11
-9

2130

LA-4916-MS

AN INFORMAL REPORT

MASTER

A Microprogrammed Branch Driver (MBD) for a PDP-11 Computer

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7405-ENG. 36

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or 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.

In the interest of prompt distribution, this LAMS report was not edited by the Technical Information staff.

Printed in the United States of America. Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.95

LA-4916-MS
An Informal Report
UC-32
ISSUED: April 1972

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

A Microprogrammed Branch Driver (MBD) for a PDP-11 Computer

by

Lavon R. Biswell
Robert E. Rajala

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or 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.

A MICROPROGRAMMED BRANCH DRIVER (MBD) FOR A PDP-11 COMPUTER

by

Lavon R. Biswell and Robert E. Rajala

ABSTRACT

A microprogrammed branch driver (MBD) is the interface between a Digital Equipment Corporation (DEC) PDP-11 computer and a multicrate CAMAC system. This unit is a multiple, direct memory access (DMA) channel branch driver.

The requirements for an MBD are discussed in the introduction, and stem from a Los Alamos Scientific Laboratory (LASL) study group report on the design of the Los Alamos Meson Physics Facility (LAMPF) data-acquisition system. Because of the desire for a standard system, the types of experiments and data rates, and the varying complexity of data-acquisition systems, it was decided that a microprogrammable, multiple DMA channel branch driver was required.

Basic specifications and capabilities of the MBD are discussed and a block-diagram-type operation is presented, which includes the modes, registers, priority structure, interrupts, DMA channels, instruction set, branch driver, and initialization and run procedures. Modes of operation and design options are given, and the standard logic used to fabricate the MBD, the hardware system, is discussed. Included are diagnostic and standard driver routines, and a program example.

INTRODUCTION

In August 1970 a group met at Los Alamos to develop a system design for the on-line data-acquisition facilities at LAMPF. The results of this study are documented in the report, "LAMPF Data-Acquisition System."[1] A brief summary of this report is given in order to establish the requirements for the microprogrammed branch driver (MBD) discussed in this paper. The work evolved from a cooperative venture between LASL and Bonner Research Laboratories, Rice University, to develop a CAMAC branch driver for the PDP-11 computer.[2]

It was concluded that the system should be developed around a small, dedicated computer. The computer would acquire, preprocess, and record data; monitor the condition of the experimental apparatus; execute necessary control functions; and perform preliminary analyses which forecast the results of the experiment. It was proposed that LAMPF provide hardware and software support for the standard system. Recognizing the need to share expensive peripherals and to have access to arithmetic capability of a large computer, it was recommended that a computer-based terminal be installed at LAMPF and linked to the Central Computing Facility (CCF) at the Laboratory. All the dedicated computers would have a link to the terminal, which would give access to its peripherals, accelerator data, and a path to CCF.

It was recommended that in order to accommodate a large number of users with varied interests at LAMPF, all interfacing between the experimental equipment and the dedicated computers be implemented in the CAMAC standard. The results of the study led to a selection of the DEC PDP-11 as the standard for the LAMPF data-acquisition system and terminal computer.

The CAMAC system of instrumentation[3,4] has been adopted by numerous laboratories in Europe and the United States as a standard method of interfacing research apparatus, with a goal of establing a stable boundary between instrumentation and computing. A brief description of CAMAC will help clarify the objectives of the unit described in this report. CAMAC replaces the great variety of I/O buses found on computers with a single, nonproprietary design standardized both mechanically and electrically. The system features a crate which will accept up to 24 modules, and a branch which will accept up to seven crates. The crate dataway and the branch highway provide the communication link to the computer. The CAMAC specification restricts the instrumentation contained in a module only to the extent necessary to insure compatability with the crate and dataway.

The dataway has a 24-bit read bus, a 24-bit write bus, and a control bus. The control bus provides for 16 subaddresses and for performing up to 32 different operations on a module (function codes). Provision is made for polling the station via a common "Q" response line and for verifying valid commands by a common "X" response line. In addition to the buses, each individual module has a pair of private lines for module station selection (N-line) and one for its service requests, LAM (Look-at-me).

The above characteristics are not embodied in the I/O bus of any computer and thus CAMAC must itself be interfaced to the computer. The interface must resolve the line and logical differences between the two structures and its design will therefore have a significant influence on the performance of the total system.

To review requirements, the unit must interface the PDP-11 computer to the CAMAC branch highway. It is also desirable to free the PDP-11 computer for as much real-time data analysis as possible. Many of the experiments have high event rates and with the coming of age of multiwire proportional counters, many of the experiments may have very high data rates. The minimum system requirements are two DMA channels for experimental data (scalers, wire chambers, pulse-height analyzers, etc.)--one DMA channel to display accumulated data on a storage scope and one DMA channel for communication with the LAMPF terminal via the CAMAC data link.

From the requirements it was decided that what was needed was a multiple DMA channel branch driver that could be easily modified or programmed. With the advent of microprogrammable processors, this would give the flexibility required and exploit the maximum capabilities of the PDP-11 computer and the CAMAC system. Therefore, this is the approach taken in the design of the MBD discussed in this report.

