digital INTEROFFICE MEMORANDUM

8-0012-70

SUBJECT: PDP-15 Floating-Point Processor

DATE: August 19, 1970

TO: Engineering Committee

FROM: Tom Holmes TH.

DEPARTMENT: PDP-15 Engineering

The attached writeup is an introduction to the PDP-15 Floating-Point Hardware project - E-15-07635. It serves as a brief orientation in preparation for a presentation before the Engineering Committee on August 20, 1970. Included in the writeup are an opening introduction from Marketing, a discussion of floating-point architecture, an estimate of cost and development schedule, and several explanatory appendices.

/jan

Enclosure

ENGINEERING COMMITTEE

K. Olsen

R. Cady

D. Chertkow

A. Kent

F. Wilhelm

L. Gale

J. Cudmore

R. Doane

R. Dow

J. Milton

D. Best

G. Butler

B. Melesky

E. Correll

PDP-15

FLOATING-POINT PROCESSING UNIT (FPU)

ARCHITECTURE

THOMAS G. HOLMES
24 JULY 1970
EXT. 3440

TABLE OF CONTENTS

MARKETING INTRODUCTION

PRODUCT DEFINITION AND CONCEPT

THE FLOATING-POINT PROCESSOR IS DESIGNED TO BE A SEPARATE HARDWARE DEVICE WITH UP TO 16 INSTRUCTIONS PERFORMING SINGLE AND DOUBLE PRECISION ARITHMETIC.

THE UNIT IS DESIGNED TO SIT ON THE PDP-15 MEMORY BUS AND OPERATE AUTONOMOUSLY FROM THE CENTRAL PROCESSING UNIT. IT, THUS, ACTS AS A SEPARATE PROCESSOR PROVIDING THE USER WITH FOUR DISTINCT UNITS: MEMORY, CPU, I/O, AND FLOATING-POINT.

MARKETING

THE FLOATING-POINT HARDWARE PROVIDES A FURTHER COMPLEMENT TO THE PDP-15. IT SERVES MANY MARKETS AND CONSTITUTES A NECESSARY PRODUCT ENHANCEMENT. IT WILL ALSO HELP TO FURTHER DIFFERENTIATE PDP-11 AND PDP-15 PROSPECTS.

MARKETS THAT WILL BE AFFECTED BY THIS HARDWARE INCLUDE THE PHYSICS AND ANALYTICAL INSTRUMENTATION WHERE APPROXIMATELY 20% (15-20% FOR PHYSICS, 25% FOR AI) OF THE MACHINES ARE USED PRIMARILY FOR NUMERICA ANALYSIS; THE HYBRID MARKET WHERE ALMOST ALL MACHINES (90-100%) REQUIRE THE CALCULATION SPEED EITHER FOR PACING OR FOR DIGITAL SIMULATION; AND OUR GENERAL SCIENTIFIC MARKET WHERE HIGH SPEED ARITHMETIC IS DESIRABLE TO ACHIEVE BETTER PROGRAM THROUGHPUT (30% OF MACHINES). THE HIGH SPEED CAPABILITY WHEN COMPLEMENTED BY FORTRA SOFTWARE CAN REDUCE CALCULATION TIME BY AN ORDER OF MAGNITUDE AND MA ALLOW THE PDP-15 TO ENTER THE TRADITIONAL COMPUTATION MARKET NOW SERVED BY THE IBM 1130.

THE SCIENTIFIC MARKET IS PRESENTLY VERY TIGHT ON FUNDS. WE THEREFOR SEE THIS OPTION AS PRIMARILY AN ADD-ON. SCIENTISTS ARE PRESENTLY BUYING BASIC SYSTEMS; LATER AS END-OF-YEAR FUNDS AND A LOOSER ECONOM PERMIT, THE FLOATING-POINT PROCESSOR WILL BECOME A DESIRABLE ADD-ON. WE EXPECT THAT THE DEMAND WILL START UP SLOW, ACCELERATE AS THE POPULATION OF PDP-15'S EXPANDS AND SLOW DOWN AS THE PDP-15 IS PHASED OUT. THE PRODUCT LIFE IS ESSENTIALLY THAT OF THE PDP-15 ALTHOUGH THERE SHOULD STILL BE SOME RESIDUAL ADD-ON LIFE PAST THE PDP-15 PHASE OUT.

COMPETITION

FLOATING-POINT HARDWARE WILL ALLOW THE PDP-15 TO PERFORM WELL ON BENCHMARKS. CURRENTLY WE BENCHMARK POORLY EVEN COMPARED TO MACHINES WITH SLOWER CYCLE TIME DUE TO OUR SOFTWARE FLOATING-POINT ARITHMETIC IMPLEMENTED VIA EAE.

HEWLETT PACKARD HAS DEMONSTRATED BUT NOT YET DELIVERED FLOATING-POINT HARDWARE AT AN APPROXIMATE PRICE OF $20K. DELIVERY IS PLANNED IN JANUARY, 1971. THE HARDWARE WILL PERFORM TRIGONOMETRIC FUNCTIONS AS WELL AS FLOATING-POINT ADD, SUBTRACT, MULTIPLY, DIVIDE, AND NORMALIZE.

OUR PROPOSED HARDWARE WILL GIVE US A SIGNIFICANT ADVANTAGE IN THE MARKET OVER XDS AND HONEYWELL AND HELP IN THE MATCH WITH HEWLETT PACKARD. THE HARDWARE INTEGER ARITHMETIC ALLOWS US TO PERFORM DOUBLE PRECISION INTEGER CALCULATIONS AND THUS ANSWERS ONE OF HONEYWELL'S SELLING ARGUMENTS. IT ALSO PROVIDES AN EFFECTIVE MARKETING TOOL IN COMPARISON WITH THE HEWLETT PACKARD HARDWARE.

VOLUME

AT A SELLING PRICE OF $10-12K WE EXPECT TO SELL AROUND 250 FLOATING-POINT PROCESSORS.

