

30-1-11-85 JS
NOTICE
COPIES OF THIS REPORT ARE
IT HAS BEEN REPRODUCED FROM THE
available copy to permit the
possible availability.

THE SLAC SCANNER PROCESSOR: A FASTBUS MODULE FOR DATA COLLECTION AND PROCESSING*

H. BRAFMAN[†], T. GLANEMAN, A. J. LANKFORD
J. OLSEN, AND L. PAFFRATH

Stanford Linear Accelerator Center

Stanford University, Stanford, California 94305

[†] Current Address: Weizmann Institute, Rehovot, Israel

MASTER

ABSTRACT

A new, general purpose, programmable FASTBUS¹ module, the SLAC Scanner Processor (SSP), is introduced. Both hardware and software elements of SSP operation are discussed. The role of the SSP within the upgraded Mark II Detector at SLAC is described.

1. INTRODUCTION

The SLAC Scanner Processor (or SSP) is a general-purpose, high-speed, programmable FASTBUS master. It has been designed for crate-level readout and processing of data from TDC and FADC modules which instrument a new 6000-wire central drift chamber for the upgraded Mark II detector.² The SSP, however, provides a general and powerful means of moving and processing data in a FASTBUS system. Therefore, it has also been adopted by the Mark II for use at the system-level as a cable-segment master and buffer module.

2. FASTBUS SYSTEM OVERVIEW

The Mark II detector at SLAC is being upgraded to study Z⁰ production and decay along with other new phenomena at the new 100 GeV e⁺e⁻ SLAC Linear Collider (SLC).³ Cosmic ray data runs will begin in April 1985, with test runs at PEP following later in the year. Physics runs at the SLC are expected to begin in early 1987. Among the major components of this detector⁴ is a new 6000 wire drift chamber⁵ with both time (TDC) and pulse height (FADC) digitization. Readout of the drift chamber data will be accomplished via a new FASTBUS¹ system. FASTBUS was selected to perform this task for several reasons, among the most important being the high channel densities possible, the fast readout time, and the capability for providing significant amounts of preprocessing of the data before it reaches the VAX host. A partial system allowing for the readout of all TDC channels, and approximately one-sixth of the FADC channels is being prepared for the cosmic ray running, and is shown schematically in Fig. 1. (Not shown is the existing CAMAC system to read in data from the remaining detector elements).

SSPs perform two distinct roles in the data acquisition system. The primary use, as a 'crate SSP', is to provide control, buffering, and preprocessing for each of the TDC and FADC crates. The readout of all crate SSPs on a single FASTBUS cable segment is controlled by a 'system SSP'. SSPs also participate in a number of other functions, including system initialization, FASTBUS segment verification, module initialization and verification, and detector calibration.

2.1 CRATE SSPs

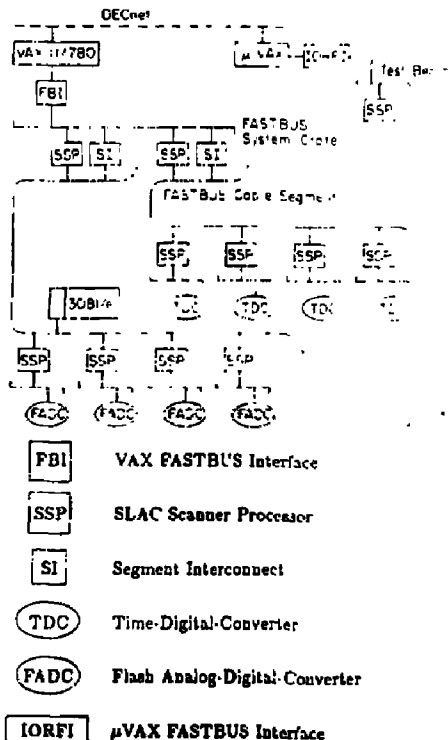
Crate SSPs reside in FASTBUS crates of data acquisition modules, consisting of LeCroy 1879 TDCs or SLAC-built FADCs.