GENERAL SPECIFICATIONS AND OPERATION OF MBD

Figure 1 shows a functional diagram of the MBD. The unit can be divided into three major areas: the PDP-11 computer interface, the CAMAC branch driver, and the microprogrammed processor.

The branch driver is a very conventional design. It contains three basic registers: the command register (CNAF), the 24-bit branch data register (BDR), and the 24-bit graded-L register (GLR). In order to get around the problem of the command requiring 17 bits, three different command types are defined: read, write, and control/test. Bit F8 is omitted from the command word and is supplied by the processor, depending on the type of command in which it appears. The processor is in complete control of the branch driver. It controls reading and loading the registers, starting the branch operation, and testing the "Q" and "X" lines. A branch time-out error will generate an interrupt identifiable in the CSR register. The computer INIT initialize command generates a "BZ" command to the branch. A branch demand (BD) will result in a graded-L (GL) operation if the processor is in the stop mode, but if in the run mode, the processor will not allow a GL operation until it completes the current job and generates an exit command. The results of a GL operation are that the GL's from the branch are stored in the GLR. Bits 17-24 of the GLR result in DMA channel requests 0-7. Bit 24 corresponds to channel 7 and is highest priority. Bits 16-1 of GLR are masked by the MASK register in computer interface and appear as 16 unique interrupt vectors to the PDP-11 computer. Bit 16 is the highest priority L request. The MASK register can be loaded only from the PDP-11. Since the BDR is 24 bits in length and the processor and PDP-11 are 16-bit machines, two transfers are required to read

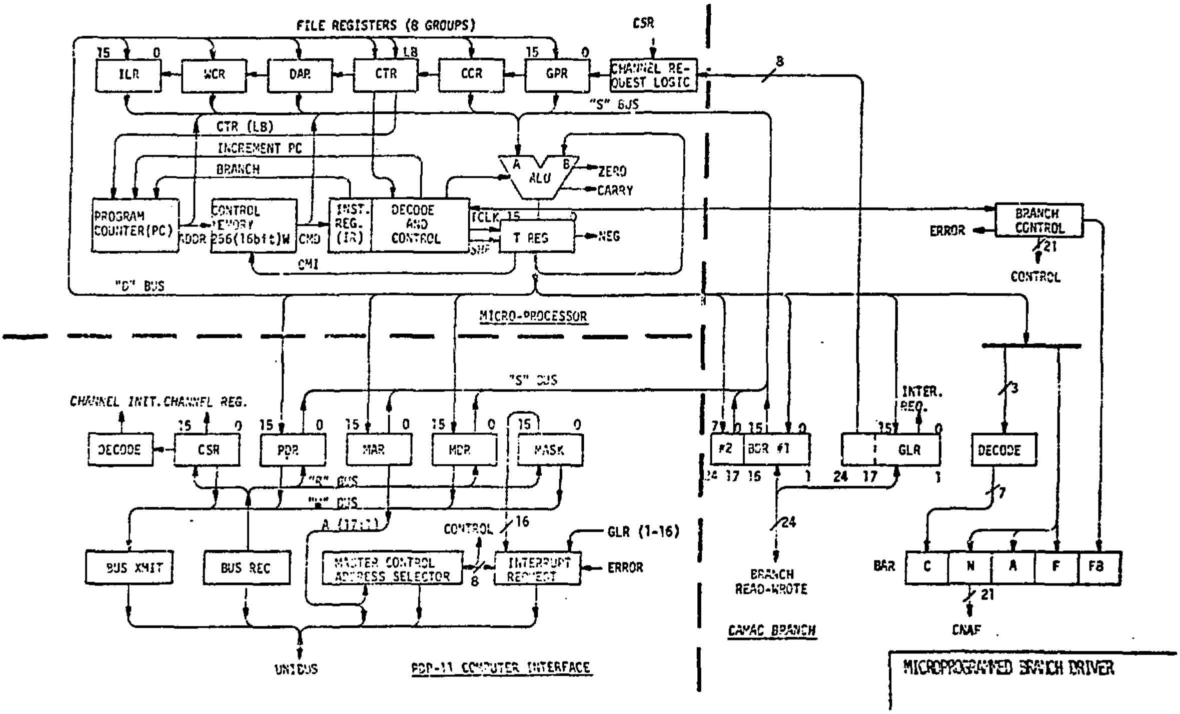


Fig. 1

or load the BDR. Bits 24-17 are the low byte of the second transfer.

The computer interface part of the MBD is very similar to the standard DEC interface.[5,6] The five registers of the interface are detailed in Table I. The memory address register (MAR) and the memory data register (MDR) are the DMA channel registers and are controlled from the processor. The processor is bus master during all DMA transfers. A bus request is made when the processor executes a data-channel read or write. The bus is released at the end of each transfer unless the unit has been commanded into the 2-cycle mode for increment and add-to-memory function; then it is released at completion of the second transfer. A UNIBUS transfer time-out error sets an error interrupt to the computer and an error flag in the CSR register.

