PERKIN-ELMER

HIGH PERFORMANCE FLOATING-POINT PROCESSOR (HPFPP)

Installation and Maintenance Manual

29-705 R11

The information in this document is subject to change without notice and should not be construed as a commitment by the Perkin-Elmer Corporation. The Perkin-Elmer Corporation assumes no responsibility for any errors that may appear in this document.

The hardware description in this document is intended solely for use in operation, installation, maintenance, or repair of Perkin-Elmer equipment. Use of this document for all other purposes, without prior written approval from Perkin-Elmer is prohibited.

Any approved copy of this manual must include the Perkin-Elmer copyright notice.

The Perkin-Elmer Corporation, Computer Systems Division 2 Crescent Place, Oceanport, New Jersey 07757

Printed in the United States of America

TABLE OF CONTENTS

CHAPTERS (Continued)

CHAPTERS (Continued)

APPENDIXES

FIGURES

FIGURES (Continued)

TABLES

DRAWINGS

INDEX

PREFACE

This manual provides installation and maintenance information for the High Performance Floating-Point Processor (HPFPP). Chapters 1 and 2 present an introduction to the processor and installation requirements including CPU strapping, board locations, and switch positioning. Chapter 3 discusses programming considerations, and Chapter 4 describes the operation of the HPFPP in terms of state transition diagrams and logical algorithms. Chapters 5 and 6 provide a block diagram analysis and a logic analysis. The manual appendices contain information on floating-point arithmetic, guard digits, rounding, arithmetic references, ROM maps, and mnemonics. The manual also contains relevant board schematics and assembly drawings.

Revision 11 includes revisions 9, 10 and the following:

- drawing and circuit changes to indicate capacitor C7 is terminated on the ground bus
- assembly drawing change to indicate a 14 pin IC is located at 08C
- text changes to indicate that the HPFPP is applicable to all Model 3200 series processors

The following manuals provide additional information for the 3220 Processor System:

3220 Processor Installation and Maintenance Manual, Publication Number 29-695

3220 Processor User's Manual, Publication Number 29-693

3220 Microprogramming Reference Manual, Publication Number 29-694

CHAPTER 1
INTRODUCTION

The High Performance Floating-Point Processor (HPFPP) handles the single- and double-precision floating-point operations of the 3210, 3220, or 3230 user instruction repertoire. The HPFPP contains both the single- and double-precision floating-point registers.

The unit is composed of two standard size 381 mm (15") circuit boards, HPFPP-A (35-715) and HPFPP-B (35-716), and uses high speed Schottky TTL logic. The boards plug into the CPU backpanel in two dedicated slots. Communication with the CPU is accomplished over the C bus.

CHAPTER 2
INSTALLATION

2.1 INTRODUCTION

This chapter provides the necessary information to install the High Performance Floating-Point Processor (HPFPP). The HPFPP module consists of two standard 381 mm (15") printed circuit boards, one C bus interface card, and two cables, as follows:

1 each	HPFPP-A	35-715
1 each	HPFPP-B	35-716
1 each	C-Bus Interface	35-735
2 each	Cable	17-234F08

2.2 POWER REQUIREMENTS

The HPFPP requires +5 VDC @ 20 amperes (maximum) for operation. Because of this power requirement, additional power may be needed when configuring a processor system with the HPFPP. For additional information, refer to the appropriate Processor Installation and Maintenance Manual.

2.3 CPU STRAPPING

Strapping is required on the CPU-A board in order to install the HPFPP. Strapping details are provided in the appropriate appendix covering your particular processor.

2.4 SWITCH POSITIONING AND CLOCKS

On installation of the HPFPP, the 8-position DIP switch on the HPFPP-A board, location 16A, should be set to the run mode with nominal clocks (100 ns) regardless of the processor being used. Set switches as follows:

1. OFF
2. ON
3. OFF
4. OFF
5. OFF
6. OFF
7. ON
8. OFF

Refer to the appropriate Processor Installation and Maintenance Manual for clock adjustment information, if necessary.

2.5 TESTING

The following diagnostic is available for testing the HPFPP:

06-231 High Performance Floating-Point Test Program.

CHAPTER 3
PROGRAMMING CONSIDERATIONS

3.1 FUNCTION CODES

The operation to be performed by the HPFPP is determined by RD bits 09, 10, 11, and 25, as well as SFTEN0 and DFTEN0. If RD bit 25 is set, the source operand is supplied from memory via the C bus. Otherwise the source operand must be resident in a floating-point register. SFTEN0 and DFTEN0 indicate whether the operation is single-precision or double-precision. Only one of these lines is active during an HPFPP operation. RD bits 09, 10, and 11 are interpreted as indicated in Table 3-1.

TABLE 3-1 HPFPP FUNCTION CODES

RD091	RD101	RD111	MNEMONIC SP	MNEMONIC DP	OPERATION
0	0	0	RCC	LW	Read Condition Code/Load MS Half
0	0	1	RRE	RRD	Read
0	1	0	LE	LD	Load
0	1	1	CER	CDR	Compare
1	0	0	AER	ADR	Add
1	0	1	SER	SDR	Subtract
1	1	0	MER	MDR	Multiply
1	1	1	DER	DDR	Divide

3.2 UNNORMALIZED LOAD

The signal UNNLD0, when active, causes the HPFPP operation to be completed without post-normalization. This feature may be used to accomplish unnormalized load instructions. The results of arithmetic operations which use unnormalized operands is not defined. Performing an unnormalized load on a single-precision register also clears to zeros the unused portion (bits 32:63) of that register.

3.3 CONDITION CODES

Valid condition codes are generated after each arithmetic instruction as specified by the appropriate Processor User's Manual. The condition code after a Read instruction is unspecified. The condition code is retrieved by the Read Condition Code (RCC) instruction subsequent to the arithmetic operation.

3.4 REGISTER SELECTION

The HPFPP contains 16 floating-point registers. Eight registers store only single-precision data and the other eight store only double-precision data. At user level, each register in each set has an even numbered address in the range X'0' to X'E'.

3.4.1 Destination Register

The floating-point destination register is specified by the YD field of the instruction register. Where two operands are required (add, subtract, multiply, divide, compare), the YD field also designates the A operand source register. This convention is identical to that of the user-level instruction.

The YD field also specifies the floating-point register to be stored on a Read instruction. If a double-precision register is being stored, the least significant bit of the YD field indicates whether the most significant half or the least significant half of the double-precision register is to be stored.

3.4.2 Source Register

The source register for the operation is specified by the YS field of the instruction register. If RD bit 25 is set, the source operand is taken from the C bus and the YS field is ignored.

3.5 SINGLE-PRECISION VS. DOUBLE-PRECISION

The HPFPP is capable of performing floating-point arithmetic in both single-precision and double-precision modes. Operations performed in single-precision are fast, and each operand requires one fullword of storage. Operations performed in double-precision are not always as fast, and each operand requires two fullwords of storage, but the greater precision may make the trade-off worthwhile. It is important to note that the penalties involved in double-precision arithmetic are not always as great as they may seem at first. In most cases, double-precision load, add, subtract, or compare may be performed just as fast in double-precision as in single-precision. If one of the operands comes from memory, there is always a little more overhead involved because two fullwords must be moved over a one fullword bus.

Multiply and divide usually take twice as long in double-precision as in single-precision. The speed of a multiply operation is partially dependent on the strings of ones and zeros in the multiplier. Operations which require multiplying variables by simple constants may be faster if the constant is the multiplier.

Storage space may also be saved through the use of mixed-mode instructions. These instructions allow double-precision data to be rounded and stored as single-precision data. They also permit single-precision data to be expanded to double-precision; however, improper use of these instructions may result in inaccurate answers. See the appropriate Processor User's Manual for descriptions of the mixed-mode instructions.

CHAPTER 4
HPFPP ALGORITHMS

4.1 INTRODUCTION

This section describes the operation of the HPFPP in terms of state transition diagrams and logical algorithms for each HPFPP operation.

Figure 4-1 shows a sample HPFPP state transition diagram for a hypothetical HPFPP operation. All HPFPP operations begin in state 0, the idle state. In the example, the operation begins in state 0 and progresses to state 1. From state 1, the operation either returns to state 0 or goes on to state 4, depending on whether or not an exponent overflow occurs. If it goes on to state 4, it must then proceed to state 0.

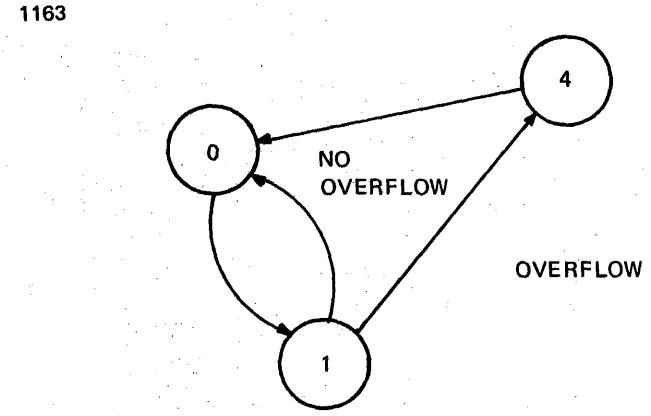


Figure 4-1 Sample State Transition Diagram

The algorithm tables show the HPFPP algorithm sequences by indicating the resulting action based on the key signals involved at each step in the operation. In the example shown in Table 4-1, the actions that may occur in state 0 (ST001) are indicated. If conditions VFLG1 and UFL0 are both true, then bits 00:24 of the A Latch (AL) are forced to ones. Also during state 0, state 0 is reset and state 1 (ST011) is set unconditionally when the clock occurs.

TABLE 4-1 EXAMPLE ALGORITHM

STATE	CONDITION	GATING
ST001	VFLG1•UFLO = 1 unconditional	AL(08:31) <- 1 ST001 <- 0 ST011 <- 1

4.2 READ CONDITION CODE

The Read Condition Code (RCC) function retrieves the condition code resulting from the last HPFPP operation, as shown in Table 4-2. The microprogrammer normally uses this instruction to load the condition code to the PSW after an operation affecting the user status. The RCC instruction is the only HPFPP operation that does not activate the HPFPP clock.

TABLE 4-2 READ CONDITION CODE ALGORITHM

STATE	CONDITION	GATING
ST001	RCC1•SFTEN1 = 1	C280 <- CCC0 C290 <- VCC0 C300 <- GCC0 C310 <- LCC0

4.3 LOAD MOST SIGNIFICANT HALF

The Load Most Significant Half (LW) instruction must precede any double-precision floating-point instruction which uses a source operand from memory. The LW instruction loads the most significant half of the MQ register. It also sets a flag in the HPFPP which indicates that the data loaded by the next floating-point instruction should be directed to the least significant half of the MQ.

Figure 4-2 depicts the state transition diagram for the load most significant half operation. Table 4-3 contains the load most significant half algorithm.

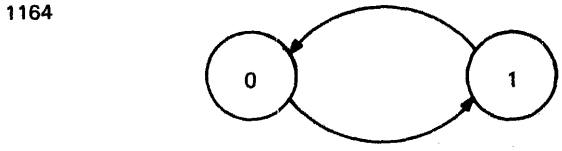


Figure 4-2 Load Most Significant Half State Transition Diagram

TABLE 4-3 LOAD MOST SIGNIFICANT HALF ALGORITHM

STATE	CONDITION	GATING
ST001	LMSH1·DFTEN1 = 1	MQ(00:31) <- C(00:31) ST001 <- 0 ST011 <- 1
ST011	unconditional	LLMSH1 <- 1 ST011 <- 0 ST001 <- 1

4.4 LOAD

The Load operation is used to transfer data from one floating-point register to another, or from memory to a floating-point register. The operation conditionally normalizes a number, depending on the state of UNNLDO. When UNNLDO is active, data is loaded unnormalized into the destination register. This feature is used to implement the Load Multiple user instruction.

Operation of the Load instruction is effected if it is preceded by an LW instruction. These two instructions must be used in conjunction to load a double-precision floating-point register.

Figure 4-3 depicts the state transition diagram for the Load operation. Table 4-4 contains the Load algorithm.

1165

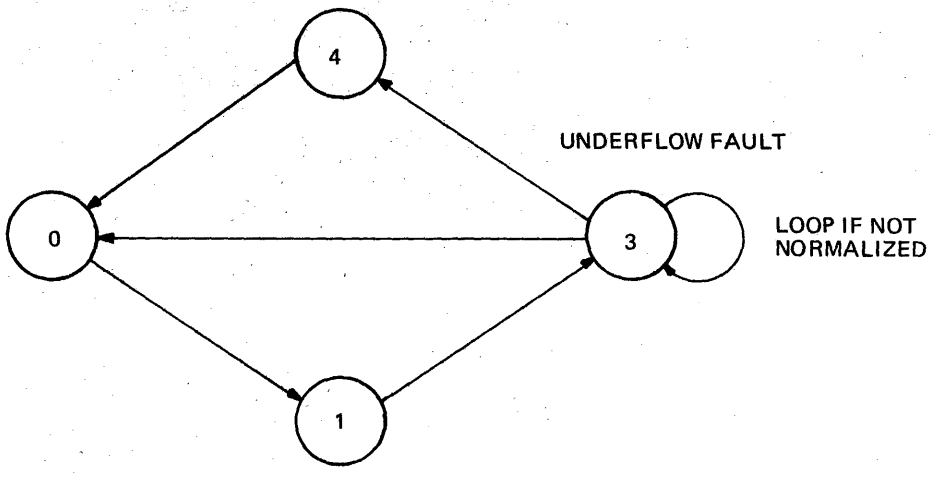


Figure 4-3 Load State Transition Diagram

TABLE 4-4 LOAD ALGORITHM

STATE	CONDITION	GATING
ST001	SFTEN1+DFTEN1 = 1	OPSTOP0 <- 0
	LLMSH1 = 0	MQ(00:31) <- C(00:31) MQ(32:63) <- 0
	LLMSH1 = 1	MQ(32:63) <- C(00:31)
	unconditional	ST001 <- 0 ST011 <- 1
ST011	RR1 = 0	BMX(00:63) <- MQ(00:63)
	RR1 = 1	BMX(00:63) <- BSTK(00:63)
	unconditional	BMX(64:67) <- 0 XL(00:07) <- BMX(00:07) AL(08:63) <- BMX(08:63) ST011 <- 0 ST031 <- 1
ST031	NRLZ1 = 0	AL(08:59) <- AL(12:63) AL(60:63) <- 0 XAL <- XAL-1
	BMX(08:11) = 0	Return to NRLZ1 = 0
	BMX(08:11) ≠ 0	NRLZ1 <- 1
	NRLZ1 = 1	ASTK(00:07) <- XAL(00:07) ASTK(08:31) <- AL(08:31) BSTK(00:07) <- XAL(00:07) BSTK(08:31) <- AL(08:31) ST031 <- 0 ST001 <- 1 OPTSTOP0 <- 1
	NRLZ1·SP0 = 1	ASTK(32:63) <- AL(32:63) BSTK(32:63) <- AL(32:63)
	XAL < 0	UFL1 <- 1
	UFL1·PSW191 = 1	ST031 <- 0 ST041 <- 1 (see Sec. 4.10)

4.5 READ

The Read instruction is used to return the contents of the floating-point registers to the CPU via the C bus. The flags have no meaning after a read instruction.

Figure 4-4 shows the state transition diagram for the Read instruction. The read algorithm is in Table 4-5.

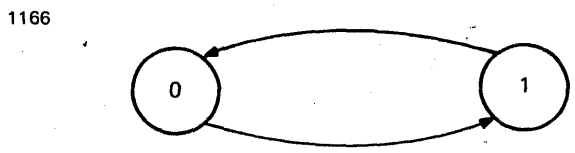


Figure 4-4 Read State Transition Diagram

TABLE 4-5 READ ALGORITHM

STATE	CONDITION	GATING
ST001		The gating and control for State 0 is the same for all algorithms. Refer to Table 4-4.
ST011	SP1=1	C(00:31) <- BSTK(00:31)
	SP0·YD110=1	C(00:31) <- BSTK(00:31)
	SP0·YD111=1	C(00:31) <- BSTK(32:63)
	unconditional	ST011 <- 0 ST001 <- 1

4.6 ADD/SUBTRACT

The HPFPP performs an algebraic add/subtract upon the two operands as determined by the function and the signs of the operands. Since the operands and the result are expressed in sign-magnitude convention, the HPFPP algorithm ensures that the larger of the two operands becomes the minuend for the subtract operation so that a two's complement number is never generated. Figure 4-5 depicts the state transition diagram and Table 4-6 contains the algorithm.