1972				1973				1974
Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	
15	30	25	40	25	40	35	40	10

ROBERT L. KATZ

PDP-15 MARKETING

1. DESIGN GUIDELINES

THE FLOATING-POINT PROCESSOR MUST BE ABLE TO RUN THE EXISTING FORTRAN COMPILER AT AN IMPROVED SPEED AND WITH A MINIMUM NUMBER OF CHANGES TO THE COMPILER. THIS IS ACCOMPLISHED BY REPLACING EACH FLOATING-POINT SOFTWARE SUBROUTINE WITH AN EQUIVALENT HARDWARE INSTRUCTION. GLENN WICKELGREN'S MEMORANDUM OF JANUARY 6, 1970, SPECIFICATION FOR FLOATING-POINT HARDWARE FOR PDP-15, GIVES THE FOLLOWING:

"II. FLOATING-POINT INSTRUCTION FORMAT

THE FORTRAN IV COMPILER GENERATES A CALL TO A SUBROUTINE FOR EACH FLOATING OPERATION TO BE PERFORMED. THIS CALL LOOKS AS FOLLOWS:

```
JMS SUB          /FLOATING OP.
.DSA ARG         /ARGUMENT
                 /RETURN
```

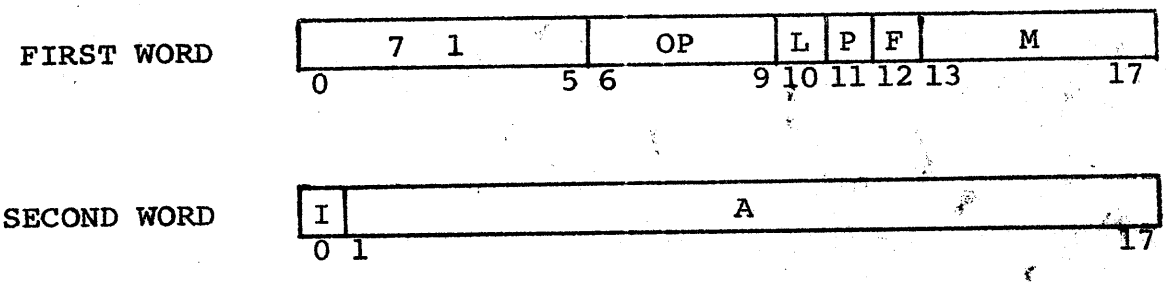
THE SUBROUTINE SUB IS DIFFERENT FOR DOUBLE PRECISION ARGUMENTS THAN IT IS FOR SINGLE PRECISION ARGUMENTS. THE ADDRESS ARG POINTS TO THE FIRST WORD (IS THE ADDRESS OF THE FIRST WORD) OF THE ARGUMENT OF THE FLOATING-POINT OPERATION (15 BIT ADDRESS). THE BITS 0-2 OF THE ARGUMENT ARE ZERO EXCEPT WHEN INDIRECTION IS INVOLVED. IF BIT 0 OF THE ARGUMENT IS 1 THEN BITS 3-17 CONTAIN THE ADDRESS OF A LOCATION WHOSE CONTENTS (BITS 3-17) POINT TO THE FIRST WORD OF THE ARGUMENT (ONLY 1 LEVEL OF INDIRECTION IS INVOLVED AND BITS 0-2 OF THIS WORD MUST BE IGNORED).* AFTER THE OPERATION IS COMPLETE, CONTROL IS RETURNED TO THE NEXT REGISTER FOLLOWING THE ARGUMENT ARG. IF FLOATING-POINT HARDWARE IS TO OPERATE SUCCESSFULLY WITH THE COMPILER, A HARDWARE INSTRUCTION MUST BE AVAILABLE WHICH CAN REPLACE THE INSTRUCTION (JMS SUB) WHICH TRANSFERS CONTROL TO SOFTWARE ROUTINES WHICH HANDLE THESE OPERATIONS. IN OTHER WORDS AN IOT (OR OTHER OP CODE) INSTRUCTION MUST REPLACE THESE ROUTINES.
* THE HARDWARE FLOATING-POINT PROCESSOR MUST ALSO PERFORM THIS ONE LEVEL OF INDIRECTION WHEN DIRECTED."

THE NECESSITY OF EXECUTING IN-LINE CODE GENERATED BY THE COMPILER MADE IT NECESSARY TO PLACE THE FPU ON THE MEMORY BUS, INSTEAD OF PLACING IT ON THE I/O BUS. THE FORTRAN COMPILER HAS DICTATED WHAT CONSTITUTES THE FPU. EARLIER DESIGNS OFFERRED A MORE POWERFUL FLOATING-POINT PACKAGE, BUT THE SERIOUS LIMITATION WAS THAT THE PRESENT FORTRAN COMPILER COULD NOT TAKE ADVANTAGE OF SUCH A UNIT WITHOUT UNDERGOING AN EXTENSIVE REWRITE. ALTHOUGH A COMPILER REWRITE WAS CONSIDERED, IT WAS REJECTED AS IMPRACTICAL AT THIS POINT IN TIME. SINCE THE MAJORITY OF PDP-15 USERS DEAL EXCLUSIVELY WITH FORTRAN, THE PROPOSED DESIGN WILL OFFER ONLY WHAT CAN BE EASILY IMPLEMENTED IN THE PRESENT FORTRAN COMPILER. THE SINGLE EXCEPTION TO THIS IS THE INCLUSION OF INTEGER ARITHMETIC, WHICH TAKES ADVANTAGE OF MOST OF THE FLOATING-POINT HARDWARE AND PROVIDES MARKETING WITH AN ATTRACTIVE MATH PACKAGE TO DISTINGUISH IT FROM ITS COMPETITION.

2. ORGANIZATION

INSTRUCTION FORMAT

THE FPU HAS BOTH SINGLE AND DOUBLE LENGTH INSTRUCTIONS.



BITS 00-05 OF THE FIRST WORD CONTAIN AN OCTAL "71" TO SPECIFY A FPU INSTRUCTION. THE "71" APPEARS AS A NO-OPERATION IN THE CPU.