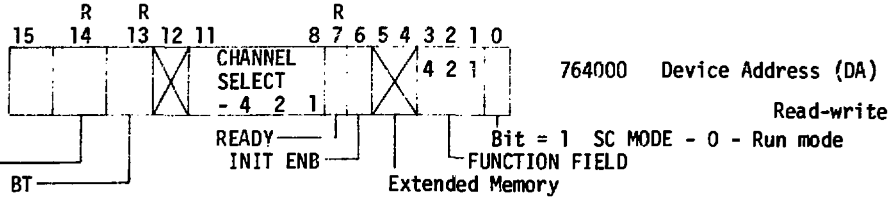
The other registers of the interface are the control and status register (CSR), the program data register (PDR), and the MASK register. These registers may be given any device addresses allowed by DEC and are used to initialize the processor; that is, load the control memory of the processor from the computer and read it to verify correct loading. They are also used in the single-cycle (manual) mode for controlling and testing the processor. The CSR register is used by the PDP-11 to make a channel request when the processor in the run mode by setting the desired channel number into the channel-select field and run-mode bit set. The balance of the CSR will be discussed in depth in the manual-mode section.

The registers are buffered and appear to the computer UNIBUS as a single unit load. The DMA channel looks like one channel to the UNIBUS and the interrupts appear as one hardware interrupt, but generate 25 unique vectors starting at location 400 in the PDP-11. The error interrupt is the highest priority, the eight end-of-channel operations are the next level, and the 16 GL's are the lowest level. All 25 interrupts appear as one level to DEC and can be assigned to any one of the BR levels. The first units are assigned to the BR5 level.

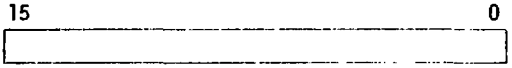
TABLE I
REGISTERS OF MBD

Dec. Reg.

CSR #1



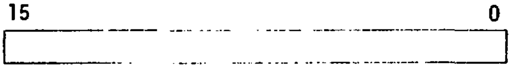
PDR



764002 (DA) Read-write

Used to load IR & Control Memory from PDP-11

MASK

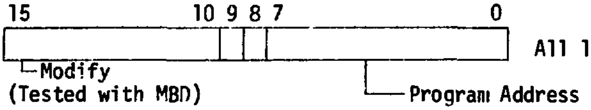


764004 (DA) Read-write

MASK's for GL 1-16 interrupts

File Reg.

CTR



All 16-bit registers

ILR	Instruction list (list pointer)
WCR	Word Count (Event length or # events in list)
DAR	Data Address (Data List START ADDRESS)
CCR	CAMAC Command (CNAF 16 bits, except F8)
GPR	General Purpose (Free register)

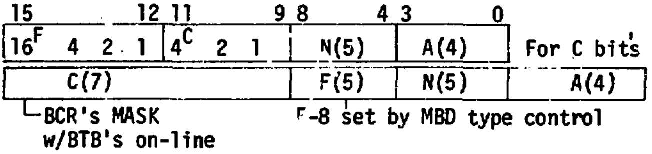
Channel Reg.

MAR	Memory Address Reg. (DMA start address)	Read-write from MBD
MDR	Memory Data Reg. (DMA Data Reg.)	

CAMAC Reg.

BDR #1,#2	Branch Data Register (#1 is 1-16) (#2 is 17-24)
GLR	Graded L's Reg. (1-16 CAMAC INT., 17-24 channel req.)
BAR	Branch Address (command register)

CNAF



For C bits binary # is the crate No.

21 bit

The microprogrammed processor is a full-fledged processor and is the control device that gives the MBD the speed and flexibility required. The processor's prime function is to command the branch and transfer data between the branch and the computer. Microprocessor design has developed to the point where it is faster and requires less hardware to connect several registers via buses and a processor, than with gates alone.

The heart of the processor is the arithmetic and logic unit (ALU) which connects the Source bus and the Destination bus and allows transfers between any of the registers connected to the buses in one micro instruction of the ALU. One of the

TABLE II

MBD INSTRUCTION SET

15 12	11 8	7 4	3 0	
OP	CTRL	SCR	DST	IR - Instruction Register

NORMAL OP-Code / IR Instruction

15	14	13	12	Name		Speed
0	0	0	0	MOV	Move S to D, CTRL	1 cycle
0	0	0	1	INM	Increment S, store in D & S, CTRL	1 cycle
0	0	1	0	DEM	Decrement S, store in D & S, CTRL	1 cycle
0	0	1	1	ADD	ADD T to S, store in D, CTRL	1 cycle
0	1	0	0	SUB	SUBTRACT T from S, store in D, CTRL	1 cycle
0	1	0	1	IOR	Inclusive OR T, S; store in D, CTRL	1 cycle
0	1	1	0	XOR	Exclusive OR T, S; store in D, CTRL	1 cycle
0	1	1	1	AND	AND T, S; store in D, CTRL	1 cycle

15 12	11 8	7 0
OP	COND	ADDR

NOT NORMAL OP-Code / IR Instruction

15	14	13	12	Name		Speed
1	0	0	0	BCT	Branch to ADDR if COND TRUE PC = PC+1 for COND FALSE	1 cycle
1	0	0	1	BCF	Branch to ADDR if COND FALSE PC = PC+1 for COND TRUE	1 cycle
1	0	1	0	JVC	JUMP VIA LOW BYTE IN CTR (File Reg.)	1 cycle

15 12	11 8	7 0	
OP	MOD	ADDR	Bit(11 = 1), S or D is PDR Reg.