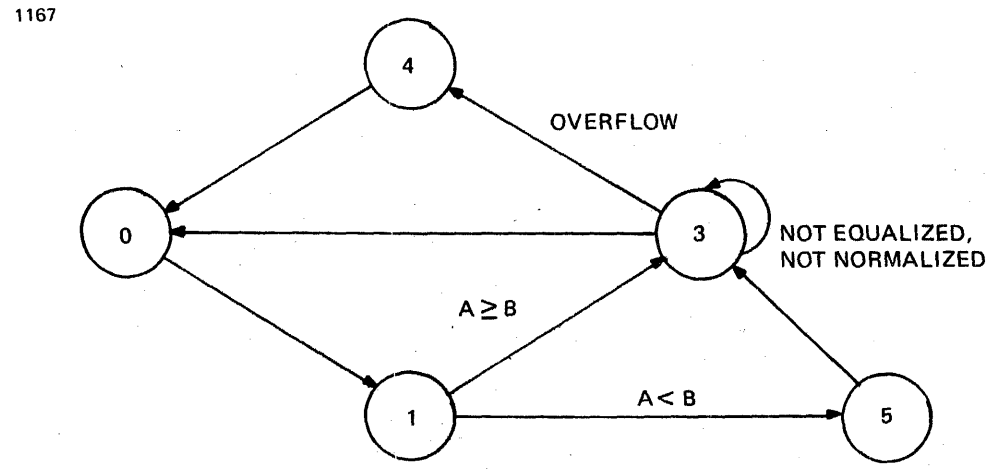


Figure 4-5 Add/Subtract State Transition Diagram

TABLE 4-6 ADD/SUBTRACT ALGORITHM

STATE	CONDITION	GATING
ST001		The gating and control for State 0 is the same for all algorithms. Refer to Table 4-4.
ST011	RR1=0	BMX(00:63) <- MQ(00:63)
	RR1=1	BMX(00:63) <- BSTK(00:63)
	BMX(01:63)>ASTK(01:63)	BGT1 <- 1 ST051 <- 1
	BMX(01:63)≤ASTK(01:63)	ST031 <- 1
	unconditional	XMX(00:07) <- ASTK(00:07) AMX(08:63) <- ASTK(08:63) XL(01:07) <- XMX(01:07)△ BMX(01:07) XCNTR <- XAL(02:07) ST011 <- 0
ST051	unconditional	ASTK(00:63) <- BMX(00:63) ST051 <- 0 ST031 <- 1

TABLE 4-6 ADD/SUBTRACT ALGORITHM (Continued)

STATE	CONDITION	GATING
ST031	VGT1=1	XCRY1 <- 1 AL(08:63) <- ASTK(08:63)
	XCRY1=0	XCNTR <- XCNTR + 1 AL(08:11) <- 0 AL(12:67) <- AL(08:63)
	XCNTR=X'3F'	XCRY1 <- 1
	XCRY1•ACRY0•VGT0=1	AL(08:63) <- ASTK(08:63) + AL (08:63)
	XCRY1=1	ACRY1 <- 1
	ACRY1•SNRLZ0=1	AL(08:63) <- AL(12:67) AL(64:67) <- 0 XL <- XL-1
	XAL<0	UFL1 <- 1 ST031 <- 0
	XAL<0•PSW191=1	ST041 <- (see Sec. 4.10)
	MOFF1=1	AL(12:67) <- AL(08:63) AL(08:11) <- X'1' XAL <- XAL+1
	XAL>X'7F'	VFLG1 <- 1 ST031 <- 0 ST041 <- 1 (see Sec. 4.10)
	SNRLZ1•ACRY1•SP0=1	ASTK(32:63) <- AL(32:63) BSTK(32:63) <- AL(32:63)
	SNRLZ1•ACRY1=1	ASTK(00:07) <- XL(00:07) ASTK(08:31) <- AL(08:31) BSTK(00:07) <- XL(00:07) BSTK(08:31) <- AL(08:31) Round (see Sec. 4.9) ST031 <- 0 ST001 <- 1

State 1 of the algorithm compares the absolute magnitude of the two operands to determine which is the greater. The exponential difference is calculated and stored in the iteration counter (XCNTR). If the B operand is larger, it is written into the A stack during state 5 so that it may be used as the A operand.

If the exponents are different, the mantissa of the smaller number is shifted right until the numbers are properly scaled. If the difference signifies a shift such that the shifted mantissa is zero, the operation is not performed and the greater of the two operands is stored as the answer.

When the number has been equalized (signified by XCRY1=1), the add/subtract is performed. If an add operation results in a mantissa overflow (carry out of the MSB), it is rescaled. If a subtract results in an unnormalized number, it is subsequently normalized. When post scaling has been accomplished, it is stored in the stacks. If the result must be rounded, the operation continues for one more clock and the rounded result is stored in the stacks.

During normalization, the exponent is decremented once for each normalize shift. If this operation causes an exponent underflow and PSW 19 is set, then the HPFPP goes to state 4 (see Section 4.9).

During the correction for a mantissa overflow, +1 is added to the exponent. If this causes an exponent overflow, the HPFPP goes to state 4 (see Section 4.10).

4.7 MULTIPLY

The multiply algorithm uses a technique of skipping over strings of ones or zeros when operating on the multiplier. This allows the multiplication process to be completed in fewer iterations than the number of bits in the multiplier. Figure 4-6 depicts the multiply state transition diagram and Table 4-7 contains the algorithm.

The concept of skipping over zeros in the multiplier is straight forward, since the presence of a zero multiplier bit implies shifting only the partial product as in a conventional algorithm. Therefore, when more than one zero is encountered in the multiplier, the partial product is shifted as many times as there are consecutive zeros. This may be performed in one operation if the capability is provided in the hardware. The HPFPP provides shifts of one, two, three, or four places in one operation.

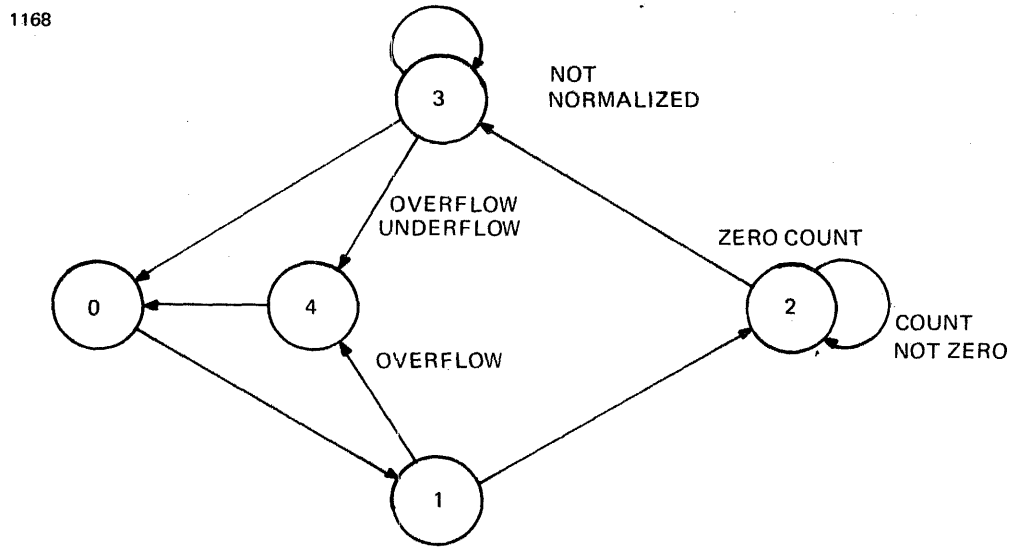


Figure 4-6 Multiply State Transition Diagram

The concept of skipping over strings of ones is less straight forward but may be understood by considering the following examples. For instance, to multiply some number, n, by seven, a conventional multiplication scheme would require three steps to calculate seven times n, as:

$$7n = n + 2n + 4n$$

Another method which requires only two steps for the same problem calculates seven times n, as:

$$7n = -n + 8n$$

When the HPFPP encounters a string of ones in the multiplier, it subtracts the multiplicand from the partial product. It then shifts the new partial product and the multiplier until the next zero bit is encountered in the multiplier. The multiplicand is then added to the shifted partial product.

For example, when the binary string 0000111 is encountered in the multiplier, the HPFPP performs the operation in two steps:

1. subtract and shift 3 places
2. add and shift 4 places

During state 1, the multiplier is loaded into the MQ. The iteration count is loaded into the XCNTR and the exponents of the operands are added. If an exponent overflow occurs, state 4 is entered; otherwise, control is transferred to state 2.

During each step in state 2, the least significant four bits of the multiplier in the MQ are examined. These bits determine whether the partial product should be passed straight through the MAL, or whether it should have the multiplicand added to it or subtracted from it. The result is then shifted, as determined by these bits, and the new partial product is latched into the AL. The MQ is shifted the same amount and the XCNTR is decremented by the same amount.

The operation is complete when the XCNTR is decremented to zero. However, the operation may require a correction cycle if a string of ones is in process. In this case the multiplicand is added to the partial product unshifted in the last period of state 2.

In state 3 the product is normalized and rounded if required. If exponent underflow occurs or occurred during state 1 and the product is not zero, then control may transfer to state 4 (see Section 4.10); otherwise, the product is stored and the HPFPP returns to state 0.

TABLE 4-7 MULTIPLY ALGORITHM

STATE	CONDITION	GATING
ST001		The gating and control for state 0 is the same for all algorithms. Refer to Table 4-4.
ST011	RR1=0	BMX(00:63) <- MQ(00:63)
	RR1=1	BMX(00:63) <- BSTK(00:63)
	unconditional	XL(00:07) <- ASTK(00:07) +BMX(00:07) MQ(08:63) <- BMX(08:63) BAD(01:03) <- AAD(01:03)
	SP1=0	XCNTR <- X'38'
	SP1=1	XCNTR <- X'18'
	unconditional	ST011 <- 0
	JVFLG1=0	ST021 <- 1
	JVFLG1=1	ST041 <- 1 (see Sec. 4.10)
ST021	SP1=0	XMQ(28:31) <- MQ(60:63)
	SP1=1	XMQ(28:31) <- MQ(28:31)
	XCRY1=0	n = 1, 2, 3, or 4 MSC(00:01) <- n XCNTR <- XCNTR-n MAL <- f[AL(08:63),BSTK(08:63)] where f = AL+BSTK, or AL-BSTK, or AL only. AL(08:67) <- MAL shifted right n places
	XCNTR=0	XCRY1 <- 1
	XCRY1=1	ST021 <- 0 ST031 <- 1

TABLE 4-7 MULTIPLY ALGORITHM (Continued)

STATE	CONDITION	GATING
ST031	SNRLZ1=0	AL(08:63) <- AL(12:67) AL(64:67) <- 0 XAL <- XAL-1
	SNRLZ1 SPO=1	ASTK(32:63) <- AL(32:63) BSTK(32:63) <- AL(32:63)
	SNRLZ1=1	ASTK(00:07) <- XAL(00:07) ASTK(08:31) <- AL(08:31) BSTK(00:07) <- XAL(00:07) BSTK(08:31) <- AL(08:31) Round (see Sec. 4.9) ST031 <- 0 ST001 <- 1

4.8 DIVIDE

The division algorithm begins by testing for a nonzero divisor and computing the exponent difference in state 1. The state transition diagram is shown in Figure 4-7, and the divide algorithm is given in Table 4-8.

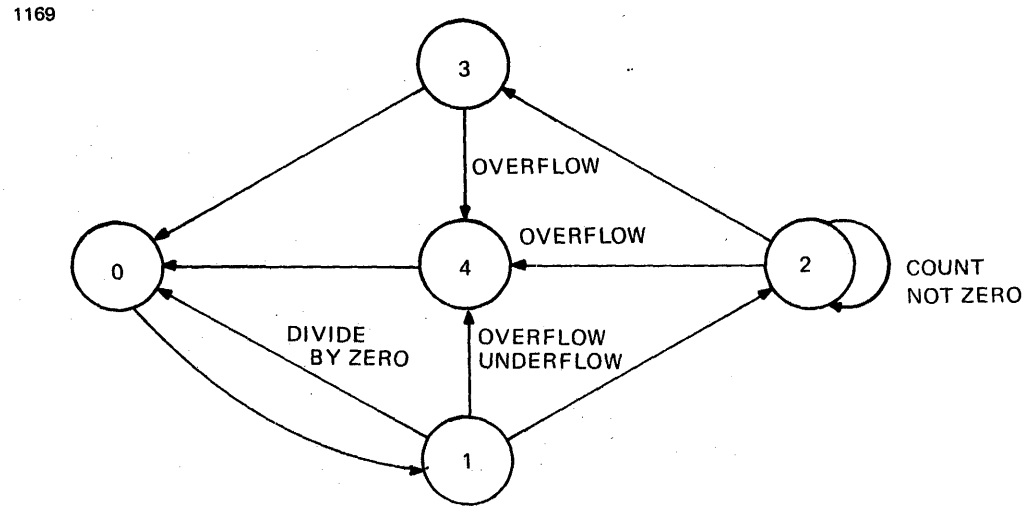


Figure 4-7 Divide State Transition Diagram

The divisor is checked for zero in state 1 by examining the first four bits of the mantissa. If the divisor is not normalized, it is assumed to be zero. The state counter is forced to state 0 and the registers remain unchanged.

The resultant exponent is calculated by subtracting the exponent of the divisor from that of the dividend. If the subtraction results in an overflow, the state is forced to state 4 (see Section 4.10). If an exponent underflow occurs, the division process continues to ensure that the dividend is not zero. If it is zero in state 3, and PSW bit 19 is not set, a zero quotient is forced and the flags are set accordingly. Otherwise, state 4 is entered (see Section 4.10).

During state 1, the divisor is moved from the MQ register to the destination register of the B stack if the operation is not register-to-register. This is necessary as the quotient is formed in the MQ during state 2.

During the first clock period of state 2, the relative magnitude of the divisor and dividend is checked. The algorithm depends on the divisor being scaled larger than the dividend before the division begins. During the second clock period of state 2, the dividend is loaded through the MALU into the AL, shifting one place left (dividend<divisor) or three places right (dividend$\geq$ divisor) to scale the mantissa properly. If the mantissa is shifted right, the exponent difference is incremented by one to compensate for the scaling. During the remaining clock periods of state 2, the remainder is shifted left and compared to the divisor. If the divisor is smaller, it is subtracted from the remainder; the difference becomes the new remainder, and a one is shifted into the least significant bit of the MQ. If the divisor is larger, the remainder is shifted again and a zero is shifted into the MQ. The process continues until the proper number of quotient bits is compiled.

When the quotient is completed in the MQ, the state is forced to state 3. The quotient is transferred from the MQ, rounded if necessary, and stored in the stacks.

4.9 ROUNDING

All results in the HPFPP are rounded in state 3. The result is rounded to the nearest number, which may be represented by the floating-point notation. In the case of a result which is exactly midway between two representable numbers, the result is rounded to the odd one (see Appendix C). Rounding up requires one clock period more than rounding down or rounding to the nearest odd number.

TABLE 4-8 DIVIDE ALGORITHM

STATE	CONDITION	GATING
ST001		The gating and control for state 0 is the same for all algorithms. Refer to Table 4-4.
ST011	SNRLZ1 = 0	ST011 <- 0 ST001 <- 1
	RR0•SP0 = 1	BSTK(32:63) <- MQ(32:63)
	RR0 = 1	BSTK(08-31) <- MQ(08-31)
	unconditional	XL(01:07) <- ASTK(01:07) - BSTK(01:07)
	SP1 = 0	XCNTR <- X'3A'
	SP1 = 1	XCNTR <- X'1A'
	unconditional	ST011 <- 0
	JVFLG1 = 0	ST021 <- 1
	JVFLG1 = 1	ST041 <- 1 (see Sec 4.10)
ST021	FCLK1•DCLK0 = 1	DCLK0 <- 0
	ASTK(08:63)≥BSTK(08:63)	FCLK1 <- 1
	ASTK(08:63)<BSTK(08:63)	FCLK1 <- 0
	FCLK1•DCLK1 = 1	AL(11:63) <- ASTK(08:60) AL(08:10) <- 0 XL <- XL + 1
	JVFLG1 = 1	ST041 <- 1 (see Sec. 4.10)
	FCLK0•DCLK1 = 1	AL(08:63) <- ASTK(09:63) M07 <- ASTK(08) AL(63) <- 0
	FCLK0•DCLK0 = 1	XCNTR <- XCNTR-1 MQ(08:63) <- MQ(09:64)
	AL(08:63)<BSTK(08:63)	AL(08:62) <- AL(09:63) AL(63) <- 0 M07 <- AL(08) MQ(64) <- 0 (SP1 = 0)

TABLE 4-8 DIVIDE ALGORITHM (Continued)

STATE	CONDITION	GATING
		MQ(33) <- 0 (SP1 = 1)
	AL(08:63)>BSTK(08:63)	MAL(08:63) <- AL(08:63) -BSTK(08:63) AL(08:62) <- MAL(09:63) AL(63) <- 0 M07 <- MAL(08) MQ(64) <- 1 (SP1 = 0) MQ(33) <- 1 (SP1 = 1)
	XCNTR = 0 XCNTR = 0	XCRY1 <- 1 XCRY1 <- 1
	XCRY1 = 1	ST021 <- 0 ST031 <- 1
ST031	SP0 = 1	ASTK(32:63) <- MQ(32:63) BSTK(32:63) <- MQ(32:63)
	unconditional	ASTK(00:07) <- XAL(00:07) ASTK(08:31) <- MQ(08:31) BSTK(00:07) <- XAL(00:07) BSTK(08:31) <- MQ(08:31) Round (see Sec. 4.9) ST031 <- 0 ST001 <- 1

4.10 FAULT PROCESSING (State 4)

Whenever a floating-point operation results in exponent overflow, control is transferred to state 4. An exponent overflow is always considered to be an arithmetic fault. In state 4, both the source and destination registers are restored to their original contents and the flags are set to indicate the overflow condition.

