digital

hardware

FPP8-A
user's manual

FPP8-A
user's manual

EK-FPP8A-OP-001

digital equipment corporation • maynard, massachusetts

First Edition, December 1976

Printed in U.S.A.

This document was set on DIGITAL's DECset-8000 computerized typesetting system.

CONTENTS

TABLES

CHAPTER 1
INTRODUCTION

1.1 GENERAL

The FPP8-A is a processor that performs arithmetic calculations with floating-point numbers. It is compatible with the FPP12-A instruction set and will run OS8 FORTRAN IV without program modification; with minor program changes, the FPP8-A will run FORTRAN IV at higher speeds.

The FPP8-A consists of two interconnected, hex-size, printed-circuit modules that plug into the Omnibus of a PDP-8/A computer (hereafter, the terms "FPP" and "PDP-8" will be used instead of the full designations). There are no connections from the FPP to external devices, and the FPP derives all of its power from the Omnibus. When the PDP-8 is turned on, the FPP remains inactive until started by IOT instructions issued by the Central Processing Unit (CPU). Once started, the FPP retrieves instructions and operands from the PDP-8 memory by data breaks; many data manipulations and arithmetic calculations are then carried out independently of the CPU and at a higher speed than is possible with CPU timing. The FPP continues to run until halted by an IOT instruction or an FPP instruction, until it encounters numbers that are too large or too small to handle, or until the PDP-8 is halted.

1.2 FPP/CPU INTERACTION

The FPP and the CPU operate in parallel in the data break system. Two modes of parallel operation are possible. In the Interleaved mode, which is automatically entered when power is turned on, the FPP can use a maximum of every other memory cycle; this permits the PDP-8 to run at no less than 50 percent of normal speed. In the Lockout mode, which is selected by an IOT instruction, the FPP can use consecutive memory cycles, as long as no interrupt requests are made by peripheral devices. If such a request is made, the FPP automatically goes to the Interleaved mode; when the interrupt request has been serviced, the FPP returns to the Lockout mode.

1.3 FPP CALCULATING MODES

The FPP can perform calculations in any one of three modes, namely, Floating Point (FP), Double Precision (DP), and Extended Precision (EP). The format of the data used in each of the calculating modes is also unique; both the modes and their respective formats are listed and explained in Table 1-1.

Table 1-1 FPP8-A Calculating Modes and Data Formats

Calculating Mode	Description
Floating Point (FP)	Floating-point calculations are carried out with numbers having a 12-bit, signed, 2's-complement exponent and a 24-bit, signed, 2's-complement fraction. Fraction calculations are made on a 36-bit word, and the result is rounded off to 24 bits at the end of the arithmetic operation. The FPP automatically enters this mode when power is turned on, when either the CAF or FPICL IOT instruction is issued, or when the INIT key is pushed.

Table 1-1 FPP8-A Calculating Modes and Data Formats (Cont)

Calculating Mode	Description
Floating Point (FP) (Cont)	Data used in FP calculations is stored in the PDP-8 memory in this way: The exponent is stored in the memory location pointed to by the FPP instruction; the most-significant word (MSW) of the fraction is stored in the memory location immediately following the exponent; the least-significant word (LSW) of the fraction is stored in the memory location immediately following the MSW.
Double Precision (DP)	Fixed-point calculations are carried out with numbers having a 24-bit, signed, 2's-complement fraction. This mode is the same as the FP mode, except that the exponent is ignored and treated as if it were zero. Data used in DP calculations is stored in the PDP-8 memory in one of two ways, depending on the addressing mode used. For base page (12-bit direct) addressing the MSW of the fraction is stored in the memory location immediately following the location pointed to by the instruction; the LSW is stored in the next consecutive memory location. For other modes of addressing, the MSW is stored in the memory location pointed to by the instruction; the LSW is stored in the next consecutive memory location.
Extended Precision (EP)	Floating-point calculations are carried out with numbers having a 12-bit, signed, 2's-complement exponent and a 60-bit, signed, 2's-complement fraction. Calculations are carried to 60 bits with no round-off. Data used in EP calculations is stored similarly to that of the FP mode; however, three additional locations are needed, with the LSW being stored in the fifth location following the exponent.

1.4 FLOATING POINT CONCEPTS

Various manipulations relating to floating-point arithmetic can be performed by the FPP logic. These are briefly described to familiarize the reader with basic concepts. The following descriptions are based on the FP calculating mode.