15	14	13	12	Name		Speed
1	0	1	1	STO	Store value in T REG into CM at ADDR	EXT. cycle
1	1	0	0	LOD	Load value at ADDR into T REG	EXT. cycle

15 12	11	10	9	8	7 4	3 0	
OP					╳	N	50 ns/bit

15	14	13	12	Name		Speed
1	1	0	1	SHF	SHIFT N (1-16) bits, Bit (8=1) Right, (8=0) Left Bit (9=1) Normal, (9=0) Rotate	
1	1	1	0	LCI	LOAD CTR LB from LB of IR through ALU	EXT. cycle
1	1	1	1	-	Spare	

functions of the processor is to create (multiplex) and control eight DMA channels. The information pertinent to the DMA operation is stored in 48 file registers, all of which are connected to the S and D buses. The registers are organized in eight groups of six registers; each group contains information for one of the eight DMA channels. The branch and computer interface registers are connected to the S and D buses, which gives the processor control of the communication between the CAMAC branch and the computer I/O. The control memory (CM) contains the microprograms that define the transfer sequences. The read-write memory has 256 16-bit words and an access time of approximately 70 nsec.

Microinstructions are executed in four clock periods, which are initially set at 400 nsec. The basic clock is 10 MHz, but can be adjusted over a range of 1-25 MHz. The instructions are detailed in Table II and the instruction register (IR) decoding is shown in Table III. The instruction set is divided into normal and not-normal instructions. Each of the normal instructions can select one of 16 control functions, a source register, and a destination register. The normal instructions are:

TABLE III

MBD INSTRUCTION REGISTER

	15 12	11 8	7 4	3 0
IR	OP	CTRL TEST	SRC	DST

IR 11 10 9 8	CTRL or	TEST	IR 7 6 5 4	SRC	IR 3 2 1 0	DST
0 0 0 0	NONE	NONE	0 0 0 0	NONE	0 0 0 0	NONE
0 0 0 1	DCH 2C	DCH BSY	0 0 0 1	ILR	0 0 0 1	ILR
0 0 1 0	DCH RD	BR BUSY	0 0 1 0	DAR	0 0 1 0	DAR
0 0 1 1	DCH WT	INT BSY	0 0 1 1	WCR	0 0 1 1	WCR
0 1 0 0		NEG FLAG	0 1 0 0	CCR	0 1 0 0	CCR
0 1 0 1	BRN C	ZERO FLAG	0 1 0 1		0 1 0 1	CTR
0 1 1 0	BR CC	ZERO LO BYTE	0 1 1 0	GPR	0 1 1 0	GPR
0 1 1 1		CARRY FLAG	0 1 1 1	PC	0 1 1 1	CTR LB
1 0 0 0	EXIT 1	"Q" FLAG	1 0 0 0		1 0 0 0	DGLR (1 - 16)
1 0 0 1	EXIT 2	"X" FLAG	1 0 0 1	MDR	1 0 0 1	MDR
1 0 1 0	EXIT 3	CTR BIT #10	1 0 1 0	MAR	1 0 1 0	MAR
1 0 1 1	XEQ INT (X)	CTR BIT #11	1 0 1 1	BDR #1	1 0 1 1	BDR #1 (1 - 16)
1 1 0 0	PRL ST (X)	CTR BIT #12	1 1 0 0	BDR #2	1 1 0 0	BDR #2 (17 - 24)
1 1 0 1		CTR BIT #13	1 1 0 1		1 1 0 1	BAR
1 1 1 0	RST CEL	CTR BIT #14	1 1 1 0	PDR	1 1 1 0	PDR
1 1 1 1	BZ COM	CTR BIT #15	1 1 1 1	CCL	1 1 1 1	

more, increment, decrement, add, subtract, inclusive OR, exclusive OR, and AND. The not-normal set consists of the branch-if-true, the branch-if-false, jump via a low byte in CTR, store, load, shift-T-register, and load CTR LB from IR LB. The branch instructions can select one of 16 conditions for test.

By programming convention only the file registers are assigned a particular function with the exception of the control register (CTR). The low bvte of the CTR is connected directly to the PC register and contains the address of the service program in control memory. Bits of the high order byte can be tested by the control program to determine which alternative procedure to execute. The instruction location register (ILR) contains the address of the next word in the PDP-11 core memory instruction list.

The data address register (DAR) contains the pointer to the next word in the data list. The word count register (WCR) contains the number of words to be transferred to the data list. The CAMAC command register (CCR) contains the CAMAC function and CAMAC device address, minus the F8 bit, which is supplied by the type of control command that loads the CCR command into the branch address register (BAR). The general purpose register (GPR) is a free register and may be used as the programmer desires. For an example, it may be used to create loops, as an index register, or as a counter.

The channel request logic is the area that ties the three major areas together and is the starting point for all run-mode operations. Understanding this area of channel requests, interrupts, exits, and priorities is the key to understanding the MBD and its capabilities.

After initializing the processor, which will be discussed later, operation begins by making a channel request. Each of the eight channels has two sources: the PDP-11 computer and a GL in CAMAC. The computer request is higher priority than CAMAC. The eight channels have a priority structure 0-7 with channel 7 the highest priority. With the selection of a channel the group of six file registers associated with that channel are connected to the S and D buses. During the arbitration of the priority the program counter is reset to zero; if the channel selected to run has as a source the PDP-11, then the PC register is incremented. A minimum of 150 nsec delay between incrementing the PC and the first read of the control memory allows sufficient time for the memory addressing to stabilize.