An exponent underflow may or may not be considered a fault, depending on the state of bit 19 of the PSW. If bit 19 is set, an underflow is considered to be an arithmetic fault and the procedure is the same as for overflow. If bit 19 is not set and an underflow occurs, the result is replaced by a true zero as an approximation to the real result. In either case, the flags are set to indicate the underflow condition.

Once fault processing is completed in state 4, the HPFPP returns immediately to state 0.

CHAPTER 5
BLOCK DIAGRAM ANALYSIS

5.1 INTRODUCTION

Figure 5-1 shows the HPFPP block diagram. The block diagram is intended to show the major paths of data flow. It does not detail any of the control logic.

1170-1

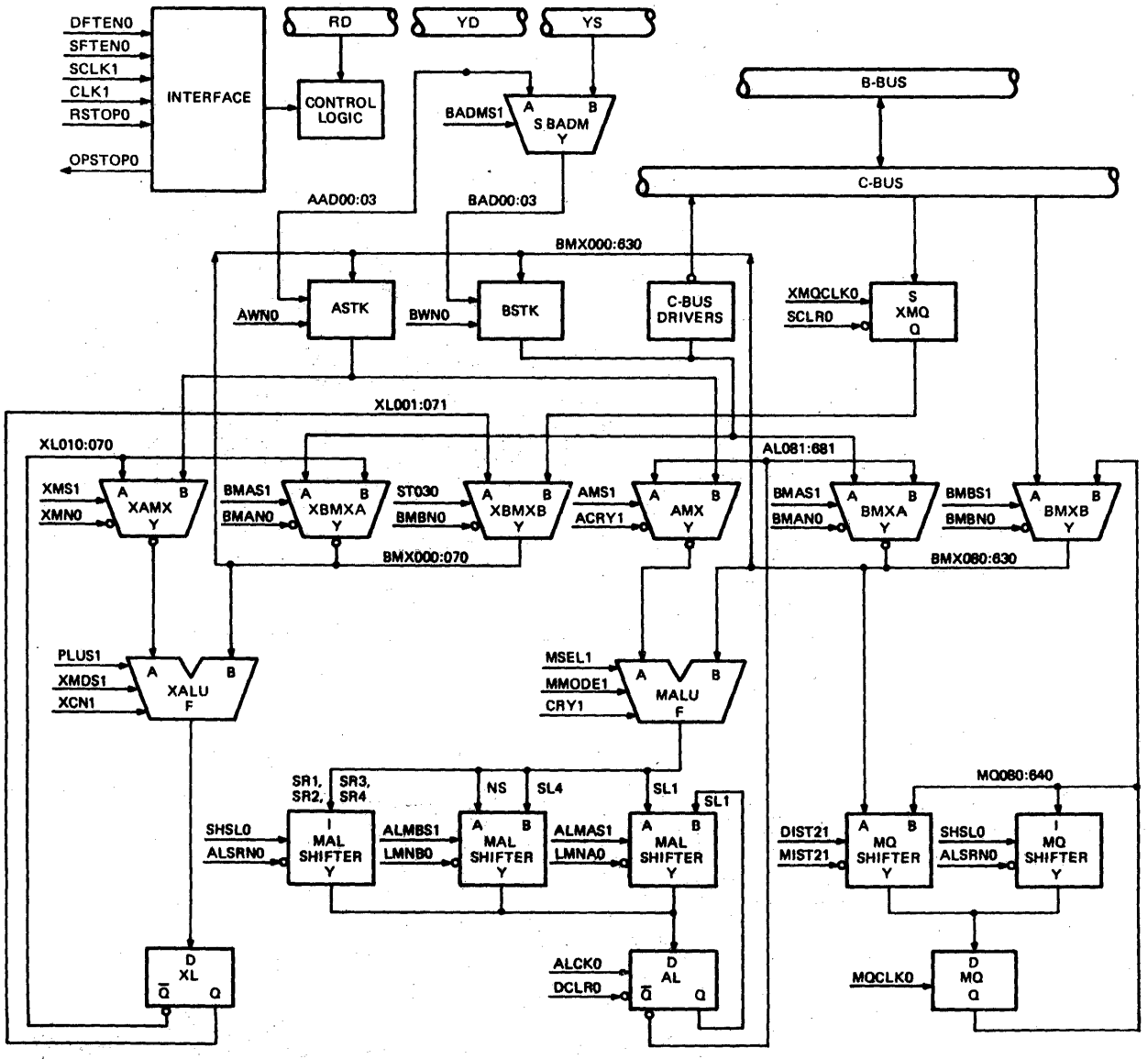


Figure 5-1 HPFPP Block Diagram

5.2 A STACK, B STACK

The A Stack (ASTK) and the B Stack (BSTK) are two independent sets of registers which operate as a single two-port register set. During register-to-register operations, the A operand and the B operand may be accessed simultaneously. There are 16 floating-point registers, each 64 bits long. Eight of the registers are used for single-precision data and eight for double-precision data. The least significant 32 bits of each of the single-precision registers contain zeros. At the completion of an operation, the results are written into the destination register of both register sets.

5.3 B ADDRESS MULTIPLEXOR

The B Address Multiplexor (BADM) allows the B register stack to be addressed by either the YD field or the YS field of the instruction register. This allows the result of an operation to be written into the destination register of both stacks.

5.4 A MULTIPLEXOR

The A Multiplexor (AMX) selects the A operand to be gated to the Mantissa ALU (MALU). The sources which can be gated are the ASTK and the AL register. The AMX can also be disabled to force the A bus to all zeros.

5.5 B MULTIPLEXOR A, B MULTIPLEXOR B

B Multiplexor A (BMXA) and B Multiplexor B (BMXB) select the B operand to be gated onto the B bus. The sources which can be gated are the BSTK, the AL register, the MQ register, and the C bus. Only one multiplexor at a time is enabled so it operates as a single 4-to-1 multiplexor. Both multiplexors may be disabled, however, in order to force the B bus to all zeros.

5.6 MANTISSA ALU

The Mantissa ALU (MALU) provides the add and subtract functions and the logical gating (A only, B only) required by the HPFPP algorithms. It is a 56-bit ALU using two-stage carry look-ahead.

5.7 MAL SHIFTER

The MAL shifter consists of two 2-to-1 multiplexors and a 4-way shifter, all feeding a tri-state bus. Only one component of the MAL shifter is enabled at any one time. Its purpose is to shift the MAL data which is to be latched into the AL register. MAL data (unshifted or shifted) may be passed right one, two, three, or four bits or shifted left one or four bits. Also, AL data may be shifted left one bit.

5.8 ARITHMETIC LATCH

The Arithmetic Latch (AL) stores the intermediate results of an operation. Data from the AL may be routed back to the MALU as the A operand via the AMX, or as the B operand via the BMXA. When the AL contains the results of an operation, the data is routed through the BMXA to the ASTK and BSTK.

5.9 MULTIPLIER/QUOTIENT REGISTER

The Multiplier/Quotient (MQ) register is a 56-bit scratch register. Any data that comes from the C bus gets latched into the MQ first. The MQ holds the multiplier during multiply operations and accumulates the quotient during divide operations.

5.10 MQ SHIFTER

The MQ shifter consists of a 2-to-1 multiplexor and a 4-way shifter. It allows the MQ register to be loaded from the B bus; or MQ data may be shifted left one bit or shifted right one, two, three, or four bits.

5.11 EXPONENT ALU AND REGISTERS

The Exponent ALU (XALU) performs any necessary exponent arithmetic. The Exponent Latch (XL) register is the exponential counterpart to the AL. The exponent multiplexors are also counterparts to the mantissa multiplexors. The A operand of the XALU is provided via the XAMX and may come from the ASTK or the XL. The B operand is provided by the XBMXA and the XBMXB and may come from the BSTK, the XL, or the exponent scratch register (XMQ). The XMQ contains the data latched from the C bus on a memory reference operation.

CHAPTER 6
LOGIC ANALYSIS

6.1 INTRODUCTION

Many of the functions of the HPFPP are split between the two boards. Therefore, the detailed analysis is presented by function rather than by analysis of individual circuits on each board.

In the following analysis, reference is made to various circuits by their location in the schematic. For example, a reference to A-6N7 refers to coordinate N7 on Sheet 6 of the HPFPP-A schematics. Most signals are referred to by their mnemonic; however, not all signals have an assigned mnemonic. These signals are referenced by the output pin number of the gate that generates the signal. For example, 05W09 refers to pin 09 of the package located at coordinate 05W on the circuit board.

6.2 INTERFACE (HPFPP-A, Sheet 2)

The interface section of the HPFPP controls the start-up and shut-down sequences of the processor. The HPFPP has its own internal clock system which is separate from that of the CPU. While the HPFPP is active, it stops the system clocks in the CPU and starts its own clocks. When the floating-point operation is completed, the HPFPP clocks are stopped and the CPU clocks are released. This start-up and shut-down sequence is the same for all of the HPFPP functions except Read Condition Code (RCC). The RCC function does not stop CPU clocks and inhibits the start of the HPFPP clocks.

6.2.1 HPFPP Start-Up Sequence

The HPFPP is activated whenever a floating-point microinstruction is decoded (see Figure 6-1). The floating-point microinstruction is decoded on the falling edge of the CPU system clock, CLK1. This causes either SFTEN0 or DFTEN0 to become active, depending on whether the operation is single- or double-precision. When either one of these signals becomes active, the HPFPP is selected and SEL1 goes high. This, in turn, causes OPSTOP0 to go low and inhibit the CPU clocks (except CLK1 and SCLK1).

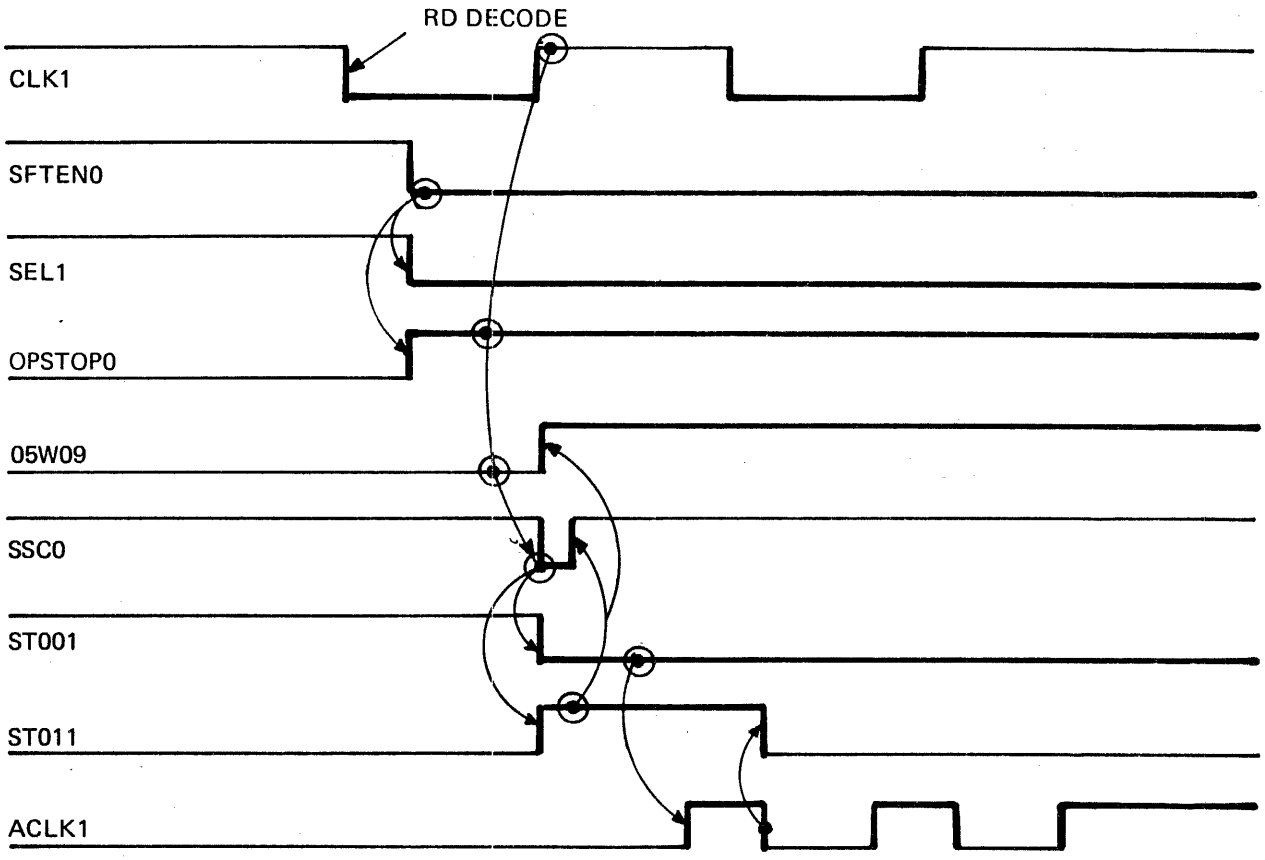


Figure 6-1 HPFPP Start-Up Timing Diagram

When CLK1 again becomes high, SSC0 (Start System Clocks) goes low. SSC0 asynchronously resets state 0 (ST001) and sets state 1 (ST011). State 1 going high immediately sets the flip-flop at 05W, which turns off SSC0. State 0 going low enables the delay line oscillator which generates the HPFPP clocks. Also, if the operation is a memory reference operation, then state 0 going low clocks the data (which by that time has propogated in from the C bus) into the MQ register.

The first HPFPP clock unconditionally resets state 1 and sets one of the other states, depending on the operation being performed. If the operation is complete and the next state set is state 0, then the HPFPP goes into the shut-down procedure.

6.2.2 HPFPP Shut-Down Sequence

The shut-down sequence is initiated when the HPFPP returns to state 0. The purpose of the sequence is to release OPSTOP0 when CLK1 is low so that the CPU clocks are not restarted with a short cycle. (See Figure 6-2.)