1.4.1 Float

When a number is floated, it is converted from its integer form to a fractional floating-point format. To float an integer, one places the number of significant bits of the calculating mode in the exponent (27_8 significant bits plus the sign bit), moves the binary point to reflect the value of the exponent, and then shifts the fraction left until the leading zeros are eliminated, decrementing the exponent with each shift. For example: To float the integer 12_8, write down the whole number

1010.0;

then, write down 27_8 as the exponent

000 000 010 111;

move the binary point to reflect the exponent

0.00 000 000 000 000 000 001 010;

now shift the fraction left until the leading zeros are eliminated, decrementing the exponent with each shift

000 000 000 100 0.10 100 000 000 000 000 000 000.

The floating-point number is $+.101 \times 2^4$, the equivalent of 1010 (12_8).

1.4.2 Fix (or Integerize)

Fixing a number is the reverse process of floating. To fix a number one changes the exponent to 27_8 and then shifts the fraction right a number of times equal to the difference between 27_8 and the original exponent. Thus, to fix the number arrived at by the previous float, which was (in octal notation)

0004 2400 0000;

make the exponent 27_8 and shift the fraction right 23_8 places. The result is

000 000 010 111 0.00 000 000 000 000 000 001 010.

To obtain the integer, move the binary point to reflect the exponent, thereby obtaining

1010.0.

If the exponent is greater than 27_8, the floating-point number is too large to fix; the FPP uses the JAL instruction to check the possibility of fixing fractions.

1.4.3 Normalize

A non-zero floating-point number is normalized by shifting the fraction to the left until non-significant leading zeros are eliminated; each shift is accompanied by a subtraction from the exponent. The number is normalized when the first two bits are different (i.e., 0.1 or 1.0) or when only the first two bits of the fraction are ones (i.e., the number is 6000_8). In DP mode, numbers are not normalized.

1.4.4 Align

Certain operations, such as addition and subtraction, require that numbers be aligned. For example, if two numbers are to be added, their exponents must be equal; if the exponents differ, the numbers must be aligned. That is, the exponent of the smaller number must be increased until it equals that of the larger number; each increase of the exponent must be accompanied by a right-shift of the smaller number's fraction.

1.5 REFERENCES

Normalization and alignment are discussed more fully and details concerning floating-point arithmetic are presented in the publication *8/A Series Minicomputer Handbook*, 1976-1977, available from DIGITAL. Other publications that contain instructive information about the FPP and its relationship to the PDP-8 are:

a. FPP8-A Diagnostic, MAINDEC-08-DJFPA
b. FPP8-A Instruction Test and Data Exerciser, MAINDEC-08-DJFPB
c. PDP-8/E, 8/F, and 8/M Maintenance Manual
d. PDP-8/A Miniprocessor User's Manual.

CHAPTER 2
INSTRUCTION SET AND ADDRESSING

2.1 FPP REGISTERS

The FPP logic includes many data registers. Some registers are separate entities, while others occupy space in two high-speed, multiport RAMs that are part of the FPP's Data Path logic. Moreover, some registers are involved only with actual data calculations, while others are also instrumental in setting up and maintaining communication between the FPP and the PDP-8 CPU. These latter registers, which are mentioned frequently in the IOT and FPP instruction lists, are listed and described in Table 2-1.

Table 2-1 FPP Registers

<table>
<tr><th>Register</th><th>Function</th></tr>
<tr><td>APTP (Active Parameter Table Pointer)</td><td>The 15-bit APTP register in the Data Path logic is loaded with the PDP-8 memory address of the Active Parameter Table (APT) by IOT instructions (FPCOM and FPST). The APT consists of a block of 2, 8, or 11 consecutive memory locations that contain the following information (if the FPCOM instruction has directed a fast start (FS), only locations 1 and 2 are loaded into the FPP hardware; if a normal start is programmed, the information in either the first 8 locations (DP or FP mode) or all 11 locations (EP mode) are obtained by the FPP).

<table>
<tr><th>Sequence of Memory Locations</th><th>Contents of Location</th></tr>
<tr><td>1</td><td>Field Bits
Bits 0-2 = operand address field
Bits 3-5 = Base register field
Bits 6-8 = Index register address field
Bits 9-11 = FPC field</td></tr>
<tr><td>2</td><td>FPC (Floating Program Counter) 12 low-order bits</td></tr>
<tr><td>3</td><td>Index register address – 12 low-order bits</td></tr>
<tr><td>4</td><td>Base register – 12 low-order bits</td></tr>
</table>
</td></tr>
</table>

Table 2-1 FPP Registers (Cont)

Register	Function	
APTP (Cont)	**Sequence of Memory Locations**	**Contents of Location**
	5	Operand address – 12 low-order bits (this word is ignored upon FPP start-up, but is filled upon FPP exit).
	6	FAC exponent
	7	FAC bits 0-11
	8	FAC bits 12-23
	9	FAC bits 24-35*
	10	FAC bits 36-47*
	11	FAC bits 48-59*

*EP mode only

FPC (Floating Program Counter)

The 15-bit FPC register in the Data Path logic keeps track of the memory location of the FPP instructions. The register is initially loaded from the APT with the address of the MSW of the first instruction to be fetched. Each time an instruction word is fetched, the contents of the FPC are incremented (strictly speaking, only the MSW of a 24-bit instruction is fetched; the LSW is picked up by a memory-read operation).