The basic operation of the processor is that a channel is selected and if its source was CAMAC, then the instruction in control memory location zero is executed. Location zero will contain a JVC instruction, which is a jump via low byte in CTR register. This allows each channel a link to its own microprogram. If the source was the PDP-11 then execution will start at location "1" of central memory. Starting at location "1" will be a file register initialize routine using channel transfers. This requires approximately 20 instructions for the six file registers in a group, which will be used by each channel during system initialize to load all the file registers and will be used as required by channels to reinitialize during the run phase. When a transfer is completed, then an end-of-block interrupt for that channel is sent to the PDP-11 by a control command. The PDP-11 recognizes the interrupt and issues a channel request that will reinitialize that channel, if that is desired.

Once the processor is executing a program it cannot be interrupted and will relinquish control to the channel request logic by executing one of three exit control commands. Three types of exits are required to establish the desired control of the three registers in the channel select logic. The three registers are: the program request latch (PRL), the channel enable latch (CEL), and the channel initialize latch (CIL). The PRL register can be selectively set with a control command and can be selectively reset with an exit 2 command. The PRL is used to hold a channel request while giving up temporary control to the channel select logic with an exit 1 command. If a channel of higher priority is not requested, then control will return to the channel in the PRL. Control will eventually return, even if other higher priority channels take the processor for periods of time, when PRL is the highest priority. The CIL register is loaded from the CSR register channel select bits. The CIL is reset by any exit if the channel running was a computer source. The CEL register is a mask register for the 8-channel requests originating in CAMAC. Exit 2 will do a selective set of CEL which is the way of enabling that channel. Exit 3 will cause a selective reset of CEL and will be used at the end of the list when chaining. A control command will do a complete reset of the CEL if required.

The operation of the processor is such that any exit causes a temporary stop. If there is no request, the unit remains in the stop mode until a request is made from either the computer or CAMAC. If there is a request from CAMAC (BD) then a BG operation is automatically started and the channel select logic waits for completion of the GL cycle before priority is arbitrated. The CIL register may contain a request and also the PRL may have a request.

The timing diagrams for the system are shown in Figs 2 and 3. The basic clock for the MBD is an adjustable 10 MHz oscillator, which is on whenever power is applied to the unit. The basic timing is derived from a four-stage ring counter. If the clock is set at 10 MHz then the four subcycles of the counter are 100 nsec each. By proper gating of the counter subcycles (TS1-TS-4) with the clock and not-clock signals, then any 50 nsec increment of time can be obtained.

The timing for a normal instruction is shown in Fig. 2. During TS1 the instruction is loaded into the IR register and decoded. At TS2 the source is enabled to the S bus and the PC counter is incremented. During TS2 and the first half of TS3 the ALU operation takes place and at TS32 the output of ALU is loaded into the T register and gate to the D bus. At TS4 the data on the D bus are enabled to the destination register. If a control command is programmed this occurs at TS42.

