G. Bell

PDP-X Technical Memorandum # 30

Title: Control Memory Format (Preliminary)

Author(s): H. Burkhardt

Index Keys: Control Memory
Processor

Distribution
Keys: A

Obsolete: None

Revision: None

Date: 16 November 1967

The Model II processor, currently being implemented, is to be controlled by a read-only memory (ROM) containing a fixed program. Many of the reasons for using this approach are contained elsewhere and will not be discussed in this document. The actual details of implementation will not be discussed here either.

PDP-X architecture (discussed in PDP-X Technical Memorandum #16) calls for several processors connected to the same memory and IO systems. One of these processors (called the arithmetic processor) executes instructions as described in PDP-X Technical Memorandum #29. Other processors (called IO processors or selector channels) execute special instructions to control high-speed peripheral devices. Additional processors may execute floating-point instructions or special customer-specified instructions.

From the point of view of manufacturing, check-out, and field service, it is highly desirable that there be as much commonality as possible between these various processors even though the functions that they perform are often dissimilar. This, in fact, is a major feature of PDP-X architecture.

To accomplish this goal, it was realized that a processor may be logically divided into several sections:

a. *Registers, data paths, arithmetic and logical operations

b. Memory subsystem control

c. IO subsystem control

d. *Internal timing generation

e. Sequence and decision control (Main Control for a,b,c,d)

* See PDP-X Technical Memorandum #24.

If the registers, data paths, etc. (a) are general enough, all of these sections may be common to all processors. The only difference between the various types of processors described above will be in the Main Control (e). A read-only memory is the most suitable form for this control since:

a. Many of the operations performed by both the arithmetic processor and the IO processors (selector channels) are repetitive or iterative in nature.

b. The extended instructions found in PDP-X Model II are fairly complicated compared to those found in current small computers.

c. The read-only memory provides a means of organizing the many diverse functions performed by the control.

d. The actual read-only memory (not counting address drivers, sense amplifiers, etc) is a small portion of the entire processor and since it is the only non-identical part of the several types of processors, the goal of commonality is achieved. (Note that even though the read-only memories for the different types of processors are different, the actual difference lies only in the patterns wired into the memory. Mechanically and electrically, they are identical.)

To understand the operation of the read-only memory, it is helpful to note its relationship to conventional controls. Conventional controls may be characterized by sequence triggers, and by the control lines activated by the sequence triggers as a function of the operation to be performed and data conditions. Each cycle that the CPU may take represents a state of the CPU as defined by the control circuitry. Each state, in turn, specifies which control lines are to be activated during that cycle and which state is to follow next. The defined state will cause the next sequence trigger to be set in the following cycle. In some cases, the next state may be contingent upon a branch condition in which one of two or more sequence triggers must be selected.

In ROM controlled CPU's, the sequence triggers are replaced by micro-instructions. Each consists of a predetermined bit pattern and represents a state of the CPU. The addressed ROM word controls the CPU during the particular machine cycle it is in use. When decoded, the ROM word defines all control lines that are to be activated during the machine cycle. Also contained in the ROM word is the address of the next ROM word to be used. If the address of the next ROM word is dependent on data conditions (for example, branch if step counter is zero), a base address and the conditions to be tested (branch tests) are specified in the ROM word.

Each ROM word consists of 34 control fields. The number of bits within a field determines the number of unique control signals (micro-orders) available within that field. (In a 4-bit field, for example, 16 distinct micro-orders can be defined, only one of which can be activated at any one time.) The micro-orders are grouped functionally within the fields according to two rules.

1. All micro-orders grouped in a field must be mutually exclusive since only one micro-order within that field may be specified at a time.

2. Micro-orders that are functionally similar (such as micor-orders that control the adder function) are grouped in one field for ease of decoding.

ROM Addressing and Branching

Micro-instructions are not executed from sequential read-only storage addresses; rather, each micro-instruction explicitly states its possible successors. A given micro-instruction may have as many as 512 different successors, although most do not have more than one or two. The choice of a successor is made on the basis of branching tests specified by the micro-instructions. This powerful branching ability of the microprogram contributes significantly to reducing the number of words of read-only storage required.