OP — BITS 06-09 DECODE INTO SIXTEEN UNIQUE INSTRUCTIONS

L — BIT 10=0, SINGLE LENGTH INSTRUCTION
BIT 10=1, DOUBLE LENGTH INSTRUCTION

P — BIT 11=0, SINGLE PRECISION
BIT 11=1, DOUBLE PRECISION

F — BIT 12=0, INTEGER FORMAT
BIT 12=1, FLOATING FORMAT

* M — BITS 13-17 MODIFY THE LOAD AND STORE INSTRUCTIONS

13 UNASSIGNED
14 ROUND FROM DOUBLE TO SINGLE PRECISION
15 MAKE POSITIVE
16 MAKE NEGATIVE
17 COMPLEMENT

I — BIT 00=1 SPECIFIES INDIRECTION

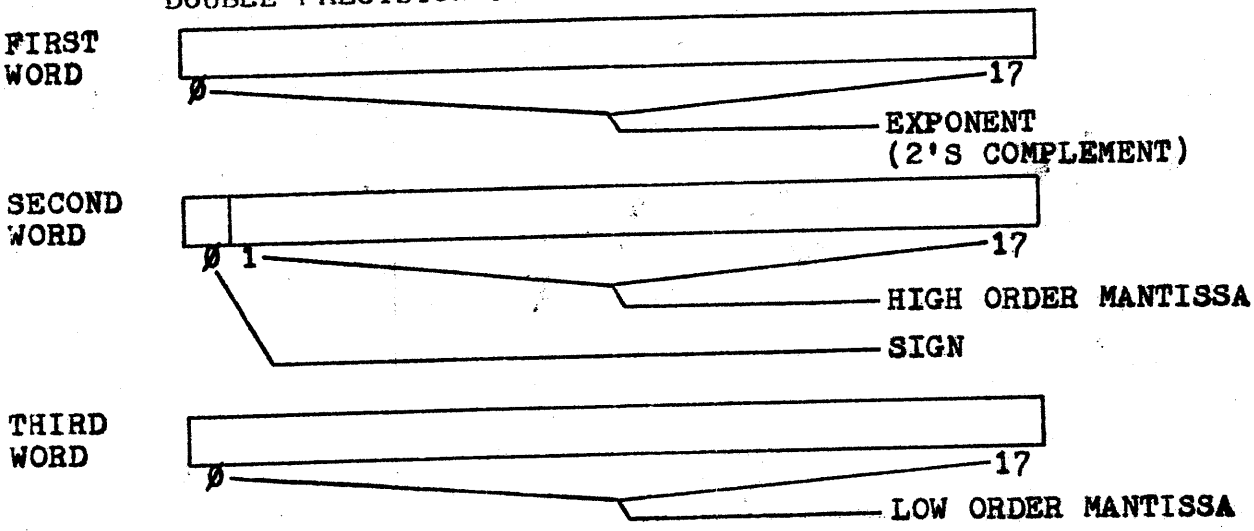
A — BITS 01-17 SPECIFY THE ADDRESS OF THE FIRST WORD OF THE ARGUMENT

* SEE APPENDIX A

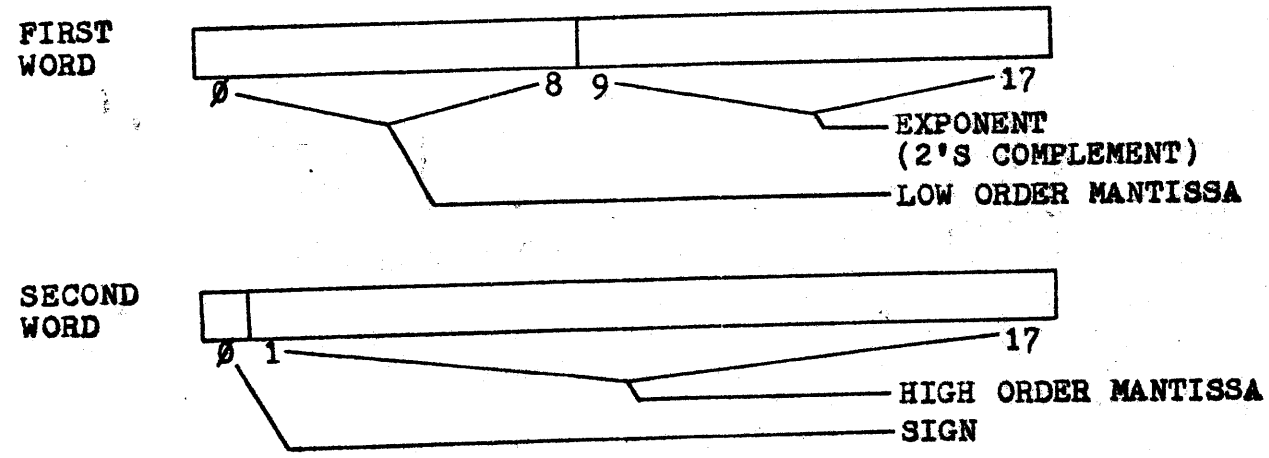
DATA FORMAT
---- ------

THE FPU SINGLE AND DOUBLE PRECISION DATA FORMATS ARE THE SAME AS THOSE IN THE EXISTING PDP-15 FLOATING-POINT SOFTWARE.

DOUBLE PRECISION FLOATING-POINT (54 BITS)



SINGLE PRECISION FLOATING-POINT (36 BITS)

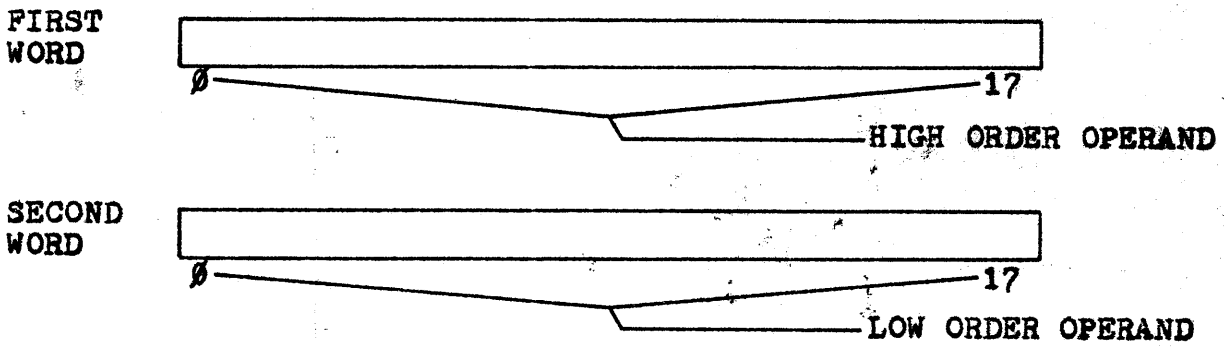


NOTE: THE FLOATING-POINT MANTISSA IS KEPT IN SIGN AND MAGNITUDE FORMAT. ALL OPERANDS ARE ASSUMED NORMALIZED.

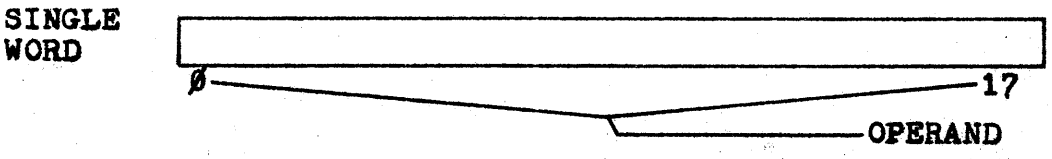
*

INTEGER ARITHMETIC ALSO HAS BOTH SINGLE AND DOUBLE PRECISION.