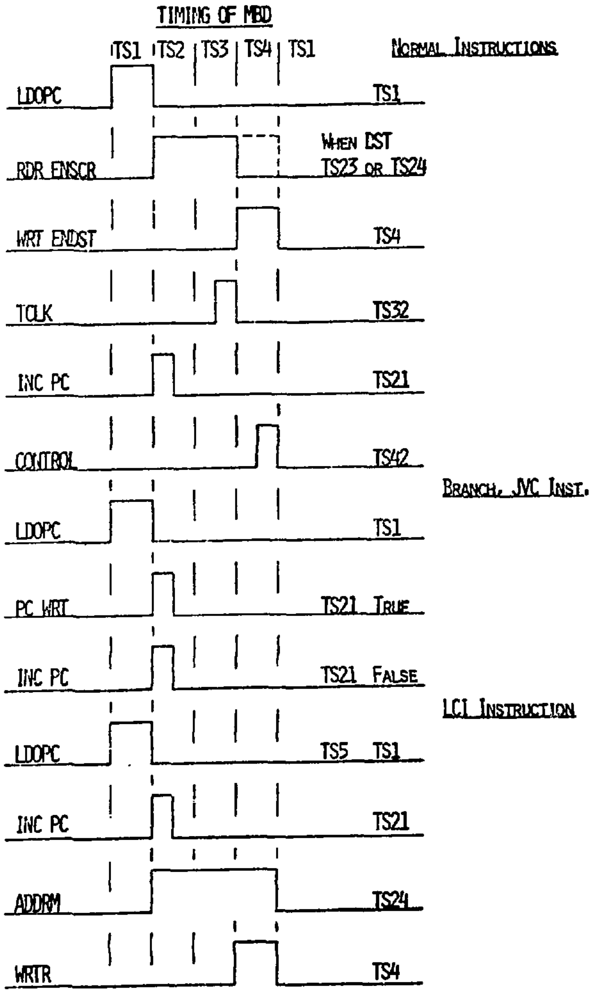


Fig. 2

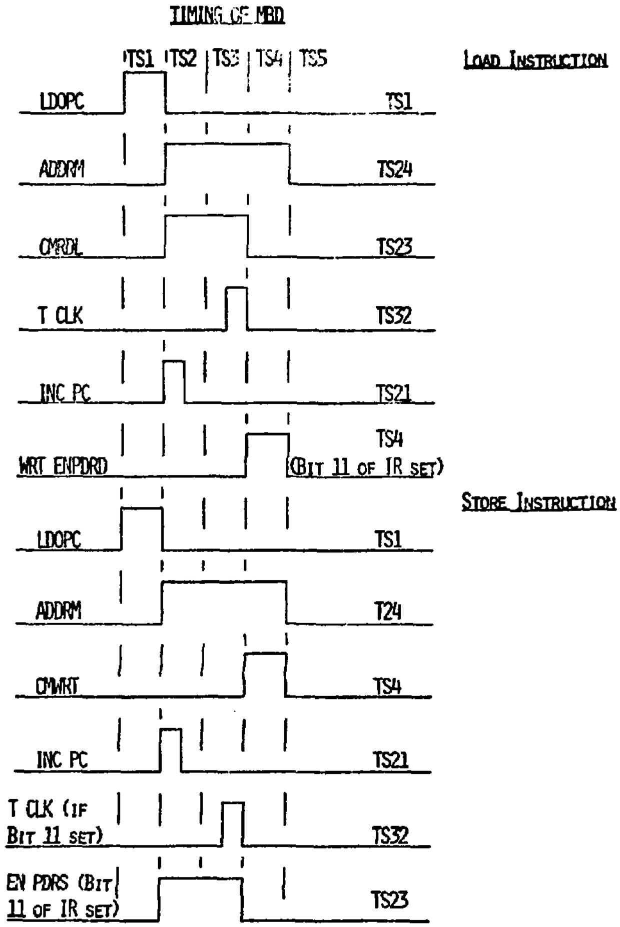


Fig. 3

The not-normal instructions require some different and special timing. The BCT, BCF, and JVC instructions are very simple and require only TS1 and TS2 times. At TS1 the IR register is loaded and decoded. At TS2 the PC counter is incremented or loaded with the low byte of the IR register, whichever is correct for the condition tested. The JVC instruction is the same except at TS2 the low byte of CTR is loaded into the PC counter.

The shift (SHF) instruction requires special timing. The shift is on the T register and is an N (1-16) bit right or left, normal or rotate, shift and the speed is 50 nsec/bit. The timing of the SHF instruction is such that TS1-4 is normal and if additional time is required to complete the shift, the ring counter is held in TS4 position until the shift is completed; then it is released to perform the next instruction.

The LOD, STO, and LCI instructions require an extra subcycle (TS5). A normal load reads from central memory the word addressed in the low byte of the load instruction and stores that word in the T register. If bit 11 is set in the load instruction then the word loaded in T register will be also loaded in the PDR register. Store instruction is the reverse operation of load. For normal STO the word in T register is stored in memory location addressed by low byte of instruction in IR register. If bit 11 is set the word in PDR is transferred to T register, then to memory as in a normal instruction. The load and store from PDR option is the path for loading and checking the control memory programs. The LCI loads the CTR low byte from the low byte of the IR register. The path is through the ALU which puts it in the same timing class as load-store instructions.

SINGLE-CYCLE MODE OF THE MBD

The run mode, or normal mode, of operation has been discussed. This is the mode after initializing, when the unit is operating as a DMA channel multiplexer.

The single-cycle mode is the mode used for initializing, loading the control memory, and checkout and testing of the MBD. The control is through the CSR register as discussed in Table I. The controlling source can be a program transfer from the PDP-11 computer or from the manual control console, which is a PDP-11 simulator. Operation of and from the control console will be discussed later.

In the single-cycle mode the MBD can be given commands through the function bits of the CSR register. Of a possible eight, five commands have been implemented as stated below:

1. Reset MBD - clears all request latches (CIL, PRL, CEL, STOP, CCL, PC), sets single-cycle mode, Ready set by strobe.
2. Execute IR - executes instruction in IR register, inhibits INCPC and LDOPC, End Instruction set Ready.
3. Load IR - load IR register with contents of PDR register, End Instruction set Ready.
4. <u>Single Instruction</u> - sequential central memory locations using PC counter, End Instruction set Ready.
5. <u>Channel Initialize</u> - bits (8,9,10) of CSR set CIL register to desired channel; bit 0 of CSR determines mode of operation, if equal to 1 set single-cycle mode, if equal to 0 set run mode. Initialize channel accepted by MBD set Ready.

0,6, <u>No Operation</u>, return Ready with strobe.
7.

Command 5 can be given with MBD in the run mode and is the path for the computer to make a channel request in the MBD. All other commands are to be used in the single-cycle mode and should be initially preceded by Reset MBD, which sets the single-cycle mode. Run mode is set with command 5 and a 0 in location "0."

The other controls in the CSR register are somewhat standard PDP-11 controls. Interrupt enable is the enable for the 25 interrupts of the MBD. The Ready bit is testable by the PDP-11 program and is the indication that MBD has completed the last command and is ready for the next command. The error interrupt does not identify the source of error, but testing bits 13 and 14 of the CSR register will identify branch or bus errors.