The normal microprogram sequencing can be overridden by the detection of an exceptional condition indicative of a program or machine error. References to non-existent memory or divide overflow are examples of conditions detected in this manner. If such an event occurs, the normal address of the successor micro-instruction derived as described above is ignored. Instead, a special wired-in fixed address, characteristic of the type of event, is forced into the read-only memory address to determine the successor micro-instruction. This technique is referred to as a microprogram trap, and is analogous to a program trap in machine-language programming, except that no record is kept of the address that would normally have been used.

Read-only memory is technically and economically advantageous in implementing a large instruction repertoire; the control function depends upon a relatively small amount of active electronics, and control signals are generated with a minimum of dynamic control logic. Benefits accrue in lower component counts (and cost), higher reliability, and simpler control function fault isolation. In addition, ROM control provides the potential for modification or creation of instructions through modification of the microprogram contained in the read-only store. The creation of micro-instructions tailored to specific applications is technically feasible and readily implemented.

The following pages define the control fields of the read-only memory word. The original specifications call for a 300 word, 72-bit read-only memory. Several bits are not defined as of the time of this writing, but it is assumed that meanings will be assigned to them as the design progresses.

Group	Field	Number of bits	Description
Control	CN_{0-7}	8	CoNstant - Provides a byte of constants for the Adder Bus or 2 Fast Memory addresses (4 bits); one for reading, one for writing.
	BF_{0-8}	9	Branch Field - Contains Address of next Control Memory word to be executed. This field is OR'ed with branch conditions selected by the Branch Conditions field.
	BC_{0-4}	5	Branch Conditions - Provides for the selection of 32 branch combinations. A configuration of these bits produce a nine bit number (that depends upon the state of the central processor) that is OR'ed with the Branch Field to locate the next control word to be executed.
	TRPEN.	1	TRaP ENable - If this bit is set and a trap condition exists in the Status Register, a ROM trap is initiated to location 0. Bits 0, 1 of the Status Register are then cleared. If a level change occurs, bit 1 of Priority Active is set.
	INTEN	1	INTerrupt ENable - If this bit is set and either an interrupt is pending, or a console interrupt is pending, initiate a ROM trap to:

Group	Field	Number of bits	Description
Control (Cont.)			location 1 for Interrupt location 2 for console (if no interrupts pending) location 3 for Interrupt (Note: if both TRPEN and INTEN are set, interrupts have the highest priority.) When the interrupt trap is taken, the Priority Active bit corresponding to the interrupt level is set.
	Total	24	

Group	Field	Number of bits	Description
Memory Control	MC_{0-1}	2	Memory Control - Provides mode control for main memory. These bits are decoded as follows: 00 Do Nothing 01 Read; Start Memory using C(MI) as address. Set FASTS or STATS if location is in fast memory or in the Status Register. 10 Write; If memory is already running, set to WRITE mode; otherwise start memory as above but in WRITE mode. 11 Pause; If memory is already running set to PAUSE mode; otherwise start memory as above but in PAUSE mode.
	AW	1	Address acknowledge Wait - If main memory is running, stop the control memory until Address Acknowledge is received. If FASTS or STATS is set, continue control memory.
	RW	1	Read restart Wait - If main memory is running, stop the control memory until Read Restart is received. If FASTS or STATS is set, continue control memory.

Group	Field	Number of bits	Description
Memory Control (Cont.)	MR	1	Memory Release - If memory is in Read, Release the memory system. If memory is in Write or Pause, release the memory and restart it.
	Total	5	

Group	Field	Number of bits	Description
Result Bus	LMI	1	Load MI - Load the Result Bus into the MI.
	LAR	1	Load AR - Load the Result Bus into the AR.
	RL_{0-2}	3	Register Load - The three bits are decoded as follows: 000 Do Nothing 001 LSC, Load the Step Counter from the Result Bus. 010 LSR, Load the Status Register from the Result Bus. (bits 10-12 are not changed. 011 LCI, Load the Console Indicators from the Result Bus. 100 LFR, Load the Fast memory at the mapped location specified by the R-bits of the Instruction Register (from the Result Bus). If R = 0, load the Status Register. 101 LFRO, Load the Fast memory at the mapped location specified by the R-bits of the Instruction Register OR'ed with 001 (from the Result Bus).