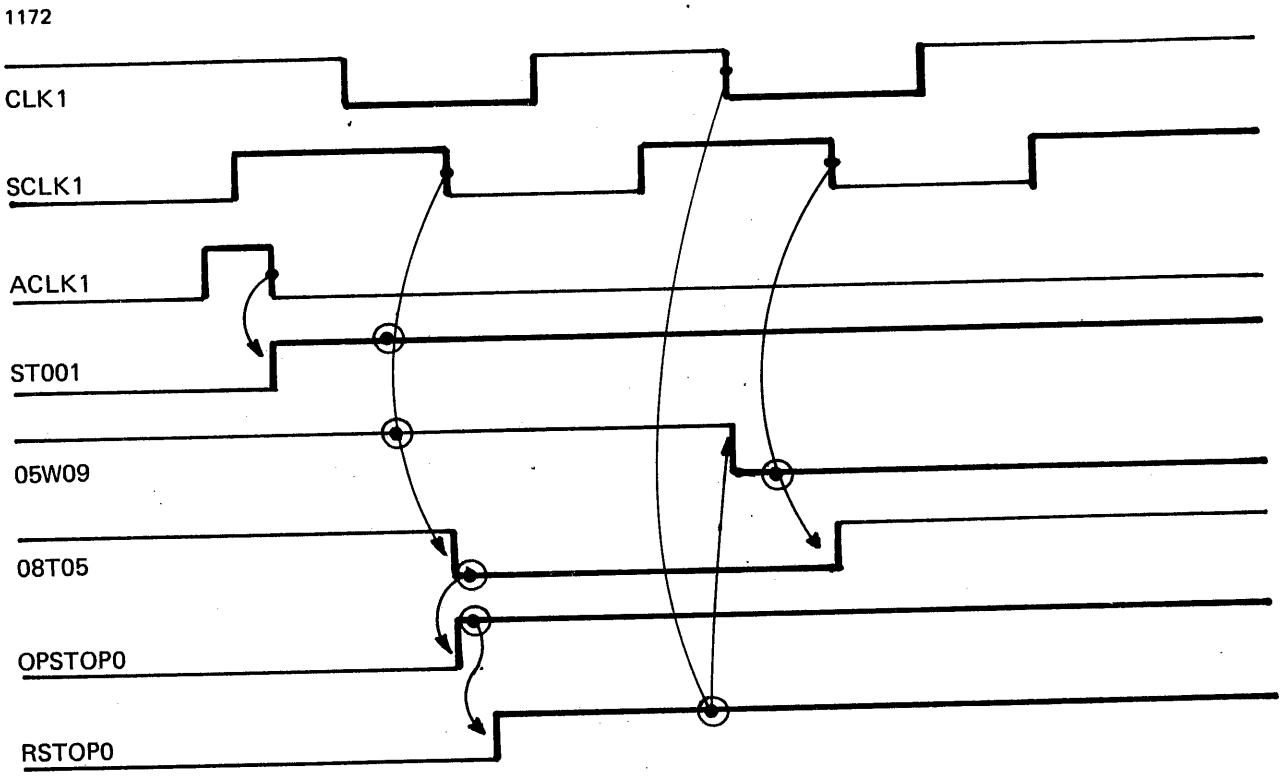


Figure 6-2 HPFPP Shut-Down Timing Diagram

The setting of state 0 stops the oscillator and inhibits any further HPFPP clocks. It also causes the flip-flop at 08T to reset on the next falling edge of SCLK1. When 08T05 goes low, OPSTOP0 is released; this, in turn, releases RSTOP0. The next falling edge of CLK1 resets the flip-flop at 05W and also decodes the next CPU microinstruction. If this is another floating-point operation (other than RCC), the start-up sequence begins again. However, OPSTOP0 does not become active until the flip-flop at 08T is reset on the falling edge of SCLK1. If the next instruction is not floating-point, or if it is an RCC, then SEL1 goes inactive and OPSTOP0 becomes active when the flip-flop at 08T is reset on the falling edge of SCLK1.

6.3 CLOCK (HPFPP-A, Sheet 4)

The HPFPP clock is generated by a gated delay line oscillator. The oscillator is active whenever the HPFPP is in a state other than state 0. The clock is user-programmable through a set of switches at location 16A on HPFPP-A (A-4E5). Switches three, four, five, and six are not used. The settings for the other switches are shown in Table 6-1. The HPFPP is designed to run at a nominal clock speed of 100 ns. The fast clock speed (95 ns) is used for testing purposes and does not necessarily enable floating-point instructions to execute faster. There is also provision for executing HPFPP clocks in single-step mode, or for bypassing internal clocks completely and running the HPFPP with an externally generated clock.

TABLE 6-1 HPFPP CLOCK SWITCH SETTINGS

CLOCK	SWITCH NUMBER 1	2	7	8
Fast clock	ON	OFF	ON	OFF
Nominal clock	OFF	ON	ON	OFF
Single step	OFF	ON	OFF	OFF
External clock	X	X	X	ON

A trimming capacitor (A-4M3) is used to fine tune the period of the clocks. This adjustment is made at the factory and should not be changed during normal operation.

The HPFPP operates in two modes: normal mode and slow mode. The clock period is 100 ns in normal mode and 125 ns in slow mode. A slow-mode clock cycle is initiated when SLWCLK0 (A-6N3, A-4B8) becomes active. The purpose of the slow clock is to provide for stack settling time during the clock period following a period in which the register stacks were accessed. If the divisor is in the MQ register, the registers may be written into during state 5 in the add/subtract algorithms, and state 1 in the divide algorithm. The registers are also accessed when the BSTK address is changed to select the multiplicand when entering state 2 in the multiply algorithm if the multiplier came from the BSTK in state 1 (register-to-register). The slow clock is also used to compensate for the particularly slow data paths associated with normalization.

6.4 STATE REGISTERS (HPFPP-A, Sheet 3)

The state of the HPFPP is controlled by six flip-flops, each uniquely associated with an HPFPP state. During state 0 (the idle state) the clock is inhibited. An operation begins when state 0 is reset and the clock is activated. The clock logically toggles the state registers, as determined by the HPFPP algorithms, until state 0 again becomes active, disabling the clock. No more than one state ever becomes active at any time.

Each state performs a specific function or set of functions depending on the operation being performed. Table 6-2 shows each HPFPP state and lists the functions that may be performed in that state. Table 6-3 indicates all the possible state transitions and lists the conditions necessary for each transition to occur.

TABLE 6-2 FUNCTIONS OF HPFPP STATES

STATE	FUNCTION
0	Idle, ready to accept next operation
1	Perform Load Most Significant Half Perform Compare Gate contents of BSTK to C-Bus for Read Compare exponents for Add/Subtract Add exponents for Multiply Subtract exponents for Divide
2	Perform Multiply or Divide
3	Equalize operands for Add/Subtract Perform Add/Subtract Normalize result from Load, Add/Subtract, or Multiply Write result to register
4	Restore registers if exponent overflow Restore registers if exponent underflow fault
5	Store B operand in ASTK destination for Add/Subtract when B>A

TABLE 6-3 HPFPP STATE TRANSITION CHART

TRANSITION	CONDITIONS
ST001 -> ST011	Unconditional (asynchronous)
ST011 -> ST001	Divide by zero Load Most Significant Half Compare
ST011 -> ST021	Multiply or Divide and divisor normalized, and no exponent overflow or underflow
ST011 -> ST031	Load Add or Subtract and B<A
ST011 -> ST041	Exponent overflow or underflow on Multiply or Divide
ST011 -> ST051	Add or Subtract and B>A
ST021 -> ST031	Multiply or Divide and XCNTR=0
ST021 -> ST041	Exponent overflow on divide when rescaling mantissa
ST031 -> ST001	Result written to register
ST031 -> ST041	Exponent overflow while correcting mantissa overflow after add Exponent overflow during normalization
ST041 -> ST001	Contents of stacks restored on arithmetic fault
ST051 -> ST031	Unconditional

6.5 FUNCTION SELECT AND DECODING (A-5C3, B-2H6)

The operation to be performed by the HPFPP is determined by RD bits 09, 10, and 11 of the current microinstruction. These bits are decoded by a 3-line to 8-line decoder to select the function, as indicated in Table 3-1.

6.6 EXPONENT LOGIC (HPFPP-A, Sheets 9, 10, 11, 12)

The exponent logic is composed of the exponent multiplexor 10D (A-11H6) and 15D (A-12H6); the exponent ALUs 11C (A-11L6) and 14C (A-12L6); the exponent latches 12C (A-11M3) and 13C (A-12M3); some control logic (A-9D3); and the underflow/overflow detection logic (Sheet 10).

The exponent multiplexors gate either ASTK01:07 or XL01:07 to the A input of the exponent ALU (XALU). The ASTK bits are gated through the exponent multiplexor for the purpose of exponent loading or arithmetic. The XL is gated through the multiplexor for the purpose of recirculating, incrementing, or decrementing exponents. The ASTK is selected by the multiplexor during state 1 or during an add/subtract until normalization begins. During a load operation, the exponent multiplexor is disabled in state 1 so that no exponent arithmetic takes place.

Control of the XALU is performed by the signals XMDS1 (A-9G3) and PLUS1 (A-9G4) according to Table 6-4. The exponent is incremented or decremented by XCN1 (A-9G2).

TABLE 6-4 XALU MODE SELECT TABLE

XALU SELECT	STATE	A	S	M	D	C	L
A+B=HLLH A-B=LHHL A,A-1=LLLL A,A+1=HHHH	0	X	X	X	X	X	X
	1	A-B	A-B	A+B	A-B	A-B	A+B
	2	X	X	A	A,A+1	X	X
	3	A,A+1	A,A-1	A,A-1	A	X	A,A-1
	4	A	A	A	A	X	A
	5	A	A	X	X	X	X

During the exponent arithmetic for multiply or divide, the excess 64 operation is performed by an XOR gate on the most significant bit of the XALU by complementing the result of the operation. Signal MDNRLZ1 is inactive for all other operations which gate the XALU bit to the XL unchanged.

Exponent underflow is detected by the AOI gate at 14E (A-10G3) and the condition is latched in the flip-flop at 09E (A-10J3). Exponent overflow is detected by the AOI gate at 13E (A-10G5) and the condition is latched in the flip-flop at 09E (A-10J5).

A special case of overflow is detected by the gate at 16E (A-10D6). Multiplication of two numbers may seem to result in exponent overflow, but post-normalization of the mantissa brings the exponent back into the acceptable range. For example, 7F100000 multiplied by 41100000 appears at first to overflow; but once the product mantissa is normalized, it is obvious that this is not true. The flip-flop at 12B (A-10F8) latches the special case and inhibits overflow detection until after the mantissa multiplication and post-normalization have been completed.

6.7 XCNTR CIRCUIT (HPFPP-B, Sheet 5)

The XCNTR provides the iteration counts for the HPFPP. In add/subtract, the XCNTR is loaded with the exponent difference in state 1 and incremented or decremented in state 3 as the operands are equalized. In multiply or divide, the XCNTR is loaded in state 1 with a constant and decremented in state 2.

The XCNTR circuit consists of four main elements: control logic, an adder, a multiplexor, and a register.

The control logic determines whether the current count is to be incremented or decremented and by how much. The current state of the counter is passed through the adder along with BGT1, XCTA020, and XCTA010. The result, changed by the amount indicated in Table 6-5, goes through the multiplexor and is loaded into the XCNTR register.

TABLE 6-5 XCNTR INCREMENT/DECREMENT COUNT

COUNT	BGT1	XCTA020	XCTA010	OPERATION
-1	0	0	0	Add/subtract (+ diff.) Multiply, Divide
-2	0	0	1	Multiply
-3	0	1	0	Multiply
-4	0	1	1	Multiply
+1	1	1	0	Add/Subtract (- diff.)

The multiplexor allows the register to be loaded in state 1 with either the exponent difference (add/subtract) or a predetermined iteration count (multiply or divide). The iteration count (see Table 6-6) is dependent on whether the operation being performed is single- or double-precision.

TABLE 6-6 ITERATION CONSTANTS FOR MULTIPLY AND DIVIDE

OPERATION	ITERATION COUNT SP	DP
Multiply	X'18'	X'38'
Divide	X'1A'	X'3A'

The XCRY flip-flop monitors the output from the XCNTR multiplexor and sets when the register count goes to zero. This, in turn, disables the clock to the counter. The XCRY flip-flop is forced set in add/subtract on the signal VGT1 (exponent difference very great) so that the equalization need not be carried out if the exponent difference is greater than 7 in single-precision or 14 in double-precision. When VGT1 is active, the larger operand is gated to the AL unchanged and the add/subtract operation is ignored.

6.8 MULTIPLY CONTROLLER (HPFPP-B, Sheet 4)

The multiply control circuit controls the multiplication algorithm during state 2 of the operation. It looks at the least significant four bits of the MQ register and MSUM1 to determine the shift count (shift right one, two, three, or four bits) and the MALU function (add, subtract, or A only).

The least significant four bits of the MQ may be bits 28 to 31 or bits 60 to 63, depending on whether the operation is single- or double-precision. The selection is made by the multiplexor at 16B (B-4G5). The bits are latched into the register at 16A (B-4J5). The output from the register, XMQ28:31, and MSUM1 are decoded to form the multiply shift code, MSC0:1. MSUM1 indicates whether the last active function performed in the algorithm was an add or a subtract. Signals XMQ30 and XMQ31 are passed to the HPFPP-A to control the MALU functions. The decoding of the XMQ bits and MSUM1, and the actions taken are detailed in Table 6-7.

TABLE 6-7 MULTIPLY ALGORITHM CONTROL

XMQ28:31	MSUM1	OPERATION	SHIFT
0000	0	NOP	4
	1	ADD	4
0001	0	ADD	4
	1	NOP	1
0010	0	NOP	1
	1	ADD	1
0011	0	SUBT	2
	1	NOP	2
0100	0	NOP	2
	1	ADD	2
0101	0	ADD	2
	1	NOP	1
0110	0	NOP	1
	1	ADD	1
0111	0	SUBT	3
	1	NOP	3
1000	0	NOP	3
	1	ADD	3
1001	0	ADD	3
	1	NOP	1
1010	0	NOP	1
	1	ADD	1
1011	0	SUBT	2
	1	NOP	2
1100	0	NOP	2
	1	ADD	2
1101	0	ADD	2
	1	NOP	1
1110	0	NOP	1
	1	ADD	1
1111	0	SUBT	4
	1	NOP	4

6.9 MALU CONTROL (HPFPP-A, Sheet 7)

Data movement within the HPFPP is centered around the mantissa ALU. This element provides much of the gating and all of the arithmetic operations required by the algorithms. The four functions performed by the MALU and the signals that control them are:

AONLY: The A bus is gated through the MALU unchanged.
BONLY: The B bus is gated through the MALU unchanged.
APLSB: The sum of the A bus and the B bus is produced.
AMINB: The difference between the A bus and the B bus is produced.

These four signals are then decoded into the function select lines (MS00:03) and the mode select (MMODE1).

6.10 MAL SHIFTER AND A LATCH

The purpose of the A Latch is to retain intermediate results of HPFPP operations. It is loaded through the MAL shifter which consists of the ALMA and ALMB multiplexors and the 4-way shifter. The AL is loaded from the MAL shifter with each clock except during state 1 of multiply.

ALMA: The ALMA multiplexor is used exclusively for divide and can shift either the AL or the MAL one bit to the left. If the divide algorithm requires the difference of the operands shifted, the MAL is selected. If the partial remainder is to be shifted, the AL is selected.

ALMB: The ALMB multiplexor provides shifts of zero and left four bits. The zero shift is used for loads, add/subtract, etc., where the MAL must be kept unscaled. The shift left four is used to normalize the mantissa for a load or after a subtract or multiply.

4-Way Shifter: This shifter can shift the MAL to the right by one, two, three, or four bits. These shifts are required by the multiply algorithm. In addition, the shift right four is used for equalizing mantissas in add/subtract, and in rescaling the AL after mantissa overflow in add.

6.11 MQ AND MQ SHIFTER

The Multiplier/Quotient (MQ) register is named for the function it serves in the algorithms. During multiply operations, the multiplier is contained in the MQ register and examined, four bits at a time, by the multiply controller (see Section 6.7). In the divide algorithm, the quotient is formed in the MQ by shifting the current quotient bit into MQ33 (single-precision) or MQ64 (double-precision). The MQ also provides intermediate storage for memory reference operations by latching the contents of the C bus on the transition to state 1.

The MQ 4-way shifter is used to shift the MQ in the multiply algorithm to the right by one, two, three, or four bits. The MQ multiplexor can load the MQ from the B bus, or provide the shift left one required during a divide.

6.12 AMX CIRCUIT

The AMX controls the A input to the MALU. It can gate either the AL or the ASTK to the MALU or can be disabled to force the A bus to zeros. The AMX is disabled by ACRY1, which becomes active during state 3 after all operations, except storing and rounding, have been completed. Disabling the AMX on ACRY1 allows the MAL to be in APLSB mode so that AMX310 or AMX630 can be forced active to achieve rounding.

The AMX is switched by AMXS1 (A-8F6). The ASTK is gated onto the A bus, except during multiply or during divide after the dividend has been gated to the AL. In these cases, the AL is gated onto the A bus to be iteratively added to or subtracted from the B bus.

6.13 BMXA, BMXB CIRCUITS

The BMXA and the BMXB gate data onto the B bus. The B bus supplies data to the MQ register, the ASTK, the BSTK, and the B input to the XALU and the MALU.

Selection of either BMXA or BMXB is controlled by the BMXN flip-flop (A-8E3, B-401). This flip-flop is initialized in state 0 to select either BMXB, if the B operand is to be from the B bus (memory reference operation), or BMXA, if the B operand is in a floating-point (register-to-register operation). BMXA is enabled at all other times except during the first clock of state 3 during a divide operation (to retrieve the quotient from the MQ) and the first clock after state 1 during a memory reference add/subtract operation. If an operation results in true zero, then both sides of the flip-flop are forced high. This disables both multiplexors, and the B bus is forced to all zeros, which are then written into the ASTK and BSTK as the result of the operation. This also occurs if the operation results in an underflow and the PSW indicates that the underflow is not to be treated as an arithmetic fault.

6.14 ASTK, BSTK CIRCUITS

The ASTK and BSTK are treated as a two-part register array, allowing simultaneous access of operands from different registers. They are always updated with identical data upon completion of each operation to ensure integrity.

The ASTK is addressed by the AAD01:03 lines which are decoded from the YD field of the instruction register. The BSTK is addressed by the BAD01:03 lines which may be decoded from either the YD field or the YS field of the instruction register. The stacks are broken into two sets of eight registers by using the single/double-precision bit (SP1) to form AAD00 and BAD00.