MBD MANUAL CONTROL CONSOLE

The control console, a separate unit, was designed for two purposes: to facilitate checkout and testing, and to operate the MBD as a stand-alone branch driver.

The operation of the console is that of a PDP-11 simulator and connects to the MBD through the UNIBUS connector. The console cannot be connected to the MBD at the same time as the PDP-11. The units have 16 switches for entering data or address into their respective registers in the unit. The register can be loaded either from the console switches or from the UNIBUS, depending on the type of transfer executed. Lights on the console display the contents of the data and address registers.

The units will be used primarily for program I/O transfers (DATO and DATI). The sequence for a DATO transfer is as follows:

1. Load data to be transferred into the console switches and place in data register by pressing load data switch.
2. Load address of device in address register by pressing load address switch.
3. Place the read/write switch to write position, then start the DATO operation by pressing the program transfer switch.

The DATI operation is as follows:

1. Load the address of device in address register.
2. Set read/write to read and start transfer switch. The data transferred from the device will appear in the register and data lights.

The console has an INIT switch which is the master clear for the system.

The operation of both types of priority transfer can be tested. For a bus request (BR) a light on the console indicates the request and pressing the bus grant (BG) switch allows the transfer to complete. The interrupt vector will appear in the data lamps.

The nonprocessor request (NPR) is basically the same as the BR. The NPR lamp indicates the request and pressing the nonprocess grant (NPG) switch completes the cycle. One difference is, if the transfer of data is from the console, then it must be loaded into the data register before the grant is initiated. If the transfer is to the console, the data will appear in the data lamps. Generally, transfers and interrupts will be initiated in the CAMAC hardware. A channel request can be initiated with a program transfer to the CSR register in the MBD from the control console.

MBD SOFTWARE

As of this writing, software does not exist for the MBD. What will be discussed are the initializing procedures for the MBD and CAMAC system and the type of software that is planned for development.

The first step in initializing the MBD is to load the control memory with the programs required to control the channels and the CAMAC system during the run mode of operation. The control memory is loaded and checked using the single-cycle mode of the MBD.

The following sequence is required:

1. Issue Function 1 - Reset MBD, which clears latches and puts in single-cycle mode.
2. Load PDR register with MBD instruction STO from PDR register.
3. Issue Function 3 - load IR from PDR register.
4. Load PDR register with instruction or data to be stored in MBD control memory.
5. Issue Function 2 - execute IR, which stores the word in PDR register in control memory location specified by low byte of IR register.

To verify the loading, the following sequence is required:

1. Load PDR register with MBD instruction. LOD from PDR register.
2. Issue Function 3 - load IR from PDR register.
3. Issue Function 2 - execute IR, which reads the control memory cell addressed by low byte of IR register and transfers that word to the PDR register.
4. Read PDR register and compare with word written into MBD central memory during load sequence.

After loading and verifying the loading of control memory, the system is put in the run mode and the file registers are loaded from a list in the PDP-11 memory. The program for initializing the file registers from a list in the PDP-11 is shown below. To initialize a channel the request is made via the CSR register and the ready bit is set when it is initialized.

Channel Initialize Program

```
CM Location  Instruction
     0       JVC
     1       MOV PDR, ILR, 0
     2       INM ILR, MAR, DCH RDR
     3       BCT DCH BSY, 3
     4       MOV MDR, CTR
     5       INM ILR, MAR, DCH RDR
     6       BCT DCH BSY, 6
     7       MOV MDR, DAR
     '         '
     '         ' (3 instructions per file register)
     '         '
   Last      MOV MDR, GPR, Exit 2
```

The above procedure is reentrant from all channels. The same procedure is used for reinitializing a channel during the run mode of operation. When a channel requires initializing, an end-of-block interrupt is set for that channel with a control command. When the interrupt is recognized, then the channel is initialized by a channel request from the PDP-11 as described above.

The same procedure for loading file registers from a channel can be used for loading sections of control memory during the run mode. That is, a program could be swapped in and out during run mode using a channel program.

Once the MBD has been initialized, then the CAMAC system must be initialized. This can be done using a low priority channel and using the list mode of operation. To get a channel to run from the computer can be accomplished by setting a testable bit in the CTR register during the initializing phase and testing that bit to determine the type of exit required for that channel. The general initialize routine for CAMAC depends on the system configuration. The type of commands required are: enable branch demand (BD) in each Type A, reset the inhibits, enable the L's in modules, and set up MASK registers.

Several levels of diagnostics are planned. The first level will be using the manual control console. These will vary from a few key-in instructions to a full-blown system loaded from CAMAC, and data-recording through CAMAC.

Diagnostics on the computer will verify various levels of system operation. The first level is that the computer functions properly; second, that MBD can be initialized; third, that MBD itself is functioning properly; fourth, that the branch is operating down to a Type A controller; and fifth, that the system from CAMAC modules to the computer are functioning properly.

One of the goals of standardization in the LAMPF data-acquisition system is software standards made possible through hardware standards. These standards will range from the disc operating system of the PDP-11 to the operation of the internal data links, and down to the driver (subprograms) routines used in the MBD. R. F. Thomas discusses in his LAMPF memo, "Possibilities for a Standard CAMAC Processor Language,"[7] a proposal for a CAMAC language and how the MBD would be programmed to serve as its virtual controller.