Command

The 12-bit Command register in the Control logic is loaded from the PDP-8 Accumulator (AC) register by the FPCOM instruction. The register holds the following information.

Bit Position	Logic Level (1=high)	Information
0	0	Select FP mode
	1	Select DP mode
1	0	If exponent underflow occurs, make result of calculation = 0 and continue.
	1	If exponent underflow occurs, exit.
2	0	Normal addressing

Table 2-1 FPP Registers (Cont)

Register	Function		
Command (Cont)	**Bit Position**	**Logic Level (1=high)**	**Information**
		1	Forbid access to 4K memory fields other than the one occupied by the last location of the APT. If bits (4:7) =1111 (FS), the field bits will remain equal to the APT field bits when the FPC was obtained. Otherwise, the field bits will remain equal to the APT field when FAC bits 12-23 were obtained. In actual practice the APT is located where it does not cross a field boundary; hence, setting bit 2 forces all FPP operands and instructions to be in the same field as the APT. Attempts to cross field boundaries will then produce wrap-around within the APT field.
	3	0	Disable FPP interrupt.
		1	Enable FPP interrupt.
	(4:7)	=1111	Obtain and restore only the FPC on entry and exit. All other FPP registers retain their previous values. The 9 most-significant field bits of the APT are ignored on entry and cleared upon exit. This mode of operation provides an extremely fast (2 cycle) start and exit, but sacrifices generality.
		≠1111	Obtain entire APT upon startup except for operand address. Restore entire APT upon exit, including operand address.
	8	0	Interleaved operation
		1	Lockout operation
	(9:11)		Most-significant 3 bits of APT pointer.

NOTE

Upon application of power, the APT pointer is undefined.

Table 2-1 FPP Registers (Cont)

<table>
<tr><th>Register</th><th colspan="3">Function</th></tr>
<tr><td>Status</td><td colspan="3">The 12-bit Status register in the Control logic monitors some significant aspects of FPP operation; the contents of the register, which can be transferred to the PDP-8 AC register by the FPRST or FPIST instruction, represent the following information.</td></tr>
<tr><td></td><td colspan="3"><table>
<tr><th>Bit Position</th><th>Logic Level (1=high)</th><th>Information</th></tr>
<tr><td>0</td><td>0</td><td>FP or EP mode</td></tr>
<tr><td></td><td>1</td><td>DP mode</td></tr>
<tr><td>1</td><td>1</td><td>Trap instruction caused exit</td></tr>
<tr><td>2</td><td>1</td><td>FPHLT instruction caused exit</td></tr>
<tr><td>3</td><td>1</td><td>Attempted divide by zero caused exit. FAC not altered</td></tr>
<tr><td>4</td><td>1</td><td>Fraction overflow in DP mode caused exit</td></tr>
<tr><td>5</td><td>1</td><td>Exponent overflow caused exit</td></tr>
<tr><td>6</td><td>1</td><td>Exponent underflow has occurred. Exit on underflow is optional</td></tr>
<tr><td>7</td><td>1</td><td>FADDM or FMULM instruction</td></tr>
<tr><td>8</td><td>0</td><td>Interleaved operation</td></tr>
<tr><td></td><td>1</td><td>Lockout operation</td></tr>
<tr><td>9</td><td>1</td><td>EP mode</td></tr>
<tr><td>10</td><td>1</td><td>FPP is currently paused.</td></tr>
<tr><td>11</td><td>1</td><td>FPP is currently in run state</td></tr>
</table></td></tr>
<tr><td>Field</td><td colspan="3">This is a 15-bit register located in the Data Path logic that is used only during initialization for temporary storage of the APTP field address. Do not confuse this register with the field bits contained in location 1 of the APT.</td></tr>
</table>

Table 2-1 FPP Registers (Cont)