DOUBLE PRECISION INTEGER (36 BITS)

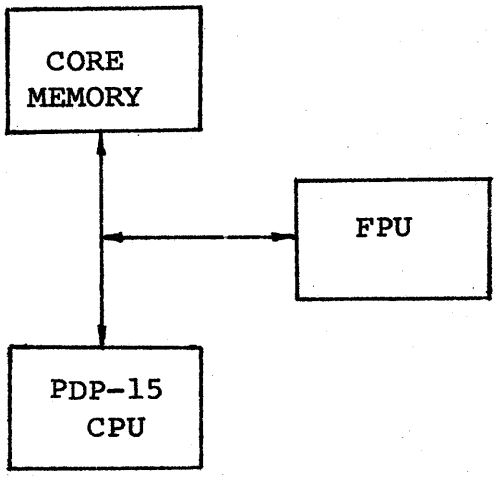


SINGLE PRECISION INTEGER (18 BITS)



NOTE: THE INTEGER OPERAND IS IN TWO'S COMPLEMENT FORMAT.
*

OPERATION



THE FLOATING-POINT SYSTEM CONFIGURATION HAS THE FPU "BLACK BOX" PHYSICALLY OR'D ON THE MEMORY BUS BETWEEN CORE MEMORY AND THE PDP-15 CPU. THUS SITUATED, THE FPU MONITORS ALL INSTRUCTIONS FETCHED FROM CORE AND EXAMINES EACH FOR THE "71" IN BITS 00-05. THE "71" APPEARS AS NO-OPERATION IN THE CPU, BUT IT IMMEDIATELY FLAGS THE FPU TO ANNOUNCE ITS OCCURRANCE. THE FPU ACKNOWELDGES BY DISABLING THE CONTROL CIRCUITRY WHICH ALLOWS THE CPU TO MAKE A MEMORY REQUEST. I/O MEMORY REQUESTS ARE STILL ALLOWED ACCESS TO CORE. THE BITS TO THE RIGHT OF "71" PROVIDE INFORMATION ON THE INSTRUCTION TYPE, INSTRUCTION LENGTH, DATA PRECISION, AND DATA FORMAT.

IF A DOUBLE LENGTH INSTRUCTION IS SPECIFIED, THE CPU IS ENABLED TO ALLOW FOR ONE MEMORY ACCESS TO OBTAIN THE SECOND WORD OF THE INSTRUCTION. THE CPU PC WILL INCREMENT PAST THIS SECOND WORD AND BE CORRECTLY POSITIONED FOR CONTINUED EXECUTION WHEN CONTROL IS RETURNED. THIS TECHNIQUE OF ALIGNING THE PC REQUIRES THAT THE SECOND WORD BE MADE TO LOOK LIKE A NO-OPERATION TO THE CPU. THE ACTUAL SECOND WORD IS LOADED INTO THE FPU. THIS SECOND WORD CAN BE EITHER THE ADDRESS OF THE MEMORY OPERAND (NO INDIRECTION) OR THE ADDRESS POINTING TO THE ADDRESS OF THE MEMORY OPERAND (INDIRECTION). EVENTUAL TRANSFER OF THE OPERAND TAKES ONE, TWO, OR THREE ACCESSES TO CORE MEMORY, DEPENDING ON THE PRECISION AND DATA FORMAT TRANSFERRED. THE LAST PHASE OF THE INSTRUCTION IS TO RELEASE THE HOLD ON THE CPU MEMORY REQUEST CIRCUITRY.

3. HARDWARE IMPLEMENTATION

SEVERAL MSI (MEDIUM SCALE INTEGRATION) DEVICES ARE PLANNED TO BE USED IN THE FPU.

DEC74181	4-BIT ARITHMETIC LOGIC UNIT
DEC74182	CARRY LOOK AHEAD
DEC74153	4-LINE TO 1-LINE DATA SELECTOR
DEC74154	4-LINE TO 16-LINE DECODER
DEC7495	4-BIT SHIFT REGISTER
DEC74193	4-BIT SYNCHRONOUS UP/DOWN COUNTER

THREE GENERAL PURPOSE BOARDS HAVE BEEN MADE TO UTILIZE THE ABOVE IC'S.

50-08912	TWO 16-PIN IC'S, ALL PINS BROUGHT OUT, POWER ON PINS 16 AND 8, SEPARATE GROUND LOOP FOR EACH IC, AVAILABLE AS W961.
50-08908	ONE 24-PIN IC, ALL PINS BROUGHT OUT, POWER ON PINS 24 AND 12, AVAILABLE AS W962.
50-08914	TWO 14-PIN IC'S, ALL PINS BROUGHT OUT, POWER ON PINS 14 AND 7, SEPARATE GROUND LOOP FOR EACH IC, AVAILABLE AS W963.

THE FIRST PROTOTYPE WILL MAKE EXTENSIVE USE OF THESE BOARDS IN THE DESIGN. SPECIAL PARTITIONING HAS ALREADY BEEN DEVELOPED FOR THE SECOND PROTOTYPE WHICH WILL COMBINE SIX OF THESE SINGLE HEIGHT CARDS INTO ONE DOUBLE HEIGHT CARD. THIS NEW CARD WILL BE USED TO REDUCE THE 96 MODULE SLOTS REQUIRED FOR REGISTERS AND ADDERS IN PROTOTYPE I TO 32 MODULE SLOTS IN PROTOTYPE II. IF THE MODULE SLOTS REQUIRED FOR CONTROL LOGIC CAN BE KEPT TO LESS THAN 60% OF THE TOTAL COUNT, THE FPU CAN BE PACKAGED IN TWO MOUNTING PANELS (128 MODULE SLOTS). AS A COMPARISION, THE PDP-15 CPU IS CONTAINED IN 2 1/4 MOUNTING PANELS AND USES 79 OF THOSE MODULE SLOTS FOR CONTROL. IF THE FPU CAN BE CONSIDERED TO BE NO MORE COMPLEX (AND/OR COMPLICATED) THAN THE CPU, THE LIKELIHOOD OF FITTING THE FPU IN THE ALLOTTED SPACE IS GOOD.