Selection of the BSTK address is performed by the B address multiplexor (BADM, B-2C7) and controlled by BADMS1 (B-3F9). The B address becomes the same as the A address under the following conditions: when the result of an operation is being written into the destination register; during a read operation; during a memory reference operation; during state 2 of a multiply operation; or during state 3 of an add/subtract operation in which the B operand is greater than the A operand.

The write pulses for the two register sets are generated by gating the HPFPP system clock (A-9F7). If the rounded-up result is required, the write pulses are delayed for one clock while rounding is performed.

6.15 CONDITION CODE AND SIGN LOGIC (HPFPP-A, Sheet 9)

In state 1, the register at 10K (A-9K6) latches the sign of the two operands, the relative magnitude of the two (BGT1), and the signal, AEQBC0, which indicates whether the two operands are equal. The output from this latch is used by the sign ROM at 09S (A-9M6) to generate the resultant sign for any HPFPP operation. The ROM also provides the condition code to the condition code multiplexor (10T, A-9M3) for compare operations. The condition code for operations other than compare is obtained from the operation status upon completion of the operation, and latched in the register at 11D (A-9K3). If the last operation was not a compare, this register provides the condition code to the condition code multiplexor.

6.16 GUARD DIGITS AND ROUNDING LOGIC (HPFPP-B, Sheet 14)

The HPFPP employs guard digits to prevent loss of accuracy due to normalization. The HPFPP uses two guard digits and a sticky bit to maintain the greatest accuracy attainable within the limits of its precision (see Appendix B).

In single-precision, the first and second guard digits are actually the first two digits of the double-precision portion of the HPFPP. Bit 32 of the resultant mantissa is used to round, and bits 08 to 31 are retained. In double-precision, the guard digits extend both the MALU and the AL. Bit 64 of the resultant mantissa is used in rounding and bits 08 to 63 are retained.

The first double-precision guard digit is implemented the same as any other digit in the mantissa. Only the most significant bit of the second guard digit is implemented. This bit is used only to perform rounding when post-normalization positions it as the first guard digit. When it is necessary to normalize beyond one digit, the second and subsequent guard digits are zero (see Appendix B).

The sticky bit monitors data that is shifted out of the AL to the right. This can occur during equalization in add/subtract, or during multiply. If any data other than zero is shifted out, then the sticky bit sets and remains so. It indicates that somewhere out to the right there exists data which was lost and can no longer be accessed. If the operation is a subtract, the lost data is subtracted from zero and a borrow is generated.

The result of an HPFPP operation is always rounded up at the end of an operation. The rounded result is retained, however, only if the rounding bit (bit 32 in single-precision and bit 64 in double-precision) is set and JAM0 is not active. JAM0 becomes active if the rounding bit is set, the remaining bits of the guard digits are zeros, and the sticky bit is not set. In this case, the least significant bit of the result is forced to a one to implement the R*-rounding technique (see Appendix C).

APPENDIX A
FLOATING-POINT ARITHMETIC

INTRODUCTION

Floating-point is a means of representing a quantity in any numbering system. For example, the decimal number 123 (base = 10), can be represented in the following forms:

$$\begin{aligned} 123.0 &\times 10^0 \\ 1.23 &\times 10^2 \\ 0.123 &\times 10^3 \\ 0.0123 &\times 10^4 \end{aligned}$$

In this example, the decimal point moved; this is called a floating-point. In actual floating-point representation, the significant digits are always fractional and are collectively referred to as fractions. The power to which the base number is raised is called the exponent. For example, in the number .45678 x 10^2, 45678 is the fraction and 2 is the exponent. Both the fraction and the exponent can be signed. In a floating-point representation such as:

(sign of fraction) (exponent) (fraction)

the following representation applies:

Number	Floating-point			
+32.94	= +.3294 x 10^2	+	+2	3294
-23760000.0	= -.2376 x 10^8	-	+8	2376
+0.000059	= +.59 x 10^{-4}	+	-4	59
-0.0000000092073	= -.92073 x 10^{-8}	-	-8	92073

Large or small numbers can be easily expressed in floating-point, making it ideally suitable for scientific computation. Note the compactness of floating-point notation in the previous examples.

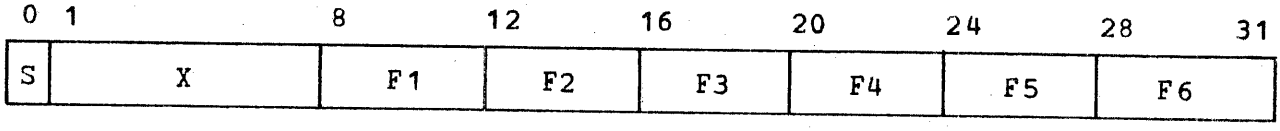
Floating-point representation in the processor is similar to the previous representation. The differences are:

1. A hexadecimal, instead of decimal, numbering system is used.

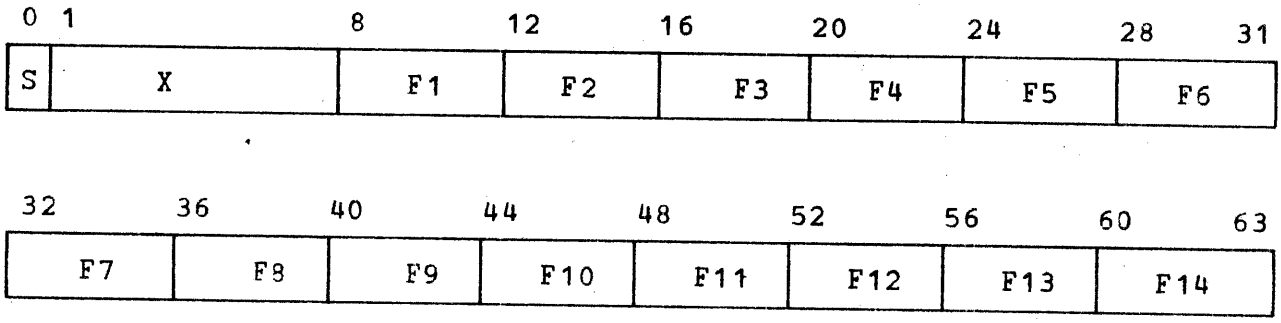
2. The physical size of the number is limited, therefore the magnitude and precision are limited.

DATA FORMATS

Floating-point numbers occur in one of two formats: single-precision and double-precision. The single-precision format requires a fullword (32 bits). When such a value is contained in memory, it must exist on a fullword address boundary. The sign (S), exponent (X), and fraction (consisting of the digits F1, F2, F3, F4, F5, and F6) fields are designated as follows:

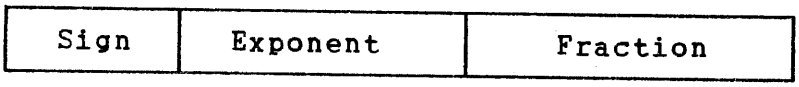


The double-precision format requires a doubleword (64 bits). When two general registers hold a double-precision value, an even/odd pair of general registers must be used. The even-numbered register contains the most significant 32 bits, and the next sequential odd register contains the least significant 32 bits. The sign (S), exponent (X), and fraction (consisting of digits F1 through F14) fields are designated as follows:



FLOATING-POINT NUMBER

In the processor, a floating-point number is represented in the following form:



Sign — The most significant bit of a floating-point number is the sign bit. The sign bit is zero for positive numbers and one for negative numbers. The floating-point value of zero always has a positive sign.

Exponent

The 7-bit field, bits 1:7, is designated as the exponent field. The exponent is expressed in excess-64 notation. The number in this field contains the true value of the exponent plus X'40' (decimal 64). This helps to represent very small magnitudes between 0 and 1. Some of the exponent values are as follows:

Exponent in excess-64 notation	True exponent in hexadecimal	True exponent in decimal	Multiply fraction by
00	-40	-64	16^{-40}
3F	-1	-1	16^{-1}
40	0	0	16^{0}
41	1	1	16^{1}
7F	3F	63	16^{3F}

The exponent field for true zero is always 00.

Fraction

The fraction field is 6 hexadecimal digits for single-precision floating-point numbers (thus limiting the precision), and 14 hexadecimal digits for double-precision floating-point numbers. As in any other fraction, the floating-point fraction is expressed with most precision when the most significant hexadecimal digit (not necessarily the most significant bit) is nonzero. The floating-point number with such a fraction is called a normalized floating-point number. In Perkin-Elmer processors, normalized numbers are always used to obtain the maximum possible precision.

Examples: The following examples illustrate the sign, exponent, and fraction concept of a floating-point number:

Numbers in Hexadecimal integer-fraction notation	Sign-exponent-fraction shown for clarity	Single-precision floating-point numbers
	S | E | F	
+1.3A25678	0 41 13A25678	4113A256
-6.89F2C	1 41 689F2C	C1689F2C
+1A.C39D21	0 42 1AC39021	421AC39D
-3C1DF.82A3	1 45 3C1DF82A3	C53C1DF8
+ABCDEF12.9AC	0 48 ABCDEF129AC	48ABCDEF
+0.0032A9CF2	0 3E 32A9CF2	3E32A9CF
-0.000002C7B5	1 3B 2C7B5	BB2C7B50

Floating-Point Number Range

The range of magnitude (M) of a normalized floating-point number is as follows:

Single precision: $16^{-65} \leq M \leq (1 - 16^{-6}) * 16^{64}$
Double precision: $16^{-65} \leq M \leq (1 - 16^{-14}) * 16^{64}$
Approximately for both: $5.4 * 10^{-79} \leq M \leq 7.2 * 10^{75}$

Table A-1 shows the floating-point range in relation to the fixed-point range, along with the decimal values.

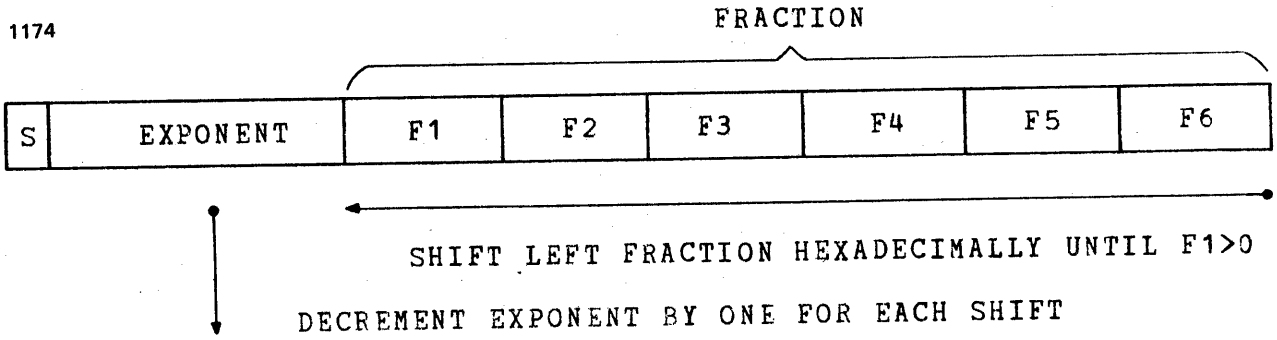
TABLE A-1 FLOATING/FIXED-POINT RANGES

1173

FLOATING-POINT NUMBERS		FIXED-POINT INTEGER	DECIMAL NUMBERS
(most negative)	FFFF FFFF		$-7.2* 10^{75}$
	C880 0000	8000 0000 (most negative)	-2 147 483 648
	C111 0000	FFFF FFFF (least negative)	-1
(least negative)	8010 0000		$-5.4*10^{-79}$
(true zero)	0000 0000	0000 0000	0
(least positive)	0010 0000		$+5.4*10^{-79}$
	4110 0000	0000 0001 (least positive)	+1
	487F FFFF	7FFF FFFF (most positive)	+2 147 483 647
(most positive)	7FFF FFFF		$+7.2*10^{75}$

Normalization

Normalization is a process of making nonzero the most significant digit (F1) of the fraction of a floating-point number. In the normalization process, the floating-point fraction is shifted left hexadecimally (i.e., four bits at a time), and its exponent is decremented by one for each hexadecimal shift until the most significant digit (not necessarily the most significant bit) of the fraction is nonzero.



Except for the load instructions, all floating-point operations assume and require normalized operands for consistent results. The load instructions normalize an unnormalized operand.

Example:

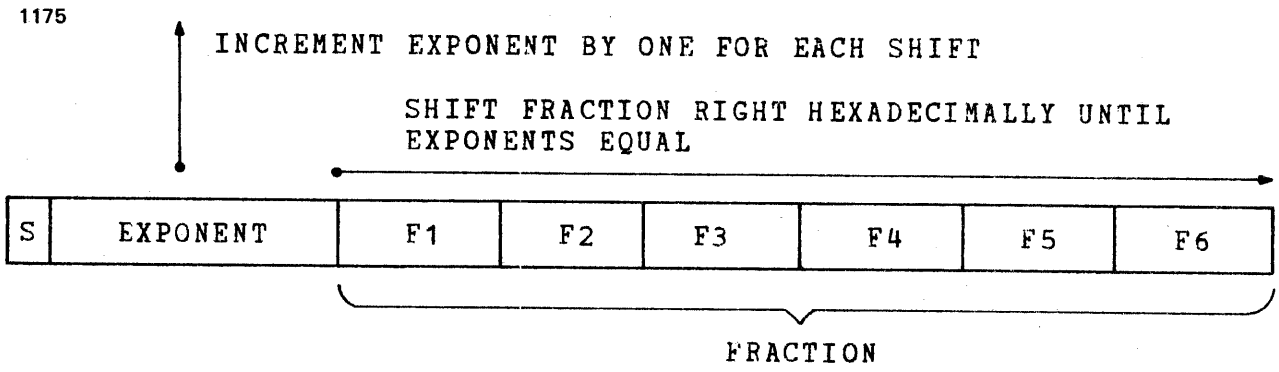
	Operands	After normalization
1.	42012345	41123450
2.	21000ABC	1EABC000
3.	C900FE12	C7FE1200
4.	6C000000	00000000 (true zero)

In Example 4, the fraction of the operand is zero. During the normalization process, such a fraction is detected, and the floating-point number is set to true zero.

Normalized results are always produced in floating-point operations, assuming the operands are normalized. Results of operations between unnormalized numbers are undefined.

Equalization

Equalization is a process of equalizing exponents of two floating-point numbers. The fraction of the floating-point number with the smaller exponent is shifted right hexadecimally (i.e., four bits at a time), and its exponent is incremented by one for each hexadecimal shift until the two exponents are equal.



During floating-point addition and subtraction, the two floating-point operands are equalized.

Example:

	Floating-point operands	After equalization
1.	43123456 3F789ABC	43123456 43000078
2.	C7FE1234 4956789A	C900FE12 4956789A

In this example, normalized floating-point numbers are shown because addition and subtraction require normalization. If the exponents differ by more than 6 for single-precision or more than 14 for double-precision, the representable significance of the lower exponent floating-point number is lost in the process of equalization. Digits shifted out are shifted through the guard digits and may still have an effect on the result, sum, or difference.

True Zero

A floating-point number is true zero when the exponent and the fraction fields are all zeros; therefore, all data bits must be zero. A zero value always has a positive sign. In general, zero values participate as normal operands in all floating-point operations.

A true zero may be used as an operand. It may also result from an arithmetic operation that caused an exponent underflow, in which case the entire number may be forced to true zero. If an arithmetic operation produces a result whose fraction digits are all zeros (sometimes referred to as loss of significance), the entire number is forced to true zero.

Examples:

Numbers	Operation	Result	Reason
030000AB	normalize	0000 0000	exponent underflow
41ABCDEF 41ABCDEF	subtract	0000 0000	loss of significance

Exponent Overflow

In floating-point operations, exponent overflow occurs when a resulting exponent is greater than +63. If overflow occurs, the result register is unchanged. The condition code is set to reflect the overflow situation and the resulting sign. Figure A-1 illustrates exponent overflow using a line representation of numbers.

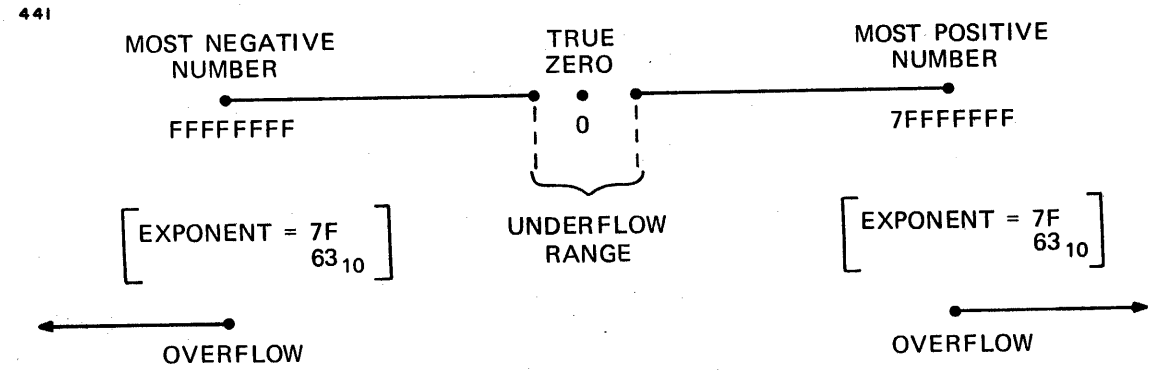


Figure A-1 Exponent Overflow

If overflow occurs, the V flag in the condition code is set, and an arithmetic fault interrupt is taken. Exponent overflow interrupts cannot be disabled.