Register	Function
FAC (Floating Accumulator)	The FAC register has a function similar to the PDP-8 AC register; it can be loaded, stored, and tested, and arithmetic can be performed on its contents (FPP calculations take place in a scratchpad area and the results are stored in the FAC). The FAC occupies space in one of the Data Path random access memories (RAMs) and can comprise 2 (DP mode), 3 (FP mode), or 6 (EP mode) data locations.
OPADD (Operand Address)	The 15-bit OPADD register in the Data Path logic holds the PDP-8 address of the instruction operand. At startup, the register is loaded with the contents of the FPC; thereafter, it is loaded during all data reference and trap instructions. At the conclusion of address decoding of a data reference instruction, OPADD contains the address of bits 12–23 of the operand fraction.
BR (Base)	The 15-bit BR register in the Data Path logic is loaded from the APT during initialization. The address loaded into the register represents the base address, i.e., the origin for relative address calculations, and can be changed at any time with the SETB instruction.
XO (Index)	The 15-bit XO register in the Data Path logic is loaded from the APT during initialization. The address loaded into the register, which may be changed at any time with the SETX instruction, defines the location of the first of eight index registers. These registers may be loaded, retrieved, and used in address calculations and are located in PDP-8 memory.

2.2 FPP8-A IOT INSTRUCTIONS

The PDP-8 IOT instructions relating to the FPP8-A are listed and described in Table 2-2. Table 2-3 lists and describes IOT instructions that are available for maintenance. All IOT instructions require one memory cycle. The FPP8-A uses device codes 55 and 56.

Table 2-2 FPP8-A IOT Instructions

Octal Code	Mnemonic	Description
6551	FPINT	Skip if the FPP8-A flag is set.
6552	FPICL	Produces same results as issuing initialize on the Omnibus. Initialize the FPP – clear flag, enable interleaved operation, stop the FPP, enable FP mode, clear all Status register flags. The APT pointer is not changed.

Table 2-2 FPP8-A IOT Instructions (Cont)

Octal Code	Mnemonic	Description
6553	FPCOM	If the FPP is not in a run state and the flag is not set, the FPP Command register is loaded with the contents of the PDP-8 AC. The AC is not changed by this IOT. If the FPP is in a run state or if the FPP flag is set, the FPCOM instruction is ignored.
6554	FPHLT	Force the FPP to exit, dump its status in the APT, set the forced-exit bit in the Status register, and set the FPP flag at the end of the current instruction. The following special features apply: 1. If FPHLT is issued prior to FPST, the FPP will single-step. FPHLT must be issued after FPIST (or FPICL) and before FPST for each instruction the FPP is to single-step. 2. If the FPP is currently in pause (result of FPAUSE instruction), the FPC will be decremented at exit. 3. If FPHLT and FEXIT occur at virtually the same time (causing a common exit), the status bit indicating forced exit (bit 2) will be cleared.
6555	FPST	If the FPP is not running and the FPP flag is not set, the contents of the AC are loaded into the 12 least-significant bits (LSBs) of the APTP register and the FPP is started. If the FPP is in the run state but paused, the FPST instruction will cause the FPP to continue. If the FPST instruction causes the FPP to start or continue, the next PDP-8 instruction is skipped. Unless the above conditions are met, the FPST instruction has no effect on the FPP and the PDP-8 skip does not occur.
6556	FPRST	Read (jam transfer) the FPP Status register into the PDP-8 AC. The FPRST IOT may be issued at any time.
6557	FPIST	Skip if the FPP flag is set. If the skip occurs, read the FPP Status register into the PDP-8 AC, clear the status bits and the FPP flag.
6567	FPEP	Select EP mode if AC0 = 1 and the FPP is not in the run state. Then clear the AC. This command must be issued after the FPCOM (6553) IOT if EP mode is desired.

Table 2-3 FPP8-A Maintenance IOT Instructions

Octal Code	Mnemonic	Description
6550	FFST	Start maintenance firmware. Forces a jump to μPC address 17. This address, in turn, contains an unconditional jump to μPC address 1700, the actual beginning of the maintenance firmware.
6560		Not used.
6561	FMODE	Enter Maintenance mode. This enabling instruction must be issued to cause the FPP to disable its internal free-running clock, and to cause the FPP to respond to IOT 6565. Maintenance mode is cleared by FPICL, FPIST (if the skip occurs), CAF and the initialize key. Entering Maintenance mode and issuing FPP instructions causes the FPP to function in an internal single-step mode. If the μPC is below 1000, the FPP will execute a data break for every FMDO instruction issued. Because of the way data breaks are synchronized on the Omnibus, this data break occurs after the memory cycle following the IOT, i.e., there is a one-memory-cycle delay between FMDO and the FPP data break. For μPC addresses of 1000 or higher, the FPP executes one microstep for every FMDO IOT. The FPP is clocked at the trailing edge of TP3 of the FMDO IOT. Instructions that require possible modification of index registers will not work properly in maintenance mode (JNX, LDX, ADDX, and indexed addressing).
6562		Not used.
6563	FMRB	Read Data buffer into AC (JAM transfer). This instruction may be executed in either Maintenance or Normal mode, but will result in erroneous information if the μPC is above 1000 and the FPP is in Normal mode.
6564	FMRP	Read μPC into AC (JAM transfer). This instruction may be issued in either mode, but will cause erroneous information to be read into bits 4–11 if the μPC is above 1000 and the FPP is in Normal mode.
6565	FMDO	Execute one step if in Maintenance mode. This instruction is ignored if not in Maintenance mode. See FMODE IOT above.
6566		Not used.
6567	FPEP	See description in Table 2-2.