Group	Field	Number of bits	Description
Result Bus (Cont.)	RL_{0-2} (Cont.)		110 LPG, Load the Paging Registers selected by AR bits 11-15 from the Result Bus. Even paging registers are loaded from the Right Half (bits 8-15) and odd paging registers from the Left Half (bits 0-7), i.e., AR = 3 indicates loading Paging Registers 7,6 from the Result Bus.
			111 LFCN, Load the Fast memory at the location specified by the Left Hand four bits of the constant field (CNO 1-3). The left most bit (CNO) is OR'ed with RG to form the left three bits of the Fast memory address. The right hand three bits of the Fast memory address corresponds to CNI_{1-3}.
	Total	5	

Group	Field	Number of bits	Description
Data Control	CI_{0-1}	2	Carry Insert control - A carry is inserted as follows: 00 No Insert 01 Insert a Carry (+1) 10 Insert MI_{16} 11
	SFT_{0-1}	2	ShiFT control - The two bits are interpreted as follows: 00 No Shift 01 Shift Left 10 Shift Right 11 Swap Bytes
	SEL_{0-1}	2	SELection control - The two bits are interpreted as follows: 00 Normal, Enable Adder outputs 01 AND 10 Multiply Step 11 Divide Step

Group	Field	Number of bits	Description
Data Control (Cont.)	EF_{0-1}	2	End Effects - Control end effects during a shift operation as follows: 00 Multiply Shift; carry into bit 0, bit 15 into AR_{16} (only for right shift) 01 Rotate with CC0 10 Decode from AR_{6-7}: 00 Arithmetic Shift 01 Rotate with CC0 10 Rotate 11 Logical Shift 11 Divide Shift Left
	Total	8	

Group	Field	Number of bits	Description
A Bus, FM address (Read), Data Read	ARR	1	AR Right - Connect the Right Half (bits 8-15) of the AR to the A bus.
	ARL	1	AR Left - Connect the Left Half (Bits 0-7) of the AR to the A bus.
	FMA	1	Fast Memory A bus - Connect Fast Memory to the A bus. If the address source is the R-bits of the Instruction Registers (FMR) and R = 0, connect STATUS to the A bus.
	FMX	1	Fast Memory indeX - If X $\neq$ 0, connect Fast Memory to the A bus using the X-bits of the Instruction Register as address. If X = 0, do nothing.
	FMR	1	Fast Memory R - Use the R-bits of the Instruction Register as Fast Memory address. If R = 0, connect STATUS. Actual data connection is established by FMA, FMC, FBR, FBL.
	FMOR	1	Fast Memory OR - When FMR is a one, this bit will cause the Fast Memory address to be the R-bits of the Instruction Register OR'ed with 001. When FMR is a zero, this bit should be zero.
	FMCN	1	Fast Memory CoNstant - Use the right hand 4 bits of the constant field (CN_{4-7}) as fast memory address. CN_4 is OR'ed with RG2 to form fast memory address bit 2.

Group	Field	Number of bits	Description
B Bus Input	DBR	1	Data B bus Right - Connect the right half (bits 8-15) of memory data to the B Bus input. Actual result depends upon previous memory start (the DAT bit must be on). Data may come from MEM FM STATUS
	DBL	1	Data B bus Left - Connect the left half (bits 0-7) of memory data to the B Bus as above. (The DAT bit must be on.)
	FBR	1	Fast memory B bus Right - Connect the right half (bits 8-15) of the fast memory to the B Bus. Fast Memory address is determined by the FMR, FMOR, FMCN fields. (See above.)
	FBL	1	Fast memory B bus Left - Connect the left half (bits 0-7) of the fast memory to the B Bus. Fast memory address is determined by the FMR, FMOR, FMCN fields. (See above.)
	IB_{0-2}	3	Input to B bus - The three bits are decoded as follows: 000 Nothing 001 CON. Connect the constant field (CN_{0-7}) to the right half of the B bus (bits 8-15). 010 SR. Connect the Status Register to the B bus.