* Work supported by the Department of Energy, contract DE-AC03-78SF00515.

During event acquisition crate SSPs are externally triggered by special trigger electronics, then read data (about 360 words/crate of TDC, 400 words/crate of FADC information for an average event) into a buffer. Data in this input buffer is then processed internally and placed into an output buffer. Preprocessing functions depend upon the module being read out. For TDC modules, this preprocessing will include:

- pedestal correction for each channel;
- translating TDC channel numbers into wire numbers;
- associating leading edge times with trailing edge times;
- reformatting the structure of each datum;
- sorting of the data into the final output buffer format.

Finally, a message is sent to the controlling system SSP that processing is complete.



3081/e SLAC/CERN On-line Emulator - Processor

Figure 1. Preliminary Mark II FASTBUS System (by March 1985).

2.2 SYSTEM SSPs

The system SSPs reside in a FASTBUS crate between the VAX host computer and the data acquisition crates. During event acquisition each system SSP is responsible for directing the flow of data between crate SSPs (on a cable segment) and the host or a 3081/e on-line processor.⁶ This function includes resetting and readying crate SSPs for a new event, reading out and buffering processed data from the data acquisition crates, and signaling (interrupting) the host when a new event is ready.

3. SSP HARDWARE OVERVIEW

A simplified block diagram showing the logical components of the SSP is shown in Fig. 2. The SSP occupies two FASTBUS boards. The control board (~400 chips) contains I/O circuitry (and connections to the two FASTBUS segments), μ code PROMs, and the program memory. The CPU board (~200 chips) contains the CPU, additional μ code PROMs, and the data memory. These boards are pictured in Fig. 3. Internal logic levels are TTL, employing standard chips. Conversion to ECL is provided for the FASTBUS connections. Together, these boards dissipate approximately 100 Watts.

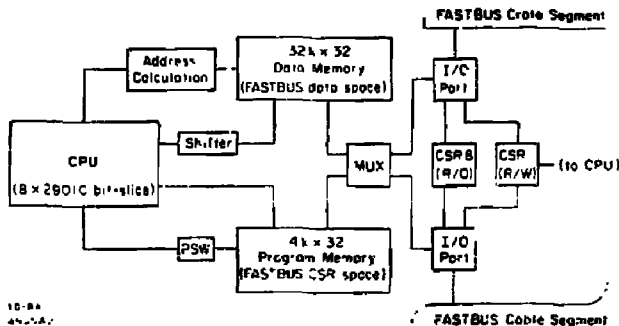


Figure 2. SSP Simplified Block Diagram

Connection to the FASTBUS crate segment is made through the normal backplane connector on the control board, while the cable segment attaches to the front of the module on the first prototypes. Internal logic and data are transferred between boards via a strap on the auxiliary backplane connectors, and a small cable near the front of the boards. Certain features of the prototype SSPs will be modified to embrace more of the spirit of the FASTBUS philosophy, such as attaching the cable segment to the auxiliary backplane connector. A front panel displays the status of current FASTBUS and CPU activity.

4. SSP FASTBUS CHARACTERISTICS

The SSP contains two I/O ports, one to a FASTBUS crate segment and the other to a FASTBUS cable segment. The primary functions of these ports include external control of the SSP and the transfer of large blocks of data. FASTBUS response times of the SSP (with the exception of block transfers) are limited by the clock cycle. During block transfers, the internal clock period is narrowed resulting in a measure of DS-to-DK

or DK-to-DS response time of approximately 90 ns.

Masters on either segment may address the SSP as a slave if the CPU is not running or if the interrupt mask bit is set. As a slave, the SSP responds to geographical and broadcast (cases 1,2,4, and 5) address cycles. Recognized data cycles include random, secondary address, and handshake block transfer modes. Once attached as a slave, the CPU is idle and does not participate in the exchange. Only one I/O port at a time may be attached and actively engaging in data cycles.