4. DEVELOPMENT SCHEDULE AND MANUFACTURING COST ESTIMATE

IN ARRIVING AT MANUFACTURING COST, AN AVERAGE COST OF TEN DOLLARS PER MODULE WAS USED FOR EXISTING MODULES AND PRESENT INTEGRATED CIRCUIT COSTS WERE USED TO DETERMINE THE COST OF NEW MODULES. INFORMATION FROM THE PURCHASING DEPARTMENT INDICATES THAT THE MSI COMPONENTS ON THE NEW MODULES WILL EXPERIENCE A 40 - 50% COST REDUCTION BY THE FIRST QUARTER OF FISCAL 1972.

MANUFACTURING COST

MOUNTING PANEL WITH A2, C2, & T1 BUSSED	2	36.75	73.50
I/O CABLES USING M904 CONNECTORS	3	15.60	46.80
M775 CLOCK AND TIME STATE GENERATOR	1	28.31	28.31
WIREWRAP SERVICE 2,000 WIRES @ $0.12	1	240.00	240.00
MANTISSA ADDER MODULE (PARTS AND MANF. COST)	9	30.00	270.00
ADDRESS ADDER MODULE (PARTS AND MANF. COST)	5	24.00	120.00
ADDITIONAL STANDARD MODULES	64	10.00	640.00
			$1,339.61

MATERIALS	$1,339.00
INSTALLATION IN FIELD (1-2HRS)	20.00
CHECKOUT (3 MANDAYS)	240.00
	$1,599.00

AFTER CONFERRING WITH J. GALVIN AND E. SIMEONE, PROVISION WAS MADE FOR THE POSSIBLE LATER ADDITION OF A POWER SUPPLY AND FOR A TEN PERCENT INCREASE IN COST.

	$1,599.00
POWER SUPPLY	154.40
	$1,743.30
10% INCREASE IN COST	174.90
	$1,923.90

DIGITAL EQUIPMENT CORPORATION — DISCRETE PROJECT AUTHORIZATION

new ☐
change ☐

Product Code: 15 | Market Code: 0 | Discrete Project Code: 7635

(To be assigned by accounting)

Project Title: PDP-15 FLOATING-POINT PROCESSOR Date: 6-26-70

Project Manager: T. Holmes Supervisor: E. Aumann

Complete Description of Project:

Estimated Completion Date: 4Q 1971 Estimated Total $'s: $189.3

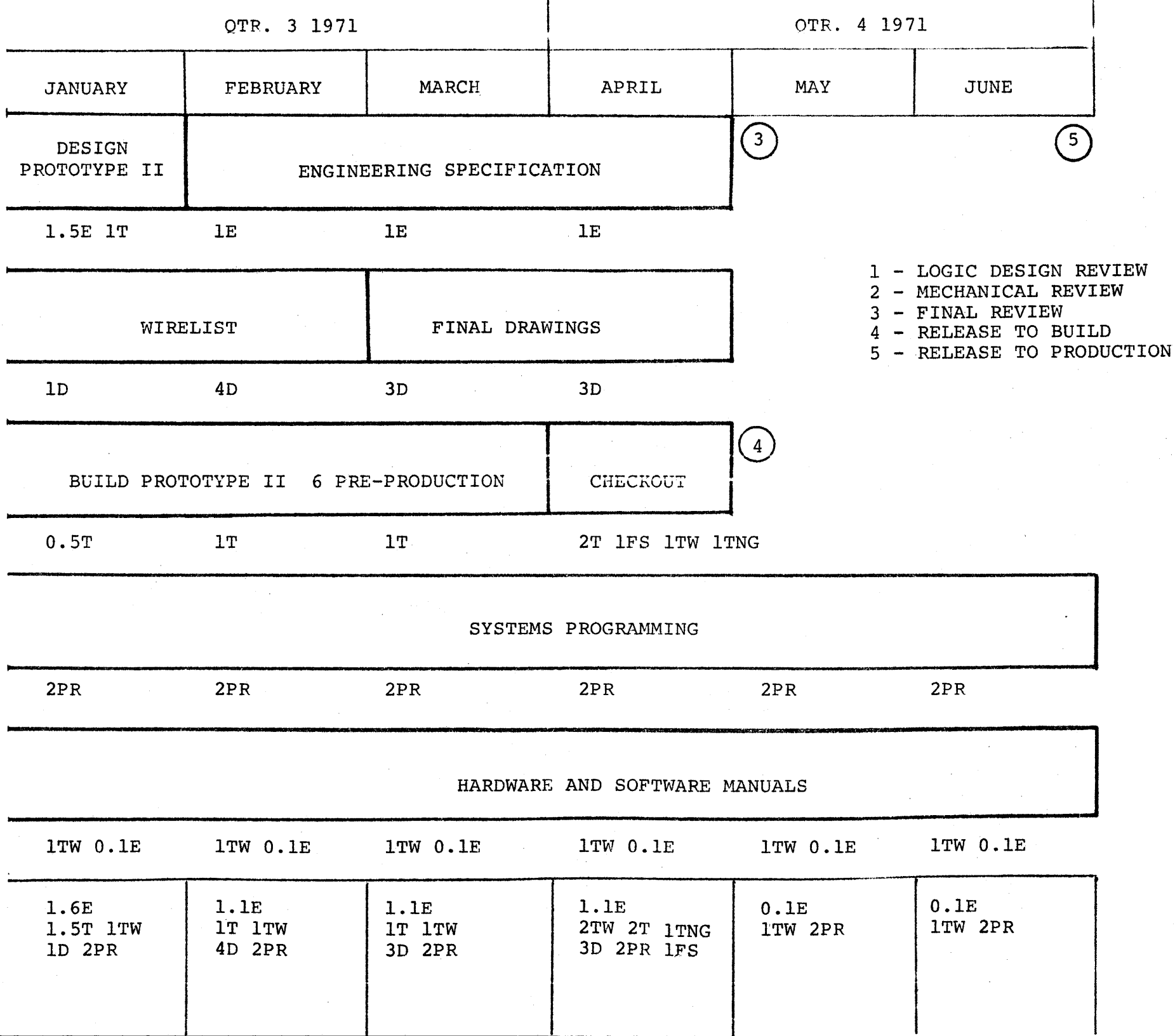
Act. Code	Cost Ctr.	Expense Centers		Fiscal Quarters FY 71 Qtr. 1	FY 71 Qtr. 2	FY 71 Qtr. 3	FY 71 Qtr. 4
E	324	Model Shop	G. Gerelds		20.0	13.0	
D	325	Drafting	R. Melanson	1.3		9.0	4.5
D	330	Mechanical Eng.	L. Prentice	0.5	0.5		
D	339	Process Eng.	T. Stockebrand				
P	360	Systems Programming	L. Portner				
V	360	Diagnostic Programming	L. Portner	3.7	18.0	18.0	18.0
D	374	Production Eng.	R. Puffer		2.0	6.0	3.0
D	386	Special Projects	J. St.Amour				
N	551	Hardware Manuals	J. Bellantoni			4.0	3.8
Y	552	Software Manuals	G. Arnold			3.0	3.0
E		Product Line Eng.		9.0	15.5	30.5	9.0
T	287	Advertising & Promotion	G. D'Annunzio				
A		Product Line Marketing					
		TOTAL EXPENSE		14.5	56.0	83.5	35.3

If this is a new project, have Product Line Manager sign below and submit to Accounting for assignment of a project number. The appropriate Vice President should sign for any projects shared by more than one Product Line.