Group	Field	Number of bits	Description		
B Bus Input (Cont.)	IB_{0-2} (Cont.)		011	CS.	Connect the Console Switches to the B bus.
			100	DATC.	Connect the complement of memory data to the B bus. Actual result depends upon previous memory start. The DAT bit must be on.
			101	SE.	Connect sign-extended memor data to the B bus. Source depends upon the previous memory start. If the X-bits of the IR are zero, connect only DBR. The DAT bit must be on.
			110	FMC.	Connect the Fast Memory complement to the B bus. Fast Memory address is determined by the FMR, FMOR, FMCN fields. (See above.)
			111	ARC.	Connect the complement of the AR to the A bus.
		Total 7			

Group	Field	Number of bits	Description
IO	IOEX	2	IO EXecute - The two bits are decoded as follows: 00 Nothing 01 IODO; Execute the IO command in the IOCT register when RTN has dropped. (i.e., drive Command Out 5-7 from IOCT 13-15.) 10 IOCH; IO Channel operation: Execute the IO command in the IOCT register when RTN has dropped. Command Out driven as follows: CO5 (IOCT13) + (IOCT11) CO6 (IOCT14) + (MI16) CO7 (IOCT15) (Note: Both of these stop the control memory until RTN is raised. If time-out occurs first, set bit 6 of the PSW and continue. If the operation is an output, drop SYNC when RTN is received and continue; otherwise continue.)

Group	Field	Number of bits	Description
IO (Cont.)	IOEX (Cont.)		11 IOL; If IOCT indicates an input operation; Connect the IO Bus receivers to the left half of the B Bus (bits 0-7). After the input gates are latched, drop SYNC.
	$IOCL_{0-1}$	2	IO Control Load - The two bits are decoded as follows: 00 Do Nothing 01 Load Constant bits 5-7 into IOCT bits 13-15. IOCT bits 9-12 remain undisturbed. 10 Load the IO Bus register (IO Bus driving register) from the Result Bus (bits 8-15). 11 Load Constant bits 1-7 into IOCT bits 9-15.
	Total	4	

Group	Field	Number of bits	Description
MISC	FS_{0-1}	2	Flag Select - These bits select one of four flags to be operated upon by the FC field. (See below.) Flag selection occurs as follows: 00 QSIGN 01 RSIGN 10 RUNFLG 11 KEYSYNC
	FC_{0-1}	2	Flag Control - The flag selected by the FS field is operated upon as follows: 00 unchanged 01 clear selected flag 10 set selected flag (KEYSYNC may not be set) 11 complement selected flag (RUNFLG and KEYSYNC may not be complemented) if FS = 10 (RUNFLG) and FC = 10 (Set), the following operation is performed: IRLD, Load the Instruction Registers with Memory Data. (This occurs before the next cycle begins). The actual Source of the memory data depends upon the previous memory start. Data may come

Group	Field	Number of bits	Description
MISC (Cont.)	FS_{0-1} (Cont.)		from: Main Memory Fast Memory
	MSC_{0-3}	4	MiScellaneous Control functions: - These four bits are decoded as follows: 0000 Nothing 0001 Decrement the Step Counter by one (this occurs at the beginning of the processor cycle). 0010 Disable Adder output to Selector and connect MI to shifter. 0011 Jam Carry Out into MI_{16}. 0100 Push Down List Overflow: carryout is OR'ed into SR bit 3. 0101 Divide overflow: carryout is OR'ed into SR bit 2. 0110 Set Condition Code bits 1,2 to reflect the state of the Result Bus. 0111 Set Condition Code bits 1,2 for compare instruction. 1000 Set Condition Code bits 0,1,2 and SR2 for add or sub instructions.