2.3 FPP8-A OPERATING INSTRUCTIONS

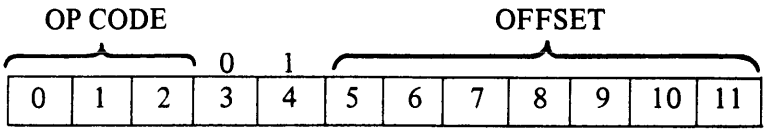
There are two basic classes of floating-point instructions: data reference instructions and special instructions. Data reference instructions perform arithmetic operations on specified data and transfer data between memory and the FAC. Special instructions cause jumps, branches, Index register modifications, pointer moves, manipulations of the FAC, and various housekeeping movements (e.g., alignment and normalization).

2.3.1 Data Reference Instructions and Formats

The 12 data reference instructions are listed in Table 2-4, along with a description of each. The operation carried out by each instruction is noted in the Operation column. For example, the FLDA instruction causes the operand (i.e., the contents, "C," of the effective address, "Y") to be loaded into the FAC. Each of the instructions, except LEA and LEAI, can use any one of three modes to specify the effective address. The format of these modes is illustrated in Figure 2-1. Bits 0, 1, and 2 (which represent the op code) identify the instruction, while bits 3 and 4 identify the mode of addressing. The remaining bits of each instruction determine the operand address, as described by the equations below each format. For example, the operand address for the single word, direct reference format is derived by multiplying bits 5 through 11 by 3 and adding the result to the 7 (or 8, since the product might overflow) LSBs of the base address.

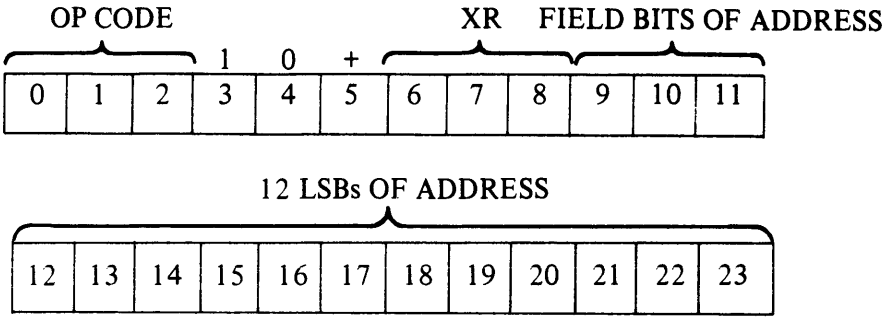
Table 2-4 FPP8-A Data Reference Instructions

Mnemonics	Op Code	Operation	Description
FLDA	0	C(Y)→FAC	The contents of the effective address are loaded into the FAC. If the mode is DP or FP, bits 24–59 of the FAC fraction are not used.
FADD FSUB	1 2	C(Y)+C(FAC)→FAC C(FAC)-C(Y)→FAC	The contents of the effective address are added to or subtracted from the contents of the FAC. In DP mode, no alignment or normalization occurs. The 24 bits from memory are combined with bits 0–23 of the FAC. In FP or EP mode, the two words are aligned by right-shifting the fraction with the lesser exponent until the two exponents are the same. In FP mode bits shifted out of bit 23 are shifted into bits 24–35. Bits shifted out of bit 35 (FP) or bit 59 (EP) are lost. The two fractions are then added or subtracted, using either the 24 MSB (FP) or all 60 bits (EP). The result is normalized. In FP mode the result is then rounded. If either argument is zero, or if its exponent is of such a value that alignment will shift the fraction completely out of its register, no shifting occurs. Under these circumstances, the FAC is either cleared or loaded with the contents of the effective address.
FDIV FMUL	3 4	C(FAC)/C(Y)→FAC C(FAC)*C(Y)→FAC	The old FAC is the multiplier or dividend; the contents of the effective location are the multiplicand or divisor. For multiply, the 36 (FP or DP) or 72 (EP) MSB of the product are computed. For divide, the division is carried to 26 or 61 bits. Lesser bits of product, or the division remainder are lost. In DP mode, the result is rounded to 24 bits. In FP mode, the result is normalized and then rounded to 24 bits. In EP mode, the result is normalized and truncated to 60 bits. For division in FP and EP modes, a preliminary test is made to ensure that the divisor is a normalized number. If the divisor is not normalized, it is first normalized before proceeding with the divide. This operation eliminates the possibility of divide overflow.
FADDM FMULM	5 7	C(Y)+C(FAC)→Y C(FAC)*C(Y)→Y	The calculation described above under FADD or FMUL occurs, except that the FAC is not changed. The result of the computation is stored at the effective address.
FSTA	6	C(FAC)→Y	The contents of the FAC are stored at the effective address. The FAC is not changed.
IMUL IMULI	6 7	C(FAC)*C(Y)→FAC	Available in DP mode only. The contents of the effective address are multiplied by the contents of the FAC, using the rules for integer arithmetic. (The binary point is to the right of bit 23.) The result is loaded into the FAC. A continuous test of overflow is maintained. If overflow occurs, the 24 bits in the FAC are the 24 LSB of the answer, but an unknown number of MSB have been lost.
LEA LEAI	6 7	Y→FAC	Available in FP and EP modes only. The effective address (not its contents) is loaded into bits 9–23 of the FAC. FAC bits 0–8 are cleared. The mode is then changed to DP.