As a master, the SSP supports a wide range of FASTBUS protocols. For example, the mode select bits (MS0 and MS1) during an address cycle are instruction parameters which support all current addressing modes. Pipelined write (SSP-to-slave) block transfers are supported in addition to random, secondary address and block transfer data cycle modes. During FASTBUS master operations the I/O logic maintains a Program Status Word (PSW). When an unexpected SS code or timeout is encountered, the I/O logic dumps the PSW to data memory and causes a program trap.

5. SSP CPU CHARACTERISTICS

CPU power in the SSP is provided by eight AMD2801C bit slice processor chips configured in a 32-bit architecture. Instructions from a 3-level pipeline are decoded by thirteen 8-bit PROMs to control data movement and CPU functions. Separating program and data memories significantly improves the performance of this device. In addition, a dedicated shifter is utilized both for optimizing byte shift operations and accommodating other half-word and byte instructions. The basic cycle time of 120 ns (8.3 MHz) is narrowed to 80 ns (12.5 MHz) for certain multiple-cycle operations such as multiply, divide, shift, and FASTBUS I/O block transfers. Other features of the design include dedicated address calculation logic, branch logic and word count logic.

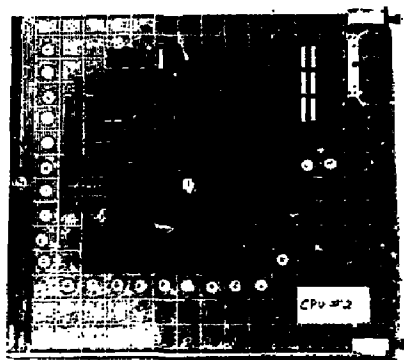
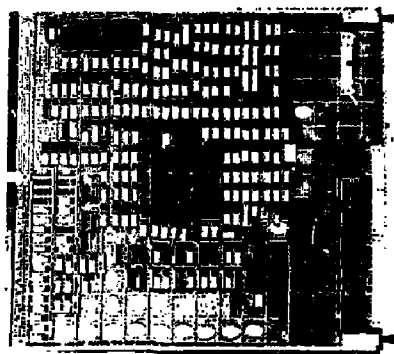


Figure 3. SSP Control (top) and CPU (bottom) boards

5.1 SSP INSTRUCTION SET

Microcode PROMs in the SSP provide a rich instruction set, facilitating the integer arithmetic, control, and FASTBUS I/O tasks required by the Mark II applications. Instructions are stored in the program memory in the IBM System/370 instruction format, prior to being decoded by the PROMs. Many instructions complete within one or two clock cycles. Notable exceptions include the multiply, divide, load/store multiple, and shift instructions as well as FASTBUS I/O block transfers.

In the event of an error or interrupt (if enabled), a hardware trap forces program execution to continue from location 0 where an error-handling routine resides. Errors include CPU exceptions (unsupported opcode, division by zero, and (if enabled) overflow), and FASTBUS I/O errors (timeout, and unexpected slave status response).

The SSP's instruction set may be subdivided into four categories.

1. **IBM Integer Instruction Set** - The following 62 IBM System/370 instructions are emulated exactly by the SSP. These instructions include:

A, AH, AL, ALR, AR, BAL, BALR, BC, BCR, BCT, BCTR, BXH, BXLE, C, CH, CL, CLI, CLR, CR, D, DR, IC, L, LA, LCR, LH, LM, LMR, LPR, LR, LTR, M, MH, MR, MVI, N, NI, NR, O, OI, OR, S, SH, SL, SLA, SLDA, SLDL, SLL, SLR, SR, SRA, SRDA, SRDL, SRL, ST, STC, STH, STM, TM, X, XI, and XR.

2. **Optimized IBM instructions** - This set of five instructions utilizes the power of the dedicated shift circuitry, enabling shifts of one byte per machine cycle and four bytes per machine cycle.