Exponent Underflow

The normalization process, during a floating-point operation, may produce an exponent underflow. This underflow occurs when a result exponent is less than -64. Figure A-2 illustrates exponent underflow using a line representation of numbers.

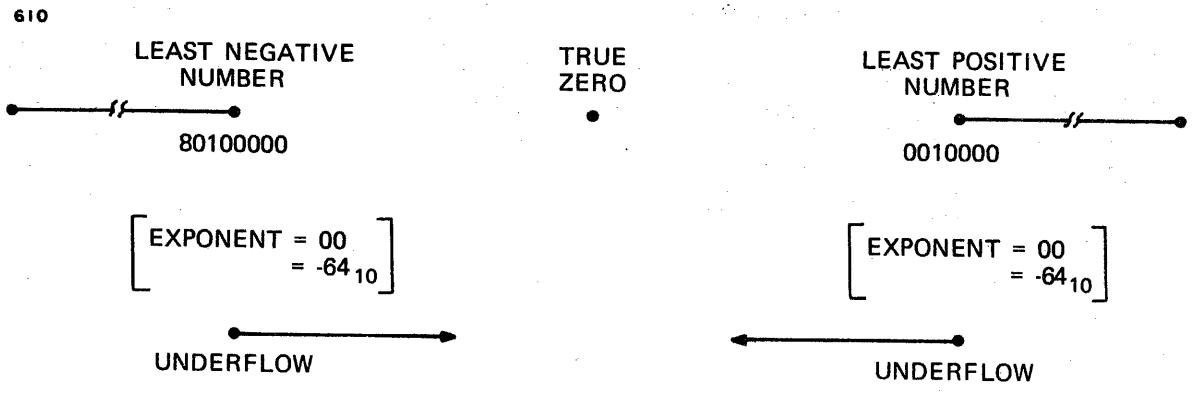


Figure A-2 Exponent Underflow

If underflow occurs, an arithmetic fault interrupt is taken, if enabled by the current PSW. Both operands remain unchanged. If underflow is disabled by the current PSW, the result is forced to zero (the closest possible answer), the V flag in the condition code is set, and the next sequential instruction is executed.

Guard Digits and R*-Rounding

When an intermediate floating-point result has been formed, it consists of a sign, an exponent, and a fraction field. The fraction field is extended by a number of guard digits containing the least significant fraction digits of the intermediate result. Before the result is copied to a destination, it is rounded to compensate for the loss in the final result of the guard digits.

The rules for the R*-rounding scheme are:

- If the most significant guard digit is hexadecimal 7 or less, no rounding is performed. (See Example 1.)
- If the most significant guard digit is hexadecimal 8, and all other guard digits are 0, the least significant bit of the final result is forced to 1. (See Example 2.)
- If the most significant guard digit is hexadecimal 8, and another guard digit is nonzero, or if the most significant guard digit is hexadecimal 9 or greater, 1 is added to the fraction field of the final result. (See Example 3.) If this addition produces a carry-out of the fraction field (i.e., fraction field was all 1s), the result exponent is incremented by 1, the most significant fraction digit (F1) is set to hexadecimal 1, and all other fraction digits are set to 0. (See Example 4.) Note that exponent overflow could occur as the result of rounding.

Examples of R*-Rounding

	Intermediate Result	Final Single-Precision Result
1.	42ABCD12 32680000	42ABCD12
2.	C1183756 80000000	C1183757
3.	3E265739 80100000	3E26573A
4.	41FFFFFF F0000000	42100000

Conversion from Decimal

To convert a decimal number into the excess-64 notation used internally by the processor, the following steps must be taken:

1. Separate the decimal integer from the decimal fraction:

$$182.375_{10}=(182 + .375)_{10}$$

2. Convert each part to hexadecimal by referring to the integer conversion table and the fraction conversion table in Appendix D.

$$182_{10} = B6_{16} \qquad .375_{10} = .6_{16}$$

3. Combine the hexadecimal integer and fraction:

$$B6.6_{16} = (B6.6X16^{0})_{16}$$

4. Shift the radix point:

$$(B6.6X16^{0})_{16} = (.B66X16^{2})_{16}$$

5. Add 64 (X'40') to the exponent:

$$40_{16} + 2_{16} = 42_{16}$$

6. Convert the exponent field and fractions to binary allowing 1 bit for the sign, 7 bits for exponent field, and 24 or 56 bits for the fraction.

42B66 = 0100 0010 1011 0110 0110 0000 0000 0000

APPENDIX B
GUARD DIGITS

To clarify the role of guard digits in floating-point machine arithmetic, several factors must be considered. The notation used in the following discussion is as follows:

- f: number of digits in the fraction portion of a floating-point number
- g: number of guard digits
- q: number of equalization or alignment shifts

In normalized floating-point multiplication, the unnormalized product can have, at most, one leading zero. This requires a post normalization left shift of one digit. For this reason, multiplication requires a 2f+1 digit result register to produce an exact result. If rounding is used to obtain an f digit product, f+2 digits must be developed, one for post normalization and one for rounding.

Normalized floating-point addition is trickier than multiplication. If all the terms to be summed are positive, all post normalization shifts are to the right, so only one guard digit is of value in rounding. If the terms in the sums are of mixed signs (+ and -), then more guard digits may be of use. One guard digit may be of use for post normalization, as in multiplication. For instance, if forming A-B, assume that B<=A. The subtrahend B is aligned by a right shift of q>=0 digits. If q=0, no alignment shift has occurred, therefore guard digits would be of no help. If q>=1, the subtrahend shift could be helped by guard digits. In this case, due to borrows, the result can have a maximum of one leading zero digit; thus one guard digit can be used to hold the digit which may be returned when normalizing the result. Another guard digit would be useful for rounding the result.

A third use for guard digits is in mixed sign summations. Notice that a minuend of the form 10...0 could have a very small number subtracted from it. This would lead to a very long sequence of borrows; thus, the following hexadecimal subtraction with f=6 and g=5 makes good use of all five guard digits:

```
        f          g
1 0 0 0 0 0   0 0 0 0 0     A
          1   0 0 0 0 1     B
-------------------------------
0 F F F F E   F F F F F     C
```

Any number of guard digits (within the range necessary for representable numbers) could be used; however, the number of guard digits can be limited by a careful analysis of the borrow process.

In the case of binary subtraction, any sequence of borrows propagates from right to left across a sequence of zeros in the minuend until a one appears in the minuend. At this point, the borrow sequence stops. The difference bits generated in this process are a one at the right end and then a sequence of one and zeros, which are the complements of the corresponding bits in the subtrahend.

Since the g low-order digits of the minuend are always zero, the number of guard digits (g) can be limited to three by using a special trick in the third digit. The low-order third guard digit is defined as a sticky bit with the property that if a one is ever shifted into it during an alignment, it is set to one; otherwise, it remains a zero. In renormalizing, a maximum of one digit can be shifted to the left so the second digit takes part in rounding. The third is not used, except in the case of R*-rounding, where it can be used to test if all shifted-through digits have been zero. Any borrow sequences are generated by the sticky bit to its left, the same as they would be if more guard digits were present.

APPENDIX C
R*-ROUNDING

When performing floating-point operations, it is often necessary to dispose of extra low-order digits. This allows fast and inexpensive implementation in hardware, with results of numerically high quality. The following discussion relates to a floating-point number which consists of a fractional mantissa of f digits with g extra digits to be disposed. The result is f digits which are used in succeeding calculations to represent the original f+g digits.

In conventional rounding, an amount is added to the f+g digits equal to 1/2 in the last of the f digits and the original f digits are then retained. For example, 7.FF + .8 = 8.7F, so 8 is retained while 7.7F + .8 = 7.FF, so 7 is retained. The maximum error is incurred with a case such as 7.8 + .8 = 8, and is equal to 1/2 in the last retained digit in f.

As shown in Figure C-1, round (x) is nearly symmetric with respect to the ideal line. However, the dots above the ideal line indicate that rounding up always takes place at the points exactly halfway between the x and y values. Thus, over a long sequence of operations, a slight positive bias is expected because all of these dots are above the ideal line. Table C-1 shows the digits of f and g=2 bits. By summing the absolute error in all possible cases, the total bias is obtained. For any g, there is a pairing of positive and negative biases, with the 10...0 case in the center; thus, a total bias of 1/2 in the last retained digit of f is obtained.

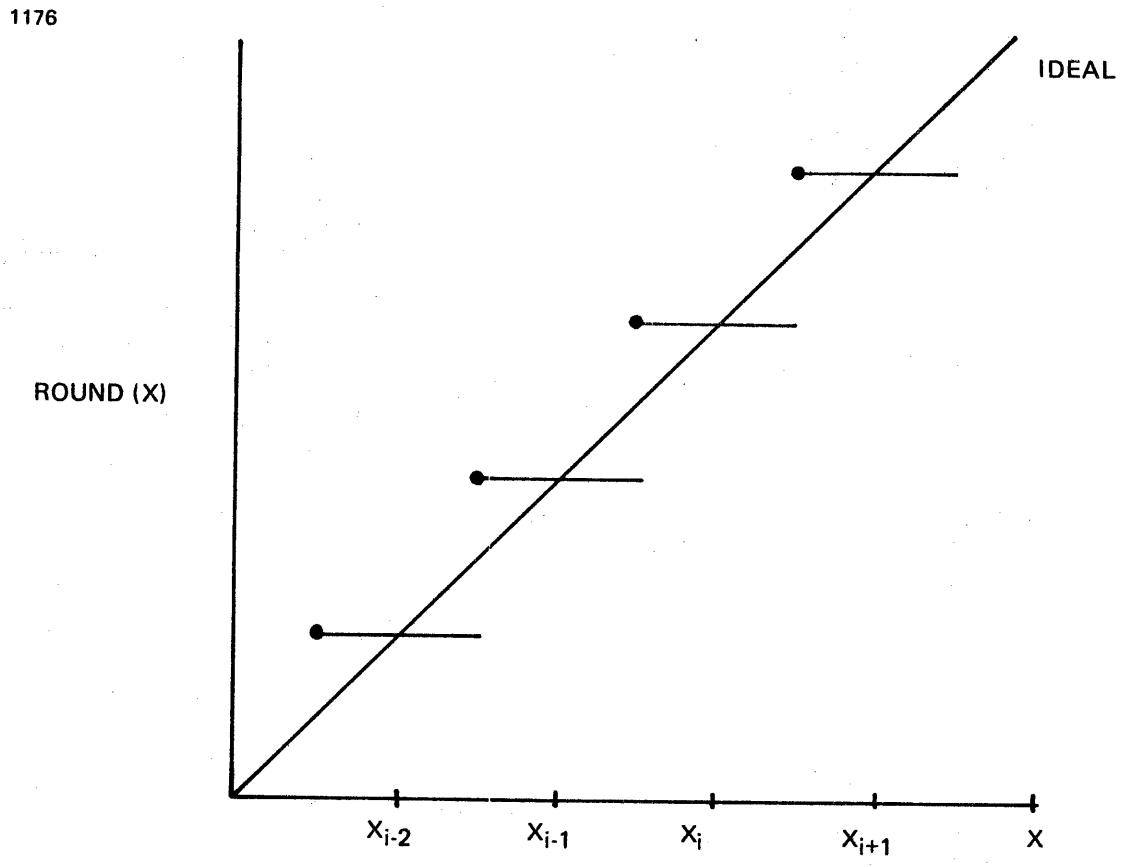


Figure C-1 Round (X) Plot

TABLE C-1 TOTAL BIAS

NUMBER REPRESENTED			ROUNDED RESULT	2^f x ERROR
2^{-f} .	$2^{-(f+1)}$	$2^{-(f+2)}$	2^{-f} .	
X .	0	0	X .	0
X .	0	1	X .	-1/4
X .	1	0	X .	1 - 1/2 = 1/2
X .	1	1	X .	1 - 3/4 = 1/4
				Total Bias = 1/2

R*-rounding operates the same as rounding, except when the g digits have the form 10...0. As indicated in Figure C-2, alternate representable floating-point numbers are forced to lie on intervals which are closed at both ends. Therefore the "exactly one-half" values are treated in a symmetric way. Half the dots are above the ideal and half are below.

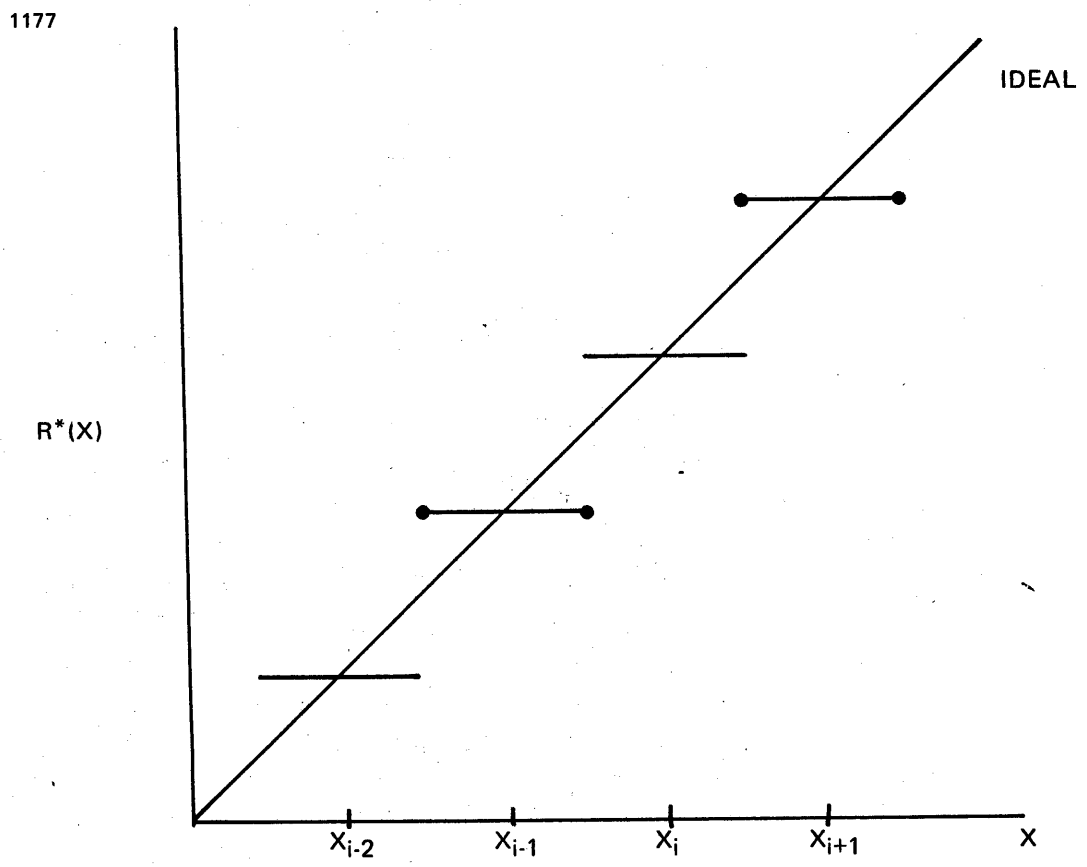


Figure C-2 R* (X) Plot

In hexadecimal, the 80...0 case is detected and a 1 is ORed into the last retained digit of f. Figure C-1 shows that the bias is removed by this procedure. If the low-order bit of f is a 1, 1/2 is truncated, so this case is biased by -1/2. If the low-order bit of f is a 0, 1 is added and 1/2 is dropped, so this case is biased by +1/2. The sum of these two cases is zero, giving the R* scheme a total bias of zero.