Y = BASE ADDRESS + 3* OFFSET

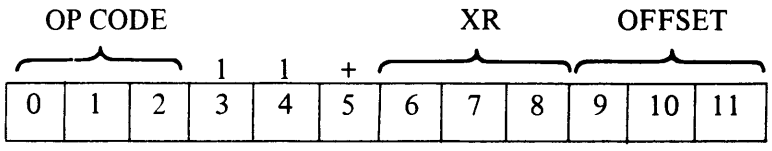
A. SINGLE-WORD, DIRECT REFERENCE



IF XR=0, Y=15-BIT ADDRESS AS GIVEN. THE CONTENTS OF INDEX REGISTER 0 ARE INCREMENTED IF BIT 5 IS LOGIC 1, BUT ARE NOT USED AS PART OF THE ADDRESS CALCULATION.

IF XR≠0, Y=ADDRESS + M*C (XR), WHERE M=2 (DP), 3 (FP), OR 6 (EP). IF BIT 5 IS LOGIC 1, THE CONTENTS OF THE ADDRESSED INDEX REGISTER ARE INCREMENTED BEFORE USE.

B. DOUBLE-WORD, DIRECT REFERENCE



IF XR=0, Y=BITS 21–35 OF THE TRIPLE WORD LOCATED AT THE BASE ADDRESS + 3* OFFSET. IF BIT 5 IS LOGIC 1, THE CONTENTS OF INDEX REGISTER 0 ARE INCREMENTED, BUT ARE NOT USED AS PART OF THE ADDRESS CALCULATION.

IF XR≠0, Y=BITS 21–35 OF THE TRIPLE WORD LOCATED AT THE BASE ADDRESS + 3* OFFSET + M*C (XR), WHERE M=2, 3, OR 6. IF BIT 5 IS LOGIC 1, THE CONTENTS OF THE ADDRESSED INDEX REGISTER ARE INCREMENTED BEFORE USE.

C. SINGLE-WORD, INDIRECT REFERENCE

Figure 2-1 Data Reference Formats

2.3.1.1 Single Word Direct Reference – The single word, direct reference format is employed when the operand is stored on the base page, which consists of a block of 384 12-bit locations. The origin of the base page is determined by the base address, which can be changed at any time; thus, the base page can encompass a block of locations anywhere in memory. The base address is contained in the BR, which is initially set from the APT and which can be changed with the SETB (Set Base Register) instruction. Data on the base page is stored in the FP format; i.e., the operand consists of three 12-bit words, namely, the 12-bit exponent followed by the 24-bit fraction. Consequently, 128 operands are available on the base page. The relative address of any operand exponent can be specified by multiplying the seven offset bits by 3. When this quantity is added to the base address, the location of the operand exponent is completely identified.

2.3.1.2 Double Word Direct Reference – The double word, direct reference format allows one to specify the 15-bit absolute address of an operand. In addition, this format permits address indexing, which simplifies programming techniques like loop counting and manipulation of push-down stacks.

Address indexing is accomplished by using the contents of an Index register to modify the 15-bit absolute address specified by the data reference instruction. There are eight consecutive 12-bit locations in the PDP-8 memory that are designated as FPP Index registers. The address, XO, of the first of these registers is provided initially by the APT, but can be changed by the special SETX instruction whenever necessary.