Approved by: Date:

Authorization of this project does not constitute budget approval. Each project must operate within and under the control of their appropriate Product Line and its budget.

QTR. 4 1970	QTR. 1 1971			QTR. 2 1971		
JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER
LAYOUT AND DESIGN OF PROTOTYPE I			(1) DOCUMENT	LAYOUT AND DESIGN OF PROTOTYPE II		
1E	1.5E 1T		1E	1.5E 1T	1.5E 1T	1.5E 1T
WIRELIST						
1D 1T	1D 1T					
MECHANICAL LAYOUT AND DESIGN OF SYSTEM			(2)	FINAL DESIGN		
0.1E 0.5D				0.2E		
SPECIFY PARTS	BUILD PROTOTYPE I	CHECKOUT OF PROTOTYPE I				
0.5T	0.5T	1T		1T	0.5T	
DIAGNOSTIC PROGRAMMING						
1PR	1PR	2PR		2PR	2PR	
1E 1T 1D	1.6E 2.5T 1.5D 1PR	1E 0.5T 1PR		1.5E 2T 2PR	1.7E 2T 2PR	1.5E 1.5T 2PR



1 - LOGIC DESIGN REVIEW
2 - MECHANICAL REVIEW
3 - FINAL REVIEW
4 - RELEASE TO BUILD
5 - RELEASE TO PRODUCTION

APPENDIX A

INSTRUCTION SET

	INSTRUCTION LENGTH
ADD	DOUBLE
SUBTRACT	DOUBLE
REVERSE SUBTRACT	DOUBLE
MULTIPLY	DOUBLE
DIVIDE	DOUBLE
REVERSE DIVIDE	DOUBLE
LOAD	SINGLE AND DOUBLE
STORE	DOUBLE
FLOAT	SINGLE AND DOUBLE
FIX	SINGLE AND DOUBLE
DIAGNOSTIC	DOUBLE
SWAP MQ	SINGLE
UNASSIGNED	
UNASSIGNED	
UNASSIGNED	
UNASSIGNED	

THE FIRST OPERAND LOCATION IS A 54-BIT HARDWARE ACCUMULATOR IN THE FPU. FOR DOUBLE LENGTH INSTRUCTIONS THE SECOND OPERAND IS HELD IN CORE MEMORY. WHEN THE INSTRUCTION LENGTH IS SINGLE, THE FPU HARDWARE ACCUMULATOR IS USED AS THE SECOND OPERAND.

IN FLOATING-POINT MULTIPLY THE NORMALIZED RESULT HAS THE MOST SIGNIFICANT BITS IN THE FPU HARDWARE ACCUMULATOR AND THE LEAST SIGNIFICANT BITS IN THE FPU MQ. THE MQ, A 36-BIT EXTENSION OF THE HARDWARE ACCUMULATOR, MAY BE ACCESSED WITH THE DIAGNOSTIC AND SWAP MQ INSTRUCTIONS. IN INTEGER MULTIPLY, THE LEAST SIGNIFICANT BITS OF THE RESULT ARE IN THE FPU HARDWARE ACCUMULATOR AND THE MOST SIGNIFICANT BITS ARE IN THE FPU MQ. MULTIPLICATION IS BETWEEN TWO 36-BIT OPERAND REGISTERS AND RESULTS APPEAR IN THE 36-BIT MANTISSA PORTION OF THE FPU HARDWARE ACCUMULATOR AND THE 36-BIT FPU MQ.

BOTH FLOATING-POINT DIVIDE AND INTEGER DIVIDE RESULT WITH THE QUOTIENT IN THE 36-BIT MANTISSA PORTION OF THE HARDWARE ACCUMULATOR AND THE REMAINDER IN THE MQ. INTEGER DIVISION IS WHOLE NUMBER DIVISION. IF THE DIVIDEND IS LESS THAN THE DIVISOR, THE QUOTIENT IS ZERO.

PROGRAM INTERRUPTIONS:
OVERFLOW - MAGNITUDE OF RESULT EXCEEDED MAXIMUM REPRESENTABLE NUMBER LIMIT.
UNDERFLOW - MAGNITUDE OF RESULT EXCEEDED MINIMUM REPRESENTABLE NUMBER LIMIT.
DIVIDE - DIVISION BY ZERO IS ATTEMPTED.

ADD

THE SECOND OPERAND IS ADDED TO THE FIRST OPERAND, AND THE SUM IS PLACED IN THE FIRST OPERAND LOCATION.

PROGRAM INTERRUPTIONS:
UNDERFLOW
OVERFLOW

SUBTRACT

THE SECOND OPERAND IS SUBTRACTED FROM THE FIRST OPERAND, AND THE DIFFERENCE IS PLACED IN THE FIRST OPERAND LOCATION.

PROGRAM INTERRUPTIONS:
UNDERFLOW
OVERFLOW

REVERSE SUBTRACT
------- --------

THE FIRST OPERAND IS SUBTRACTED FROM THE SECOND OPERAND, AND THE DIFFERENCE IS PLACED IN THE FIRST OPERAND LOCATION.

PROGRAM INTERRUPTIONS:
UNDERFLOW
OVERFLOW

MULTIPLY

THE PRODUCT OF MULTIPLIER (THE SECOND OPERAND) AND MULTIPLICAND (THE FIRST OPERAND) REPLACES THE MULTIPLICAND.

PROGRAM INTERRUPTIONS:
UNDERFLOW
OVERFLOW

DIVIDE

THE DIVIDEND (THE FIRST OPERAND) IS DIVIDIED BY THE DIVISOR (THE SECOND OPERAND) AND REPLACED BY THE QUOTIENT. THE REMAINDER IS LEFT IN THE MQ.