SLLB Shift Left Logical by Bytes
SLAB Shift Left Arithmetic by Bytes
SRLB Shift Right Logical by Bytes
SRAB Shift Right Arithmetic by Bytes
SR32 Shift Right Double Arithmetic by 32 bits
(used in preparation for integer divides)

3. **Miscellaneous Instructions** - The first three miscellaneous instructions approximate the functions of the corresponding IBM codes, while the remaining two support FASTBUS operations.

SPM Set SSP Program Mask (overflow mask)
SSM Set SSP System Mask (interrupt mask)
LPSW Load SSP PSW
WCSRO Write to CSRO
RCSRO Read from CSRO

4. **FASTBUS I/O Instructions** - A set of 24 I/O instructions provides the SSP programmer with access to FASTBUS. Performing a FASTBUS action requires the execution of one or more of the instructions listed below. This set of instructions allows for both primitive and complex actions. All primary address instructions imply a bus arbitration cycle if bus mastership was not achieved or maintained by the previous instruction. Execution of a non-I/O instruction results in the release of any existing AS/AK lock initiated by the SSP.

FASTBUS Primitive Instructions

FPA	Primary Address
FPRNTA	Secondary Address, read
FPWNTA	Secondary Address, write
FPR	Random read
FPW	Random write
FPW1	Random write immediate
FPRBC	Block transfer read, end on count
FPRB2	Block transfer read, end on SS=2
FPRB2C	Block transfer read, end on SS=2 or count (whichever occurs first)
FPWBC	Block transfer write, end on count
FPWBP	Block transfer write, pipelined, end on count

FASTBUS Macro Instructions

FRD	FPA+FPWNTA+FPR
FRDBC	FPA+FPWNTA+FPRBC
FRDB2	FPA+FPWNTA+FPRB2
FRDB2C	FPA+FPWNTA+FPRB2C
FWDI	FPA+FPWNTA+FPWI
FWDBC	FPA+FPWNTA+FPWBC
SKD	FPWNTA+FPR
SRDBC	FPWNTA+FPRBC
SRDB2	FPWNTA+FPRB2
SRDB2C	FPWNTA+FPRB2C
SWD	FPWNTA+FPW
SWDI	FPWNTA+FPWI
SWDBC	FPWNTA+FPWBCI

5.2 SSP MEMORY ORGANIZATION

The memory layout of the SSP is indicated in Fig. 4. CS0 contains processor control bits, broadcast class, and access to the Service Request (SR) and Reset Bus (RB) lines. CS8 contains the single arbitration level used on both crate and cable segments. Current CSR registers and their bit assignments will be rearranged and new registers added to conform to the current FASTBUS standard in the production SSP. The 4K words of program memory reside within Program CSR space. The 32K words of data memory are accessible both through Data Space or through Parameter CSR Space, a feature designed to give an external master access to CS0 and data memory without dropping the AS/AK lock.

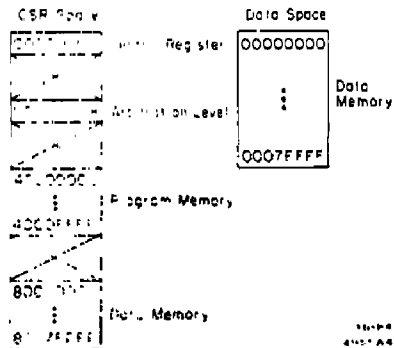


Figure 4. SSP Memory Organization

6. Software Support

Operation of the SSP requires a certain amount of software support. Equipped with this support, application programs can be written in a high-level language. The steps necessary to execute a program on the SSP in the Mark II environment are diagrammed in Fig. 5. Software may be written in IBM assembler language or FORTRAN, keeping in mind the restrictions inherent in the instruction set (e.g. no floating point operations). The IBM assembler or FORTRAN compiler next processes the program. A SLAC-written translator/linker program then performs a variety of necessary functions:

1. Warns of illegal op codes.
2. Separates program and data memories.

3. Alleviates a particular register contention problem arising from the SSP instruction pipeline.
4. Optimises certain sequences of code (e.g. shift operations).
5. Links together subroutines, relocates program and data memories, and loads an error handling routine.
6. Generates symbolic information tables for use with a debugger.
7. Produces an output file in one of several formats suitable for downloading to a VAX or other host computer.