Bits 6, 7, and 8 (XR bits) of the double word, direct reference instruction identify Index registers 0 through 7 in octal notation. If Index register 0 is designated, no indexing is to be performed. Instead, the operand absolute address is given by bits 9–23 of the instruction, and the contents of Index register 0 may or may not be incremented. However, if an Index register other than 0 is specified, the 15-bit absolute address is modified by the contents of the selected Index register; note that the contents of the register may or may not be incremented before the operand address is calculated.

2.3.1.3 Single Word Indirect Reference – Indexing is also permitted by the single word, indirect reference instruction. Once again, bits 6, 7, and 8 identify the Index register that will be used in an address modification. However, in this case, the address is specified indirectly, using the base address as the point of reference. As before, bit 5 of the instruction determines if the contents of the Index register are incremented.

2.3.2 Special Instructions and Formats

The FPP special instructions are listed in Table 2-5, along with a description of each function.

Table 2-5 FPP8-A Special Instructions

Mnemonic	OP Code											
LTR	0	1	2	3	4	5	6	7	8	9	10	11
	1	0	1	0	0	0		COND		X	X	X

Function
Load Truth – If the condition is met, +1 (2000 0000 in DP mode) is loaded into the FAC. If the condition is not met, the FAC is cleared.

Conditions:

Octal	Meaning
0	FAC fraction = 0
1	FAC fraction greater than or equal 0
2	FAC fraction less than or equal 0
3	Always
4	FAC fraction not equal 0
5	FAC fraction less than 0
6	FAC fraction greater than 0
7	FAC exponent greater than 27 (octal)

Mnemonic	OP Code											
TRAP 3,	0	1	2	3	4	5	6	7	8	9	10	11
TRAP 4	(3 or 4)			0	0	X	X	X	X		MSB	
	12	13	14	15	16	17	18	19	20	21	22	23
	LSB of Address											

Trapped Instructions – The instruction trap status bit is set, and the FPP exits. The 15-bit address is placed in the APT.

Mnemonic	OP Code											
JNX	0	1	2	3	4	5	6	7	8	9	10	11
	0	1	0	0	0	+		XR			MSB	
	12	13	14	15	16	17	18	19	20	21	22	23
	LSB of Address											

Function
Jump if Index Register is non-zero – The specified Index register is incremented if bit 5 = 1. If the (incremented) Index register is not 0, bits 9–23 are loaded into the FPC, causing a jump.

Mnemonic	OP Code											
JSR	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	1	0	0	1	0	1	1		MSB	
	12	13	14	15	16	17	18	19	20	21	22	23
	LSB of Address											

Function
Jump and Save Return – A 'JA' to the current value of the FPC is constructed and stored in core memory locations BR+1 and BR+2. (1030+FPC field is stored in BR+1; 12 LSB of FPC is stored in BR+2.) Instruction bits 9–23 are then loaded into the FPC. This instruction is one of two ways to call a subroutine. Return from the subroutine is made by either doing a JA to BR+1, or by doing an FLDA base 0 followed by a JAC. The latter method is slightly slower, but much more general.

Table 2-5 FPP8-A Special Instructions (Cont)

Mnemonic	OP Code											
JSA	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	1	0	0	1	0	1	0		MSB	
	12	13	14	15	16	17	18	19	20	21	22	23
	LSB of Address											

Function
Jump and Save at Address – A 'JA' to the current value of the FPC is constructed and stored in core memory at the address specified by bits 9–23 of the instruction. The FPC is then changed to equal 2+bits 9–23 of the instruction. This is the second method for calling a subroutine, and stores the return in two memory locations at the top of the subroutine. Return is accomplished by an unconditional jump to the subroutine entry point.

Mnemonic	OP Code											
SETB	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	1	0	0	1	0	0	1		MSB	
	12	13	14	15	16	17	18	19	20	21	22	23
	LSB of Address											

Function
Set Base Register – Bits 9–23 are loaded into the BR.

Mnemonic	OP Code											
SETX	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	1	0	0	1	0	0	0		MSB	
	12	13	14	15	16	17	18	19	20	21	22	23
	LSB of Address											

Function
Set Index Register Pointer – Bits 9–23 are loaded into X0.

Mnemonic	OP Code											
BRANCH	0	1	2	3	4	5	6	7	8	9	10	11
INSTRUCTIONS	0	0	1	0	0	0		COND			MSB	
	12	13	14	15	16	17	18	19	20	21	22	23
	LSB of Address											

Function
Branch Instructions – If condition is met, bits 9–23 are loaded into the FPC, causing a jump to that address.