PROGRAM INTERRUPTIONS:
UNDERFLOW
OVERFLOW
DIVIDE

REVERSE DIVIDE
------- ------

THE DIVIDEND (THE SECOND OPERAND) IS DIVIDED BY THE DIVISOR (THE FIRST OPERAND). THE QUOTIENT IS PLACED IN THE FIRST OPERAND LOCATION.

PROGRAM INTERRUPTIONS:
UNDERFLOW
OVERFLOW
DIVIDE

LOAD

THE SECOND OPERAND IS PLACED IN THE FIRST OPERAND LOCATION. BITS 13-17, M, OPERATE ON THE SECOND OPERAND PRIOR TO THE LOAD. USE OF THESE BITS CREATES A SUBSET OF LOAD INSTRUCTIONS:

LOAD ROUNDED
LOAD POSITIVE
LOAD NEGATIVE
LOAD COMPLEMENT

PROGRAM INTERRUPTIONS:
OVERFLOW (OCCURS ONLY ON LOAD ROUNDED)

STORE

THE FIRST OPERAND IS STORED AT THE SECOND OPERAND LOCATION. BITS 13-17, M, OPERATE ON THE FIRST OPERAND PRIOR TO THE STORE. USE OF THESE BITS CREATES A SUBSET OF STORE INSTRUCTIONS:

STORE ROUNDED
STORE POSITIVE
STORE NEGATIVE
STORE COMPLEMENT

PROGRAM INTERRUPTIONS:
OVERFLOW (OCCURS ON STORE ROUNDED AND WHEN POSITIVE EXPONENT BITS ARE LOST ON SINGLE PRECISION STORE)
UNDERFLOW (NEGATIVE EXPONENT BITS LOST ON SINGLE PRECISION STORE)

NOTE: LOAD ROUNDED AND STORE ROUNDED CHANGE DATA PRECISION FROM DOUBLE TO SINGLE BY ADDING 000000000400 TO THE MANTISSA AND RENORMALIZING.

FLOAT

THE DATA FORMAT OF THE SECOND OPERAND IS CHANGED FROM INTEGER TO FLOATING-POINT. THE SECOND OPERAND IS A 18 OR 36-BIT INTEGER. THE RESULT IS PLACED IN THE FIRST OPERAND LOCATION.

PROGRAM INTERRUPTIONS: NONE

FIX

THE DATA FORMAT OF THE SECOND OPERAND IS CHANGED FROM FLOATING-POINT TO INTEGER. THE RESULT IS A 18 OR 36-BIT INTEGER AND IS PLACED IN THE FIRST OPERAND LOCATION.

PROGRAM INTERRUPTIONS:
OVERFLOW (SIGNIFICANT BITS LOST ON FORMAT CHANGE)

DIAGNOSTIC

THIS INSTRUCTION WILL BE IMPLEMENTED TO DIAGNOSTIC PROGRAMMING SPECIFICATIONS. IN GENERAL, IT WILL ENABLE THE DIAGNOSTIC PROGRAMMER TO PUT THE FLOATING-POINT UNIT IN SINGLE-STEP CHECKOUT MODE AND PROVIDE ACCESS TO ALL ESSENTIAL CONTROL CIRCUITRY AND HARDWARE REGISTERS. THIS INSTRUCTION WILL ALSO SERVE TWO OTHER FUNCTIONS. FIRST, IT WILL ALLOW USER INFORMATION IN THE FPU TO BE SAVED OR RESTORED WITH A SINGLE INSTRUCTION. SECOND, IT WILL GIVE ACCESS TO AN INTERRUPT EXCEPTION REGISTER, WHICH MAY BE EXAMINED TO DETERMIN THE CAUSE OF THE FLOATING-POINT INTERRUPT.

SWAP MQ
---- --

THE 36-BIT MANTISSA PORTION OF THE FPU HARDWARE ACCUMULATOR AND THE 36-BIT FPU MQ ARE SWAPPED.

PROGRAM INTERRUPTIONS: NONE

APPENDIX B

HANDLING OF INTERRUPT EXCEPTIONS

OVERFLOW - MAGNITUDE OF RESULT EXCEEDED MAXIMUM REPRESENTABLE NUMBER LIMIT.

UNDERFLOW - MAGNITUDE OF RESULT EXCEEDED MINIMUM REPRESENTABLE NUMBER LIMIT.

DIVIDE - DIVISION BY ZERO IS ATTEMPTED.

WHEN AN INTERRUPT EXCEPTION OCCURS, EXECUTION STOPS AND THE FPU AUTOMATICALLY FORCES THE CPU TO JMS TO THE EXIT ADDRESS. THE EXIT ADDRESS IS KEPT IN A 15-BIT REGISTER IN THE FPU, WHICH CAN BE ACCESSED BY THE PROGRAMMER THROUGH USE OF THE DIAGNOSTIC INSTRUCTION. UPON COMPLETION THE EXIT ADDRESS LOCATION WILL CONTAIN THE ADDRESS OF THE LAST WORD OF THE LAST FPU INSTRUCTION EXECUTED PLUS TWO.

E.G.

	A	FLOAT. ADD
FPU INTERRUPT OCCURS	A+1	ADDRESS OF ARG.
	A+2	LAC B
EXIT ADDRESS LOCATION	A+3	DAC C
WILL CONTAIN "A+3"	.	.
	.	.

PRESUMABLY, THE EXIT ADDRESS WILL POINT TO THE ERROR HANDLING ROUTINE FOR FLOATING-POINT. THE DIAGNOSTIC INSTRUCTION MAY BE USED TO LOOK AT THE STATUS OF THE OVERFLOW, UNDERFLOW, AND DIVIDE INTERRUPT FLOPS AND THE INSTRUCTION LENGTH OF THE INTERRUPTED INSTRUCTION.