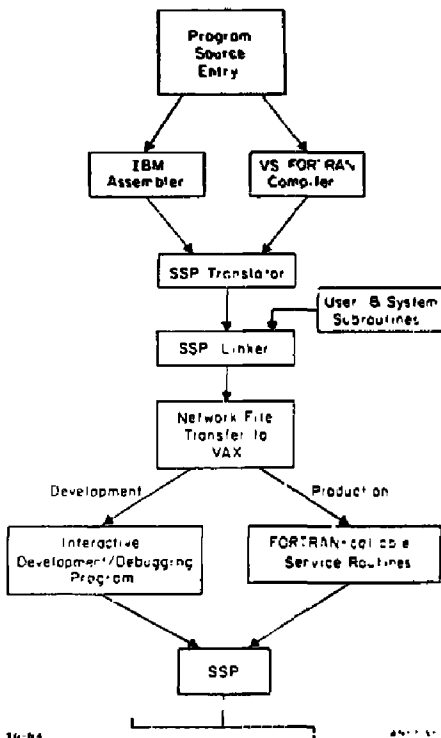


Figure 5. Steps to Executing a Program on the SSP

SSP load modules are then transferred to a VAX 11/780 over a BITNET network link. Once on the VAX, the user has two choices for program execution. An interactive program called SNOOPY provides many services suitable for SSF software development, such as downloading an SSP with a load module, starting the processor, reading/writing common blocks, dumping memory, and examine/deposit functions to individual memory locations. Comparable services are also available as a set of FORTRAN-callable subroutines suitable for use in a production environment.

A growing body of software exists at SLAC to support the SSP. One collection of programs perform diagnostic testing of the SSP and its memory. A set of FORTRAN-callable subroutines are available to perform FASTBUS operations from the

SSP. Work is in progress to write SSP software emulating certain segment interconnect functions. These functions eliminate the need for SIs in every data acquisition crate; furthermore, the SSP is currently the only device at SLAC capable of communicating with independent crate and cable segments.

7. CURRENT STATUS AND FUTURE PLANS

There are currently three debugged and operating SSPs at SLAC in wire-wrap form. An additional five wire-wrapped SSPs will be delivered by the end of 1984. Work is progressing on a printed circuit version of the SSP with an initial delivery of eight units scheduled for April 1985. The remaining complement of 19 additional SSPs for the Mark II is expected by Fall 1985 in time for data taking at PEP.

8. SUMMARY

A new high speed, general-purpose, programmable FASTBUS module has been designed at SLAC providing both autonomous CPU power and data transfer capability on separate crate and cable segments. Prototypes have been built and thoroughly tested. Deliveries of the production version SSPs on PC boards are expected by Spring 1985. It is expected that 25 units will be in service at the Mark II detector by this time next year.

REFERENCES

1. U.S. NIM Committee, "FACTBUS: Modular High Speed Data Acquisition and Control System for High Energy Physics and Other Applications," DOE/ER-O 189, December 1983.
2. Caltech, et.al., "Proposal for the Mark II at SLC," CALT-68-1015, April 1983.
3. SLAC, "SLAC Linear Collider Conceptual Design Report," SLAC-Report-229, June 1980.
4. A. J. Lankford, and T. Glanzman, "Data Acquisition and FASTBUS for the Mark II Detector," SLAC-PUB-3241, October 1983.
5. G. Hansen, "A New Drift Chamber for the Mark II at SLC," SLAC-PUB-3317, April 1984.
6. P. F. Kuz, et.al., "The 3081/e Processor," SLAC-PUB-3332, April 1984.
7. See IBM System/370 Principles of Operation, 10th edition, May 1963.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.