Conditions:

	Octal	Meaning
JEQ	0	If FAC fraction = 0
JGE	1	If FAC fraction greater than or equal 0
JLE	2	If FAC fraction less than or equal 0
JA	3	Always
JNE	4	If FAC fraction not equal 0
JLT	5	If FAC fraction less than 0
JGT	6	If FAC fraction greater than 0
JAL	7	If FAC exponent greater than 27 (octal). This condition signifies that the FAC contains a number too large to be fixed in 24 bits.

Table 2-5 FPP8-A Special Instructions (Cont)

Mnemonic	OP Code											
ADDX	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	0	0	0	1	0	0	1		XR	
	12	13	14	15	16	17	18	19	20	21	22	23
							Data					

Function

Add to Index Register – Bits 12–23 are added to the contents of the Index register specified by bits 9–11.

Mnemonic	OP Code											
LDX	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	0	0	0	1	0	0	0		XR	
	12	13	14	15	16	17	18	19	20	21	22	23
							Data					

Function

Load Index Register – Bits 12–23 are loaded into the Index register specified by bits 9–11.

Mnemonic	OP Code											
ALN	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	0	0	0	0	0	0	1		XR	

Function

In FP and EP mode, the fraction of the FAC is shifted until the FAC exponent equals the contents of the index register specified by bits 9–11. If bits 9–11 of the instruction are zero, the fraction of the FAC is shifted until the FAC exponent equals 27 octal (23 decimal).

In DP mode, an arithmetic shift is performed on the FAC. The number of shifts is equal to the value of the contents of the Index register specified by bits 9–11. The sign of the Index register indicates the direction of shift; a positive sign causes a shift toward the LSB. If the shift is toward the least significant bit, vacated bits are filled with FACO. If the shift is toward the most significant bit, vacated bits are filled with zeros. If bits 9–11 of the instruction are zero, a 23-bit right shift of the FAC is performed.

Mnemonic	OP Code											
ATX	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	0	0	0	0	0	1	0		XR	

Function

FAC to Index Register – If the mode is FP or EP, the contents of the FAC are fixed (i.e., shifted until the exponent = 27 octal) ATX does not test to see if fixing is possible. If the mode is DP, the contents of the FAC are already fixed, so this portion is omitted. Bits 12–23 of the result are then loaded into the Index register specified by bits 9–11. The FAC is not changed by the ATX instruction.

Table 2-5 FPP8-A Special Instructions (Cont)

Mnemonic	OP Code											
XTA	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	0	0	0	0	0	1	1		XR	

Function
Index Register to FAC – The contents of the Index register specified by bits 9–11 are loaded into FAC 12–23. FAC 0–11 is loaded with the contents of FAC 12.

FAC 24–59 are cleared, 27 octal is then loaded into the FAC exponent. If the mode is FP or EP, the FAC is then normalized. (The normalizing operation is omitted in DP mode.)

Mnemonic	OP Code	Function
NOP	004X	No Operation – None, other than a 1-cycle delay in the program. This is the only instruction which will always remain a NOP despite future expansion.
STARTE	005X	Start Extended-Precision Mode – The FPP enters EP mode. If the FPP was previously in a mode other than EP, FAC 24–59 are cleared.
FEXIT	0000	Exit Floating-Point – The contents of the FPP registers are dumped onto the active parameter table, the FPP is stopped, and the FPP flag is set.
FPAUSE	0001	Pause – Suspend FPP operations without updating the APT. IOT FPST will cause the FPP to continue.
FCLA	0002	Clear the FPP Accumulator – Make the FAC fraction zero. If the calculating mode is FP or EP, make the FAC exponent zero, also.
FNEG	0003	Complement the FPP Accumulator – The FAC fraction is replaced by its 2's complement.
FNORM	0004	Normalize – If the FAC fraction is non-zero, and if the FPP mode is FP or EP, the FAC fraction is shifted toward the MSB until the two MSBs are different from each other or until the FAC fraction equals 6000 0000. The FAC exponent is decremented by one for each position shifted. If the FAC fraction is 0, or if the mode is DP, no operation is performed.
STARTF	0005	Enter 24-Bit Floating Point Mode – The FPP enters FP mode. If issued in EP mode, the FAC is rounded to 24 bits.
STARTD	0006	Enter Double-Precision Mode – The FPP enters DP mode. If issued in EP mode, the FAC is chopped to 24 bits. The FAC exponent is ignored, but not modified.
JAC	0007	Jump Per FAC – FAC bits 9–23 are loaded into the FPC.

Reader's Comments

FPP8-A User's Manual
EK-FPP8A-OP-001

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults do you find with the manual? _____

_____

_____

_____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

_____

_____

_____

Would you please indicate any factual errors you have found. _____

_____

_____

_____

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____

— — — — — — — — — — — — Fold Here — — — — — — — — — — — —

— — — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — — —

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
Maynard, Massachusetts 01754