APPENDIX D
ARITHMETIC REFERENCES

TABLE D-1 POWERS OF TWO

1024

$(2^n)_{10}$	$(2^n)_{16}$	n	2^{-n}
1	1	0	1.0
2	2	1	0.5
4	4	2	0.25
8	8	3	0.125
16	10	4	0.062 5
32	20	5	0.031 25
64	40	6	0.015 625
128	80	7	0.007 812 5
256	100	8	0.003 906 25
512	200	9	0.001 953 125
1 024	400	10	0.000 976 562 5
2 048	800	11	0.000 488 281 25
4 096	1 000	12	0.000 244 140 625
8 192	2 000	13	0.000 122 070 312 5
16 384	4 000	14	0.000 061 035 156 25
32 768	8 000	15	0.000 030 517 578 125
65 536	10 000	16	0.000 015 258 789 062 5
131 072	20 000	17	0.000 007 629 394 531 25
262 144	40 000	18	0.000 003 814 697 265 625
524 288	80 000	19	0.000 001 907 348 632 812 5
1 048 576	100 000	20	0.000 000 953 674 316 406 25
2 097 152	200 000	21	0.000 000 476 837 158 203 125
4 194 304	400 000	22	0.000 000 238 418 579 101 562 5
8 388 608	800 000	23	0.000 000 119 209 289 550 781 25
16 777 216	1 000 000	24	0.000 000 059 604 644 775 390 625
33 554 432	2 000 000	25	0.000 000 029 802 322 387 695 312 5
67 108 864	4 000 000	26	0.000 000 014 901 161 193 847 656 25
134 217 728	8 000 000	27	0.000 000 007 450 580 596 923 828 125
268 435 456	10 000 000	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	20 000 000	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	40 000 000	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	80 000 000	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	100 000 000	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	200 000 000	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	400 000 000	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	800 000 000	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	1 000 000 000	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	2 000 000 000	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	4 000 000 000	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	8 000 000 000	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	10 000 000 000	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5

TABLE D-2 POWERS OF SIXTEEN

1022

16^n							n
						1	0
						16	1
						256	2
					4	096	3
					65	536	4
				1	048	576	5
				16	777	216	6
				268	435	456	7
			4	294	967	296	8
			68	719	476	736	9
		1	099	511	627	776	10
		17	592	186	044	416	11
		281	474	976	710	656	12
	4	503	599	627	370	496	13
	72	057	594	037	927	936	14
1	152	921	504	606	846	976	15

Decimal Values

TABLE D-3 HEXADECIMAL TO DECIMAL INTEGER CONVERSION

1023

BYTE				BYTE			
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0
1	4,096	1	256	1	16	1	1
2	8,192	2	512	2	32	2	2
3	12,288	3	768	3	48	3	3
4	16,384	4	1,024	4	64	4	4
5	20,480	5	1,280	5	80	5	5
6	24,576	6	1,536	6	96	6	6
7	28,672	7	1,792	7	112	7	7
8	32,768	8	2,048	8	128	8	8
9	36,864	9	2,304	9	144	9	9
A	40,960	A	2,560	A	160	A	10
B	45,056	B	2,816	B	176	B	11
C	49,152	C	3,072	C	192	C	12
D	53,248	D	3,328	D	208	D	13
E	57,344	E	3,584	E	224	E	14
F	61,440	F	3,840	F	240	F	15

TABLE D-4 HEXADECIMAL ADDITION AND SUBTRACTION

1020

Examples: 5+A = F; 18-D = B; A+B = 15

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	1
2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	2
3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	3
4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	4
5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	5
6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	6
7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	7
8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	8
9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	9
A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	A
B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	B
C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	C
D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	D
E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	E
F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	F
	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

TABLE D-5 HEXADECIMAL MULTIPLICATION AND DIVISION

1021

Examples: 5x6 = 1E; 75÷D = 9; 58÷8 = B; 9xC = 6C

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	1
2	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E	2
3	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D	3
4	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C	4
5	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B	5
6	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A	6
7	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69	7
8	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78	8
9	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87	9
A	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96	A
B	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5	B
C	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4	C
D	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3	D
E	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2	E
F	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1	F
	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

TABLE D-6 MATHEMATICAL CONSTANTS

1019

CONSTANT	DECIMAL VALUE	HEXADECIMAL VALUE	FLOATING POINT VALUE
			DOUBLE PRECISION
			SINGLE PRECISION
π	3.14159 26535 89793 23846	3.243F 6A88 85A3 08D3	4132 43F6 A888 5A31
$\pi-1$	0.31830 98861 83790 67154	0.517C C1B7 2722 0A95	4051 7CC1 B727 220B
$\sqrt{\pi}$	1.77245 38509 05516 02730	1.C5BF 891B 4EF6 AA7A	411C 5BF8 91B4 EF6B
Ln π	1.14472 98858 49400 17414	1.250D 048E 7A1B D0BD	4112 B67A E858 4CAA
$\sqrt{3}$	1.73205 08075 68877 29353	1.BB67 AE85 84CA A73B	411B 67AE 8584 CAA7
e	2.71828 18284 59045 23536	2.B7E1 5162 8AED 2A6B	412B 7E15 1628 AED3
e^{-1}	0.36787 94411 71442 32160	0.5E2D 58D8 B3BC DF1B	405E 2D58 D8B3 BCDF
$\sqrt{e}$	1.64872 12707 00128 14683	1.A612 98E1 E069 BC97	411A 6129 8E1E 069C
$\log_{10}e$	0.43429 44819 03251 82765	0.6F2D EC54 9B94 38CB	406F 2DEC 5A9B 9439
$\log_{2}e$	1.44269 50408 88963 40736	1.7154 7652 B82F E177	4117 1547 652B 82FE
γ	0.57721 56649 01532 86061	0.93C4 67E3 7DB0 C7A5	4093 C467 E37D B0C8
Ln γ	-0.54953 93129 81644 82234	-0.8CAE 9BC1 1F5A 5FF4	C08C AE9B C11F 5A60
$\sqrt{2}$	1.41421 35623 73095 04880	1.6A09 E667 F3BC C909	4116 A09E 667F 3BCD
Ln2	0.69314 71805 59945 30942	0.B172 17F7 D1CF 79AC	40B1 7217 F7D1 CF7A
$\log_{10}2$	0.30102 99956 63981 19521	0.4D10 4D42 7DE7 FBCC	404D 104D 427D E7FC
$\sqrt{10}$	3.16227 76601 68379 33199	3.298B 075B 4B6A 5240	4132 98B0 75B4 B6A5
Ln10	2.30258 50929 94045 68402	2.4D76 3776 AAA2 B05C	4124 D763 776A AA2B

TABLE D-7 INTEGER CONVERSION

1017

Hexadecimal and Decimal Integer Conversion Table

HALFWORD								HALFWORD							
BYTE				BYTE				BYTE				BYTE			
BITS: 0123		4567		0123		4567		0123		4567		0123		4567	
Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	268,435,456	1	16,777,216	1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	536,870,912	2	33,554,432	2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	805,306,368	3	50,331,648	3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	1,073,741,824	4	67,108,864	4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	1,342,177,280	5	83,886,080	5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	1,610,612,736	6	100,663,296	6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	1,879,048,192	7	117,440,512	7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	2,147,483,648	8	134,217,728	8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	2,415,919,104	9	150,994,944	9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	2,684,354,560	A	167,772,160	A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	2,952,790,016	B	184,549,376	B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	3,221,225,472	C	201,326,592	C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	3,489,660,928	D	218,103,808	D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	3,758,096,384	E	234,881,024	E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	4,026,531,840	F	251,658,240	F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
8		7		6		5		4		3		2		1	

1025

TO CONVERT HEXADECIMAL TO DECIMAL

1. Locate the column of decimal numbers corresponding to the left-most digit or letter of the hexadecimal; select from this column and record the number that corresponds to the position of the hexadecimal digit or letter.
2. Repeat step 1 for the next (second from the left) position.
3. Repeat step 1 for the units (third from the left) position.
4. Add the numbers selected from the table to form the decimal number.

EXAMPLE	
Conversion of Hexadecimal Value	D34
1. D	3328
2. 3	48
3. 4	4
4. Decimal	3380

To convert integer numbers greater than the capacity of table, use the techniques below:

HEXADECIMAL TO DECIMAL

Successive cumulative multiplication from left to right, adding units position.

Example: $D34_{16} = 3380_{10}$

D = 13
x16
208
3 = +3
211
x16
3376
4 = +4
3380

1026

TO CONVERT DECIMAL TO HEXADECIMAL

1. (a) Select from the table the highest decimal number that is equal to or less than the number to be converted.
   (b) Record the hexadecimal of the column containing the selected number.
   (c) Subtract the selected decimal from the number to be converted.
2. Using the remainder from step 1(c) repeat all of step 1 to develop the second position of the hexadecimal (and a remainder).
3. Using the remainder from step 2 repeat all of step 1 to develop the units position of the hexadecimal.
4. Combine terms to form the hexadecimal number.

EXAMPLE	
Conversion of Decimal Value	3380
1. D	-3328
	52
2. 3	-48
	4
3. 4	-4
4. Hexadecimal	D34

DECIMAL TO HEXADECIMAL

Divide and collect the remainder in reverse order.

Example: $3380_{10} = X_{16}$

16 | 3380 → remainder
16 | 211 → 4
16 | 13 → 3
D

$3380_{10} = D34_{16}$

TABLE D-8 FRACTION CONVERSION

1018

Hexadecimal and Decimal Fraction Conversion Table

HALFWORD													
BYTE					BYTE								
BITS 0123		4567			0123				4567				
Hex	Decimal	Hex	Decimal		Hex	Decimal			Hex	Decimal Equivalent			
.0	.0000	.00	.0000	0000	.000	.0000	0000	0000	.0000	.0000	0000	0000	0000
.1	.0625	.01	.0039	0625	.001	.0002	4414	0625	.0001	.0000	1525	8789	0625
.2	.1250	.02	.0078	1250	.002	.0004	8828	1250	.0002	.0000	3051	7578	1250
.3	.1875	.03	.0117	1875	.003	.0007	3242	1875	.0003	.0000	4577	6367	1875
.4	.2500	.04	.0156	2500	.004	.0009	7656	2500	.0004	.0000	6103	5156	2500
.5	.3125	.05	.0195	3125	.005	.0012	2070	3125	.0005	.0000	7629	3945	3125
.6	.3750	.06	.0234	3750	.006	.0014	6484	3750	.0006	.0000	9155	2734	3750
.7	.4375	.07	.0273	4375	.007	.0017	0898	4375	.0007	.0001	0681	1523	4375
.8	.5000	.08	.0312	5000	.008	.0019	5312	5000	.0008	.0001	2207	0312	5000
.9	.5625	.09	.0351	5625	.009	.0021	9726	5625	.0009	.0001	3732	9101	5625
.A	.6250	.0A	.0390	6250	.00A	.0024	4140	6250	.000A	.0001	5258	7890	6250
.B	.6875	.0B	.0429	6875	.00B	.0026	8554	6875	.000B	.0001	6784	6679	6875
.C	.7500	.0C	.0468	7500	.00C	.0029	2968	7500	.000C	.0001	8310	5468	7500
.D	.8125	.0D	.0507	8125	.00D	.0031	7382	8125	.000D	.0001	9836	4257	8125
.E	.8750	.0E	.0546	8750	.00E	.0034	1796	8750	.000E	.0002	1362	3046	8750
.F	.9375	.0F	.0585	9375	.00F	.0036	6210	9375	.000F	.0002	2888	1835	9375
1		2			3				4				

TO CONVERT .ABC HEXADECIMAL TO DECIMAL

Find .A in position 1 .6250

Find .0B in position 2 .0429 6875

Find .00C in position 3 .0029 2968 7500

.ABC Hex is equal to .6708 9843 7500

TO CONVERT .13 DECIMAL TO HEXADECIMAL

1. Find .1250 next lowest to subtract
   .1300
   -.1250 = .2 Hex
2. Find .0039 0625 next lowest to
   .0050 0000
   -.0039 0625 = .01
3. Find .0009 7656 2500
   .0010 9375 0000
   -.0009 7656 2500 = .004
4. Find .0001 0681 1523 4375
   .0001 1718 7500 0000
   -.0001 0681 1523 4375 = .0007
   .0000 1037 5976 5625 = .2147 Hex
5. 13 Decimal is approximately equal to ——————↑

To convert fractions beyond the capacity of table, use techniques below:

HEXADECIMAL FRACTION TO DECIMAL

Convert the hexadecimal fraction to its decimal equivalent using the same technique as for integer numbers. Divide the results by 16^n (n is the number of fraction positions).

Example: $.8A7 = .540771_{10}$

$8A7_{16} = 2215_{10}$

$16^3 = 4096$

$$4096\overline{)2215.000000} = .540771$$

DECIMAL FRACTION TO HEXADECIMAL

Collect integer parts of product in the order of calculation.

Example: $.5408_{10} = .8A7_{16}$

.5408
x16
8 ← [8].6528
x16
A ← [10].4448
x16
7 ← [7].1168

APPENDIX E
ROM MAPS

1178

MS H	MS G	MS F	MS E	LS D	LS C	LS B	LS A	Y4	Y3	Y2	Y1
SNRLZ1	ST051	ST021	ST011	XCRY0	F001	F011	F021				NOT USED

LS \ MS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	F	B	D	0	D	0	0	0	F	B	D	0	D	0	0	0	RCC/LMSN
1	F	B	D	0	D	0	0	0	F	B	D	0	D	0	0	0	READ
2	F	D	D	0	D	0	0	0	F	D	D	0	D	0	0	0	LOAD
3	F	B	D	0	D	0	0	0	F	B	D	0	D	0	0	0	COMPARE
4	F	7	D	0	D	0	0	0	F	7	D	0	D	0	0	0	ADD
5	F	7	D	0	F	0	0	0	F	7	D	0	D	0	0	0	SUBT.
6	F	F	D	0	D	0	0	0	F	F	D	0	D	0	0	0	MULT.
7	F	B	D	0	D	0	0	0	F	F	D	0	D	0	0	0	DIV
8	F	B	F	0	D	0	0	0	F	B	F	0	D	0	0	0	REC/LMSH
9	F	B	F	0	D	0	0	0	F	B	F	0	D	0	0	0	READ
A	F	D	F	0	F	0	0	0	F	D	F	0	D	0	0	0	LOAD
B	F	B	F	0	D	0	0	0	F	B	F	0	D	0	0	0	COMPARE
C	F	7	F	0	D	0	0	0	F	7	F	0	D	0	0	0	ADD
D	F	7	F	0	F	0	0	0	F	7	F	0	D	0	0	0	SUBT.
E	F	F	F	0	F	0	0	0	F	F	F	0	D	0	0	0	MULT.
F	F	E	F	0	D	0	0	0	F	F	F	0	D	0	0	0	DIV.

Figure E-1 19-142F67 HPFPP State Control ROM

1179

MS				LS							
H	G	F	E	D	C	B	A	Y4	Y3	Y2	Y1
BGT1	XL011	XL021	XL031	XL041	XL051	XL061	XL071			NOT USED	NOT USED

LS \ MS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	C	1	1	1	0	D	1	1	D	1	1	1	0	1	1	1
1	C	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
2	C	1	1	1	1	1	1	1	0	1	1	1	1	1	1	8
3	C	1	1	1	1	1	1	1	0	1	1	1	1	1	1	8
4	C	1	1	1	1	1	1	1	0	1	1	1	1	1	1	8
5	C	1	1	1	1	1	1	1	0	1	1	1	1	1	1	8
6	C	1	1	1	1	1	1	1	0	1	1	1	1	1	1	8
7	C	1	1	1	1	1	1	1	0	1	1	1	1	1	1	8
8	9	1	1	1	1	1	1	1	0	1	1	1	1	1	1	8
9	9	1	1	1	1	1	1	0	0	1	1	1	1	1	1	C
A	9	1	1	1	1	1	1	0	0	1	1	1	1	1	1	C
B	9	1	1	1	1	1	1	0	0	1	1	1	1	1	1	C
C	9	1	1	1	1	1	1	0	0	1	1	1	1	1	1	C
D	9	1	1	1	1	1	1	0	0	1	1	1	1	1	1	C
E	9	1	1	1	1	1	1	0	1	1	1	1	1	1	1	C
F	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	C

Figure E-2 19-188F21 HPFPP Very Great ROM

1180

	MS				LS						
	H	G	F	E	D	C	B	A	Y4	Y3	Y2	Y1
NOT USED (0)												
F021												
F011												
F001												
AEQBD0												
SBGT1												
ASTK001												
BMX000												
NOT USED												
NOT USED												

LS \ MS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	
1	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	
2	3	1	1	1	1	1	3	1	0	0	0	0	0	0	0	0	
3	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
4	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	
5	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	
6	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
7	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
8	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	
9	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	
A	3	1	1	0	1	1	3	0	0	0	0	0	0	0	0	0	
B	3	1	1	1	1	1	3	1	0	0	0	0	0	0	0	0	
C	3	0	1	1	1	1	3	1	0	0	0	0	0	0	0	0	
D	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	
E	3	0	1	0	1	1	3	0	0	0	0	0	0	0	0	0	
F	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0

	RCC/LW	ADD	LOAD	MULT.	READ	SUBT.	COMPARE	DIV.