APPENDIX C

OPERATING FEATURES AND CONSTRAINTS

- MAXIMUM OF ONE LEVEL OF INDIRECTION
- OPERANDS MUST BE NORMALIZED
- RESULTS ARE AUTOMATICALLY NORMALIZED
- IF A CALCULATION RESULTS IN A ZERO MAGNITUDE, THE EXPONENT IS ZERO'D AND THE SIGN MADE POSITIVE
- NEGATIVE ZERO IS NOT ALLOWED
- THE ADDRESS OF THE ARGUMENT IS A 17-BIT NUMBER
- ROUNDING TAKES PLACE AFTER ALIGNING AND AFTER NORMALIZING (+1 IS ADDED TO THE LEAST SIGNIFICANT BIT OF THE MANTISSA IF THE NEXT BIT OFF THE LSB END OF THE MANTISSA IS 1)
- I/O MEMORY REQUESTS ARE GIVEN PRIORITY OVER FPU MEMORY REQUESTS
- THE PROGRAMMER MAY LOAD A 15-BIT REGISTER IN THE FPU WITH AN INTERRUPT EXIT ADDRESS. WHEN A FLOATING-POINT INTERRUPT OCCURS, THE FPU AUTOMATICALLY FORCES THE CPU TO JMS TO THE EXIT ADDRESS. THE EXIT ADDRESS LOCATION WILL CONTAIN THE ADDRESS OF THE LAST FPU INSTRUCTION EXECUTED PLUS TWO.

E.G.

```
                          A       FLOAT. ADD
FPU INTERRUPT OCCURS      A+1     ADDRESS OF ARG.
                          A+2     LAC B
EXIT ADDRESS LOCATION     A+3     DAC C
WILL CONTAIN "A+3"        .       .
                          .       .
```

APPENDIX D

EXAMPLE PROGRAM CODING

OP CODES USED: LOAD=20, ADD=34, STORE=70
MNEMONIC: AD DOUBLE PRECISION FLOATING-POINT ADD
LD DOUBLE PRECISION FLOATING-POINT LOAD
STD DOUBLE PRECISION FLOATING-POINT STORE

```
000100                      .LOC    100
000100  340112             TAD     DATA1   /CPU INSTRUCTION
000101  040113             DAC     STOR    /CPU INSTRUCTION
000102  712340             LD      NUM1    /DOUBLE PRECISION F.P. LOAD
000103  000114                             /ASSEMBLES INTO TWO 18-BIT WORDS
000104  713740             AD      NUM2    /DOUBLE PRECISION F.P. ADD
000105  000117                             /ASSEMBLES INTO TWO 18-BIT WORDS
000106  717340             STD     ANSW    /DOUBLE PRECISION F.P. STORE
000107  000122                             /ASSEMBLES INTO TWO 18-BIT WORDS
000110  200125             LAC     DATA2   /CPU INSTRUCTION
000111  600126             JMP     NEXT    /CPU INSTRUCTION
000112  101127  DATA1      101127
000113  000000  STOR       0
000114  000047  NUM1       47              /EXPONENT
000115  364211             364211          /SIGN AND HIGH ORDER MANTISSA
000116  373642             373642          /LOW ORDEP MANTISSA
000117  000036  NUM2       36              /EXPONENT
000120  207416             207416          /SIGN AND HIGH ORDER MANTISSA
000121  477135             477135          /LOW ORDER MANTISSA
000122  000000  ANSW       0               /EXPONENT
000123  000000             0               /SIGN AND HIGH ORDER MANTISSA
000124  000000             0               /LOW ORDER MANTISSA
000125  000001  DATA2      1
000126  740000  NEXT                       /CPU INSTRUCTION
                           .
                           .
```

WHEN THE INSTRUCTION AT LOCATION 000102 IS FETCHED, THE WORD, 712340, IS A NO-OPERATION IN THE CPU. HOWEVER, THE FPU IMMEDIATELY RECOGNIZES THE "71" AS A FLOATING-POINT INSTRUCTION WORD. AT THIS TIME, THE FPU DISABLES THE CPU FROM MAKING FURTHER MEMORY REQUESTS, AND THE REST OF THE FLOATING-POINT INSTRUCTION IS DECODED - 2340.

SINCE L=1, THE CPU IS ALLOWED TO MAKE ITS NEXT REQUEST TO MEMORY. THE CPU NOW THINKS IT IS FETCHING THE NEXT INSTRUCTION FROM LOCATION 000103. THE CONTENT OF 000103 IS REALLY MEANT TO BE THE SECOND OPERAND ADDRESS FOR THE FLOATING-POINT INSTRUCTION. CONSEQUENTLY, THE CPU MUST NEVER SEE THE "GARBAGE" INSTRUCTION IN 000103. THIS IS ACCOMPLISHED BY DISABLING THE CPU FROM STROBING THE CONTENTS OF 000103 INTO ITS BUFFER. BECAUSE OF THE "STIMULUS-

RESPONSE" OPERATION OF THE CPU WITH CORE MEMORY, THE CPU WILL WAIT IN LIMBO UNTIL IT IS ALLOWED TO STROBE THE MEMORY BUS INTO ITS BUFFER. MEANWHILE THE FPU TAKES OVER WHERE THE CPU LEFT OFF, STROBES THE CONTENT OF 000103 INTO THE FPU BUFFER, AND ISSUES THE NECESSARY ACKNOWLEDGE SIGNALS TO END THE MEMORY CYCLE. A DUMMY NO-OPERATION INSTRUCTION WORD IS THEN PUT ON THE MEMORY BUS BY THE FPU, FOLLOWED BY A STROBE SIGNAL TO THE CPU. THE FPU NOW PLAYS THE ROLE OF A CORE BANK AND ISSUES THE NECESSARY SIGNALS TO THE CPU TO END THE CYCLE. AGAIN, THE FPU DISABLES THE CPU FROM MAKING A MEMORY REQUEST.

OP CODE = 20 IS A LOAD INSTRUCTION. P = 1 AND F = 1, COMPLETE THE DESCRIPTION OF THE INSTRUCTION, LABELING IT DOUBLE PRECISION FLOATING-POINT. THE DESTINATION OF THE LOAD IS A 54-BIT FPU HARDWARE ACCUMULATOR. THE DESTINATION ACCUMULATOR IS LOADED WITH THE EXPONENT, SIGN AND HIGH ORDER MANTISSA, AND THE LOW ORDER MANTISSA FROM LOCATIONS 000114, 000115, AND 000116 RESPECTIVELY. THE CONTENT OF 000103 IS THE SECOND OPERAND ADDRESS. SINCE BIT 0=0, THERE IS NO INDIRECTION, AND THE EFFECTIVE ADDRESS IS SIMPLY 000113. WHEN THE INSTRUCTION IS FINISHED, THE CPU IS ENABLED FOR MEMORY REQUESTS AND THE NEXT WORD FETCHED BY THE CPU IS FROM LOCATION 000104. THE OTHER FLOATING-POINT INSTRUCTIONS ARE EXECUTED IN A SIMILAR MANNER.

NOTE: I/O MEMORY REQUESTS ARE GIVEN PRIORITY OVER FPU REQUESTS TO MEMORY.