Group	Field	Number of bits	Description	
MISC (Cont.)	MSC_{0-3} (Cont.)		1001	Set Condition Code bits 1,2 and SR2 to reflect result and shift error (error condition is sensed only during arithmetic left shift.)
			1010	RESET. Do IO Reset on IO Bus. Clear PI ACT, PI INH, PI EXR, PI INR, set PI ACT 1 to a 1 if Protection System, set RG to highest PI ACT.
			1011	Set priority inhibit (PI INH) bit corresponding to AR12-15 if this is higher than the highest priority (PI ACT) bit.
			1100	Clear the highest priority inhibit bit (PI INH) if it is higher than the highest priority active bit.
			1101	Set the Priority Interrupt Internal Request (PI INR) indicator corresponding to AR13-15.
			1110	Change Map (RG) to that indicated by the highest priorit active (PI ACT) bit.

Group	Field	Number of bits	Description	
MISC (Cont.)	MSC_{0-3} (Cont.)		1111	Dismis: Clear the Highest Priority Active (PI ACT) bit and change the Map (RG) to the value of the new highest bit.
	Total	8		

Appendix 1

Fast Memory Address

The Read-only Memory generates fast memory addresses in one of five ways:

a. Constant address.

b. R bits of Instruction Register.

c. R bits of Instruction Register OR'ed with 001.

d. X bits of Instruction Register.

e. A memory reference is made with the address (in the MI) pointing to fast memory.

Logically, 15 bits of address are produced:

a. 00000000(RG0)(RG1)(RG2,C)0CCC
(The C's are Constant bits)

b. 00000000(RG0)(RG1)(RG2)0RRR
(The R's are the R-bits)

c. 00000000(RG0)(RG1)(RG2)0RR1

d. 00000000(RG0)(RG1)(RG2)00XX
(The X's are the X-bits)

e. The 15-bit address in the MI during a memory start is mapped. The mapping algorithm is:

If Address bits 9-11 = RG, replace them with 000.

If Address bits 9-11 = 000, replace them with RG.

Otherwise, do nothing to the address.

The map produces a 15-bit address which is sent to either the Main Memory or to Fast Memory. When sent to Fast Memory, this address has the form:

00000000FFF0FFF or FFFFFF

Where the F's address one of the 64 Fast Memory registers.

Appendix II

Control Memory Addressing

Every control memory micro-instruction contains a 9-bit Branch Field (BF) and a 5-bit Branch Condition field (BC). These two fields are used to form the address of the next micro-instruction to be read from read-only memory and executed.

The Branch Condition field selects 1 of 32 9-bit numbers to be OR'ed (INCLUSIVE) with the Branch Field to form the next control memory address. For example, when BC = 00000, the 9-bit number OR'ed with BF is 000000000; thus, this represents an unconditional branch to the control memory address specified by BF. If BC = 4, the 9-bit number OR'ed with BF is either 000010000 or 000000000 depending upon the state of QSIGN (a flag flip-flop internal to the processor). If BF were XXXX0XXXX where the X's indicate "don't cares", a two-way branch on QSIGN is effected. The locations branched to are either XXXX0XXXX or XXXX1XXXX depending upon the state of QSIGN. If BF had contained XXXX1XXXX, control would pass to location XXXX1XXXX regardless of the state of QSIGN. This is useful in cases where the Branch Condition field provides for a wider branch than is actually needed. For example, when BC = 10, the number OR'ed into BF is

00000 IRR0 IRR1 IRR2 0

If BF contained

XXXXX 0 0 0 X

an 8-way branch would be effected. If, however, the ROM micro-program wished merely to test whether or not the Instruction Register R-bits were odd, BF could be set to

XXXXX 1 1 0 X

and a 2-way branch would be effected.

In addition to the branching described above, a ROM trap may be initiated if the appropriate conditions are present in the processor and either TRPEN or INTEN are set. The traps force the read-only memory micro-program to a fixed location. Normal branching sequence occurs from that point on.