PROPOSED USES OF MBD AND DESIGN OPTIONS

The MBD was designed primarily to meet the requirements of the LAMPF data-acquisition system. The LAMPF terminal computer will initially use a standard MBD, but one of the design options is to expand the MBD to 16 or 24 DMA channels, which will be required of the terminal computer when LAMPF is at full capacity.

One possible major requirement is the need of a manual, flexible branch driver. With a manual console and a convenient means of loading programs, the MBD becomes a very flexible, powerful manual controller. The loading can be accomplished with peripheral interfaces in CAMAC and a bootstrap loader in the MBD. With a recording capability, or a link to recording capability (terminal computer), the MBD could control a small, simple experiment. One design option, the expansion of memory in the MBD, makes this very feasible. With the present memory design the MBD can be expanded to 512 words of memory and with the new 256-bit memory chips, the unit has space for over 1000 words of memory, if that were ever required.

Another possible use of the MBD is as a remote branch driver. The computer bus is limited to 50 ft or less, and the present branch designs are also limited to 50 ft or less. Each of these interfaces has timing limitation, large numbers of wires, and major redesign effort to go any distance. With two MBD's and using the high-speed data-link modules (developed at LAMPF for computer-to-computer transfers through CAMAC), then a branch may be operated at a remote distance of a few thousand feet without any penalty of operations performance and major redesigns of the system.

A very minor modification will convert the MBD from binary arithmetic to 2's complement arithmetic if this is required. One design option that is open, but will probably never be used, is the addition of a second branch to the MBD. The most probable solution would be to add a second MBD and modify the interrupt assignments of the second unit.

MBD HARDWARE AND FABRICATION

Figure 4 is a photograph of the MBD and control console. The system was designed and fabricated using the computer automated system hardware (CASH) developed by Standard Logic, Inc. The CASH hardware is designed for use with computer-aided design (CAD) technology and is supported by a complete line of systems software, computerized documentation, and semiautomatic wiring.

The MBD was designed in the TD-36 drawer, which has 3½ in. panel height, slides, and two cooling fans. The drawer has two separate 18-card planes. The planes are wired separately and interconnected through cables and two connector cards. The top plane has 14 CASH cards and houses the microprocessor. The bottom plane has 13 CASH cards and houses the computer interface and the branch driver. All cabling is to the bottom plane. The two CAMAC port connectors are on the front panel and the computer UNIBUS connector is on the rear panel. The control console plugs into the computer connector.

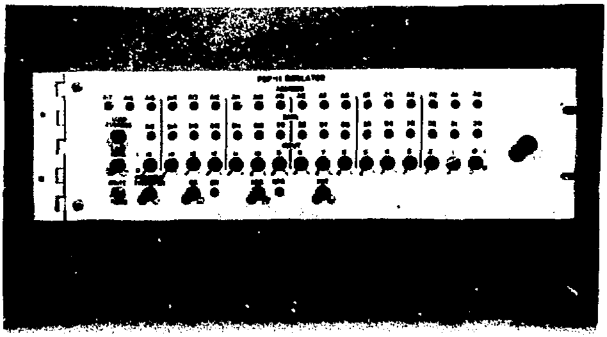


Fig. 4

The eight DMA channel, 256 words of memory model of the MBD has 500 integrated circuits and requires a separate 5 Vdc, 25 A power supply.

The CASH system has a variety of cards using 14-, 16-, and 24-pin sockets for integrated circuits, and the wiring is a two-level wirewrap using No. 30 wire. The unit has 4600 wires and was wired in 40 hours using the machine-aided wiring system.

The manual console uses the CASH hardware and is housed in a DC-01 vertical panel. The unit has three CASH cards and a 5 Vdc, 5 A power supply. One cable connects the unit to the MBD. The front panel has the control switches and the address and data lights.

ACKNOWLEDGMENTS

The authors are indebted to Richard F. Thomas, Jr., Los Alamos Scientific Laboratory, by whom the MBD systems design and operation was greatly influenced, and to James A. Buchanan and Hugh V. Jones, Rice University, for the initial development of the MBD.

REFERENCES

1. "LAMPF Data-Acquisition System," LA-4504-MS, Los Alamos Scientific Laboratory, Los Alamos, N. M. 87544, Aug. 1970.

2. J. A. Buchanan and H. V. Jones, "CAMAC Multi-Microprogrammed IO Processor," IEEE Trans. on Nucl. Sci., Vol. NS-19, No. 1, Feb. 1972, p. 682.

3. "A Modular Instrumentation System for Data Handling," EURATOM 4100e. Available from L. Costrell, Natl. Bureau of Standards, Washington, D. C. 20234.

4. "Organization of Multicrate Systems," EURATOM 4600e, ibid.

5. PDP-11 UNIBUS Interface Manual, DEC-11-H1AA-D.

6. PDP-11 Handbook 112X01269, AJOF1150.

7. "Possibilities for a Standard CAMAC Processor Language," Notes by R. F. Thomas, Jr., MP-1, LASL, Los Alamos, N. M. 87544, Oct. 20, 1971.

EE/pu:353(100)