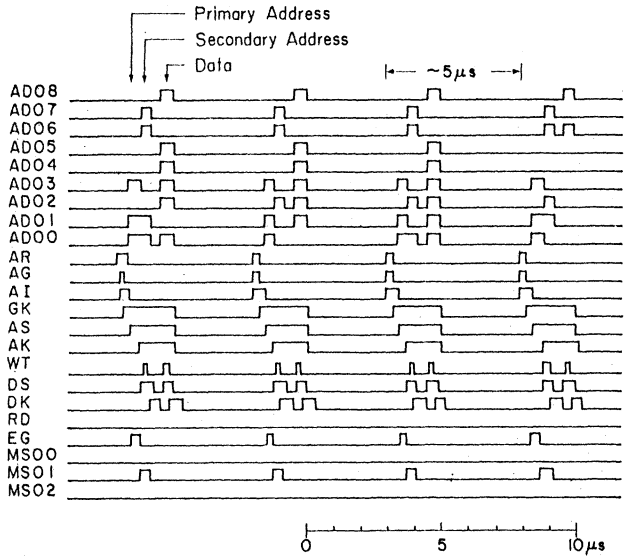



Fig.7 FASTBUS timing signals recorded by a logic analyzer. The addressing type is geographical and using secondary address. The data transfer rate is about 800 kB/sec.

## Concluding Remarks

The 68K-FPI is a versatile master module for FASTBUS. It has enough speed as local master by using the window driver. The programing in the assembler code is easy compared with the microcode programing in the bit-slice microprocessor interface. Moreover, one can control FASTBUS operation directly from high level languages if variables are assigned to window addresses.

The 68K-FPI is also used as a interface to a host computer. The software for communication to the host computer is being written.

## Acknowledgement

The authors would like to thank Y.Watase, T.Kondo and R.S.Hayano for encouraging this work.

## References

[1] Y.Arai et al., "A Status Report of FASTBUS at KEK", IEEE Trans. Nuc. Sci. NS-30, No.1(1983) 248.

[2] Y.Yasu et al., "FASTBUS Processor Interface for VAX-11/780", presented at this symposium.

[3] "Specification for Standard Routines for FASTBUS(1)", FASTBUS Software Working Group, FSGD085, FBN010, February(1983).