Figure E-3 19-188F30 HPFPP CC ROM

APPENDIX F
HPFPP-A MNEMONIC LIST

The following list provides a brief description of each mnemonic found on the HPFPP-A. The source of each signal, on Schematic Drawing 35-715D08, is also provided.

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
AAD00:30	A stack address lines. These address lines address a particular register in the A register stack. Bit 00 determines whether single- or double-precision registers are being addressed.	12K1,12K2
AAD01:31	A stack address lines. These address lines originate on HPFPP-B. They are buffered to form the HPFPP-A A stack address lines.	12J1,12J2
AB080	This is bit 08 of the A bus which comes from the A multiplexor and feeds the A side of the mantissa ALU. It is also used to set the carry bit in divide (M071).	13G8
ACLK0/1/1A	These are the main system clocks used throughout the HPFPP to clock control elements. ACLK1 and ACLK1A are decoded from EACLK0. ACLK0 is decoded from ACLK1.	4N6,4N7
ACRY0/1	After XCRY. This is XCRY0/XCRY1 delayed by one clock.	8F8
AEQB1	This signal indicates that the mantissa of the A operand is equal to the mantissa of the B operand on HPFPP-A (bits 08:31).	13J3

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
AEQB1C	This signal indicates that the mantissa of the A operand is equal to the mantissa of the B operand on HPFPP-B (bits 32:67).	6H6
AEQBC0	This signal indicates that the A operand is equal to the B operand (bits 00:67).	6N5
AL08:32	These bits are the output of the A latch. They are the latched data which has come from the mantissa ALU and passed through the A latch shifter. These bits are fed back to the A multiplexor and B multiplexor so that they may be gated onto the A bus or B bus. They are also fed back to the A latch shifter so that the entire A latch may be shifted left one bit without passing through the ALU (required by the divide algorithm). AL320 originates on HPFPP-B.	Sheets 13:18
ALCK0	A latch clock. The A latch clock is used to clock the A latch. It is decoded from ACLK1.	4N6
ALMAS1	A latch multiplexor A select line. When this signal is low, data from the mantissa ALU can be shifted left one bit. When this signal is high, data from the A latch can be shifted left one bit.	8N9
ALMBS1	A latch multiplexor B select line. When this signal is low, data from the mantissa ALU can pass to the A latch without being shifted. When this signal is high, data from the mantissa ALU can be shifted to the left four bits.	8N2
ALSRN0	A latch shift right enable. This signal enables the 4-bit 4-way shifter which allows data from the mantissa ALU to be shifted right 1, 2, 3, or 4 bits and then passed to the A latch.	8N3

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
AMINB0	This signal indicates that the mantissa ALU is to find the arithmetic difference between the A operand and the B operand.	7H8
AMN0	A multiplexor enable. This signal enables the A multiplexor, allowing either the contents of the addressed A stack register or the output of the A latch to be gated onto the A bus.	8N1
AMQCLK0	MQ register clock. This signal clocks the MQ register on HPFPP-A.	8N7
AMXS0/1	A multiplexor select line. When AMXS1 is low, the A multiplexor can gate the contents of the A latch onto the A bus. When AMXS1 is high, the A multiplexor can gate the output of the A stack onto the A bus.	8F6
AONLY1	This signal indicates that the contents of the A bus are to be passed directly through the mantissa ALU.	7H2
APLSB0	This signal indicates that the sum of the A bus and the B bus is to be produced by the mantissa ALU.	7H6
AS0/1	These signals are decoded from RD bits 09, 10, and 11. They indicate that the current HPFPP operation is either an add or a subtract.	5K5,5K3
ASF1	Add or subtract function. This signal indicates that the add or subtract function should be performed by the mantissa ALU during this clock period.	8N5
ASTK001	Bit 00 from the A register stack. This bit is used in determining the sign of the result, the condition code, and whether an add or subtract operation requires the sum or the difference of the operands.	11J7

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
ASTKWN1	A stack write pulse enable. This signal enables the write pulse which strobes data into the A register stacks.	9G8
ASUM0/1	These signals define the current add or subtract operation as being the sum of the operands.	6G6
AWN0	A stack write enable. This signal strobes the data which resides on the B bus into the currently addressed register in the A register stack.	9G8
BAD000:030	B stack address lines. These lines address a particular register in the B register stack. Bit 00 determines whether single- or double-precision registers are being addressed.	12N1,12N2
BAD001:031	B stack address lines. These address lines originate on HPFPP-B. They are buffered to form the HPFPP-A B stack address lines.	12M1,12M2
BGT0	B operand is greater than A operand. This signal can become active only during state 1 of an add or subtract operation.	6G5
BMAN0	B multiplexor A enable. This signal enables B multiplexor A which gates either the contents of the A latch or the output of the currently addressed register in the B register stack onto the B bus.	8F3
BMAS1	B multiplexor A select line. When low, this signal can gate the contents of the A latch through to the B bus. When high, the output of the B register stack can be gated to the B bus.	8F6
BMBN0	B multiplexor B enable. This signal enables B multiplexor B which gates either the contents of the MQ register or the data from the C bus onto the B bus.	8F3

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
BMBS1	B multiplexor B select line. When low, this signal can gate data from the C bus through to the B bus. When high, the contents of the MQ register can be gated onto the B bus.	8F1
BMX00:31 BMX321 BMX641	B bus. This B bus is internal to the HPFPP. It has no connection to the B bus in the CPU. The B bus is the main data path in the HPFPP. Data can be placed on the B bus from the C bus, MQ register, A latch register, or B register stack. The B bus feeds both the exponent and mantissa ALUs as well as the A and B register stacks. Bit 00 is the sign bit. Bits 01:07 are the exponent. Bits 08:64 are the mantissa or fraction part of the floating-point number. Only bits 00:31 reside on HPFPP-A. Bits 08:11 are used to determine if a floating-point number is normalized. Bits 32 and 64 originate on HPFPP-B and are used to determine whether the result should be rounded.	Sheets 3, 11, 12, 14:18
BONLY0	This signal indicates that the contents of the B bus are to be passed directly through the mantissa ALU.	7G4
BSCLR0X	Buffered system clears. These signals force the HPFPP into the idle state (state 0) and initialize it so that it is ready to accept a floating-point operation from the CPU. These signals are activated by system clear from the system console. System clear does not affect the contents of the register stacks.	19H6,19M7
BST001:311	B stack output. Data from the B register stack can be gated onto the B bus through B multiplexor A. The B stack also supplies data to the C bus drivers.	Sheets 11, 13:18
BSTKWN1	B stack write pulse enable. This signal enables the write pulse which strobes data into the B register stacks.	9G7

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
BWNO	B stack write enable. This signal strobes the data on the B bus into the currently addressed register in the B register stack.	9G7
C0/1	Compare. This signal is decoded from RD bits 09, 10, and 11 from the CPU. It indicates that the HPFPP is to perform a compare operation.	5K3
C000:310	C bus. The C bus is a bidirectional data bus which connects the HPFPP to the processor. All data which comes into the HPFPP from the C bus gets latched into the MQ register. All data which goes from the HPFPP to the CPU is gated onto the C bus from the B register stack.	Sheet 19
CGXX1	Carry generate lines. These signals are used in the carry-lookahead circuitry of the mantissa ALU.	Sheets 13:19
CLK1	This is the system clock from the CPU. Its purpose in the HPFPP is to maintain synchronization with the processor.	2B4
CPXX1	Carry propagate lines. These signals are used in the carry-lookahead circuitry of the mantissa ALU.	Sheets 13:19
CRYXX1	Carry inputs to the ALUs. These carry input signals are generated by the carry-lookahead circuitry of the mantissa ALU.	19M5,19M4
D0/D1	Divide operation. These signals are decoded from RD bits 09, 10, and 11. They indicate that the current HPFPP operation is a divide.	5K4
DIST21	This signal indicates that the HPFPP is in state 2 during a divide operation.	5K8

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
DIST30	This signal indicates that the HPFPP is in state 3 during a divide operation.	10B1
DIST40	This signal indicates that the HPFPP is in state 4 during a divide operation.	7H9
DAL070	Decoded AL bit 07. This signal simulates bit 07 of the A latch in multiply operations.	7N4
DBB0	Disable B bus. This signal disables both B multiplexor A and B multiplexor B. This causes the B bus to be pulled up so that it looks like all zeros.	8F3
DCLK0/1	Delayed clock. This signal becomes set one clock after fast clock (FCLK0/FCLK1), and gets reset on the next clock. It is used by the divide algorithm to set up the first division step.	6N1,6N5
DCLR0X	Device clears. The device clears are decoded from state 0 or from systems clear. They reset the HPFPP so that it is ready to accept the next floating-point operation.	2L4
DFTEN0	Double-precision floating-point enable. This signal comes from the CPU to enable the HPFPP to begin a double-precision operation on the next CPU clock edge.	2B1
DNRLZ1	Divide and normalized. This signal indicates that the divisor is non-zero during state 1 of a divide operation.	5K6
DST010	Divide and state 1. This signal allows the divisor to be written into the B stack during state 1 of an RX divide operation.	9G6

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
DST41	Delayed state 4. This signal is state 4 delayed by one clock period. This delay allows time to restore registers after a fault during a divide operation.	8F6
EACLK0	Early clock. This clock is decoded from WCLK1.	4N5
EQU0	Equalize. This signal indicates that one of the operands needs to be equalized or is being equalized. This signal originates on HPFPP-B.	8H3
ERND0	End rounding. This signal terminates an HPFPP operation if the result required rounding.	3D3
EXCLK0	External clock. This is a test point to which an external clock may be connected. The test point is on pin 206 of header 3. When using an external clock, it is necessary to disable internal clocks by placing switch number 8 in the ON position.	4J5
F001:021	Function select bits. These bits are buffered RD bits 09,10, and 11. When decoded, they indicate what specific function the HPFPP is to perform.	2L7
FCLK0/1	Fast clock. By remaining active in state 2 of a divide operation, this signal indicates that the divisor is larger than the dividend.	6N2,6N3
FIN1	Finished. This signal indicates that the HPFPP operation is finished and that the normalized result is ready to be rounded and stored into the register stacks.	3F1

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
GDCLK0	Gated delayed clock. This signal becomes active only when the delayed clock is active in state 2 of a divide operation (first clock in state 2).	8A6
JACRY0	Jam the after carry flop. This signal activates the ACRY0/ACRY1 flip-flop.	8F8
JAM0	Jam. This signal can become active only during the final rounding sequence. It disables the normal rounding procedure, and instead forces the least significant bit of the result to a logical "1".	3A2
JBMX0	When active, this signal causes B multiplexor B to become enabled and B multiplexor A to become disabled on the next clock.	8F5
JBMXAS1	When active, this signal causes B multiplexor A to select the B register stack as its input after the next clock.	8A6
JMQ280:310	Inputs to the MQ register. These lines feed bits 28, 29, 30, and 31 of the MQ register. They are also used by the multiplication algorithm to control the single-precision multiply operation.	18N3
JUFL0	Jam underflow flip-flop. This signal becomes active when an underflow condition is detected. The underflow flip-flop becomes active after the next clock.	10H2
JVFLG0/1	Jam overflow flip-flop. This signal becomes active when an overflow condition is detected. The overflow flip-flop becomes active after the next clock.	3K5,10K5

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
JVGT0/1	Jam very great flip-flop. This signal becomes active when the difference between the A operand and the B operand exponents is greater than the equalization capability of the HPFPP in an add or subtract operation.	6G2
JXCRY0	Jam XCNTR carry flip-flop. This signal becomes active when the XCNTR reaches a count of zero.	8A9
L0/1	Load function. This signal is decoded from RD bits 09, 10, and 11. It indicates that the current HPFPP operation is a floating-point load.	5K2
LDXCY1	Load function or XCNTR carry out. This signal enables detection of a zero result or an underflow fault.	6M9
LENB0	Enable A latch multiplexor B. This signal is one set of conditions which enables A latch multiplexor B.	8N5
LMNA0	A latch multiplexor A enable. This signal enables this multiplexor to place data on the A latch bus.	8N2
LMNB0	A latch multiplexor B enable. This signal enables this multiplexor to place data on the A latch bus.	8N5
LMSH0	Load most significant half. This signal is decoded from RD bits 09, 10, and 11. It indicates that the data on the C bus is the most significant 32 bits of a double-precision floating-point number. The flip-flop which is set by this signal is the only control element in the HPFPP which is not initialized when the HPFPP returns to the idle state (state 0).	5K1

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
LOAD0	Ready to load. This signal indicates that the HPFPP operation is complete and that the result is not yet normalized.	8N2
LOAD1	Ready to load. This signal indicates that the HPFPP has obtained a result which must be normalized, rounded, and stored.	8L2
M0/1	Multiply. This signal is decoded from RD bits 09, 10, and 11. It indicates that the current HPFPP operation is a multiply.	5K4
M071	Carry bit for divide. This signal generates the quotient during a divide operation.	8N8
M1ST20/1	Multiply and state 2. This signal becomes active during state 2 of a multiply operation.	5K8
MAEQB0	Mantissa A equals B. This signal indicates that the mantissa of the A operand is equal to the mantissa of the B operand (bits 08:67).	6N6
MAL080:350	Output from mantissa ALU. These signals are the output of the mantissa ALU. Bits 08:31 originate on HPFPP-A and bits 32:35 come from HPFPP-B. These signals are passed through the A latch shifter and latched in the A latch.	Sheets 13:18
MCOUT0/1	Mantissa carry-out. This signal indicates that a mantissa ALU function has resulted in a carry. This signal is generated by the carry-lookahead circuitry.	19M6
MDNRLZ1	Multiply, or divide and normalized.	5K7
MFN0	Multiply finished. This signal indicates that a multiply operation is finished. This signal originates on HPFPP-B.	8H5

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
MMODE1	This signal determines the mode of operation for the mantissa ALU.	7H4
MOFF0/1	Mantissa overflow. This bit indicates that the previous mantissa ALU operation resulted in a mantissa carry-out.	6N4
MQ08:32	Multiplier-quotient register. These bits are the output from the multiplier-quotient register. They may be gated back through the MQ shifter or onto the B bus.	Sheets 13:18
MQCE0	MQ register clear enable. This is a pulse which occurs only at the beginning of an HPFPP operation. The pulse initializes that portion of the MQ register which resides on HPFPP-B (bits 32:63). This signal is disabled after the execution of the LW microinstruction.	8N6
MS001:031	Mode select. These lines, in combination with MMODE1, determine the function to be performed by the mantissa ALU.	7H8,7H7
MSC01:11	Multiply shift code. These signals indicate the shift distance (1, 2, 3, or 4 bits) necessary or the next step in the multiplication algorithm. These signals are decoded to address the 4-bit 4-way shifters in the A latch shifter and the MQ shifter.	8H4
MSH0	Gate most significant half to C bus. This signal is significant only during a read operation. It indicates that either a single-precision number or the most significant half of a double-precision number (bits 00:31) is to be gated onto the C bus.	19A9
MSUM0/1	Multiply sum. During a multiplication operation, this signal indicates that the current partial product was produced by a summing of operands rather than a difference of operands.	7N5

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
NRLZ1	Normalized. This signal indicates that the data on the B bus is normalized.	5K6
OPTSTCP0	Stop CPU clocks. This signal becomes active when the HPFPP is activated. It causes the CPU to stop and wait for the HPFPP to run to completion. When the HPFPP finishes the current operation, the processor clocks are released.	2L2
PLUS1	This signal indicates that the exponent ALU is to perform a summing function.	9G4
PSW191	Bit 19 of the program status word. When active, this signal indicates that if a floating-point operation results in an exponent underflow, it should be handled as an arithmetic fault.	3G6
R0	Read operation. This signal is decoded from RD bits 09, 10, and 11. It indicates that the current operation is a read from a floating-point register.	5K2
RCC0	Read condition code. This signal is decoded from RD bits 09, 10, and 11. It indicates that the current condition code is to be returned to the processor on bits 28:31 of the C bus.	2L6
RD1	Round. This signal enables the rounding mechanism in the HPFPP.	7N3
RD090:110	ROM data bits from the CPU. These bits are buffered into the HPFPP to form the function select bits. They specify the particular function to be performed by the HPFPP.	2B7

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
RD251	ROM data bit 25 from the CPU. This signal indicates that the current floating-point operation is a memory operation rather than a register-to-register operation.	8A1
REGDIS0	Register stacks disable. This signal disables the register stacks on the HPFPP in the event that a floating-point operation is to be aborted.	11C6
REGNA0	A register stacks enable.	11C7
REGNB0	B register stacks enable.	15A8
RR0/1	Register-to-register. This signal indicates that the current operation is a register-to-register operation rather than a memory operation. This signal is decoded from RD bit 25.	8F1
RSIGN0/1	Resultant sign bit. This bit is the sign of the result of the floating-point operation.	9R5,9R4
RSTOP0	ROM stop from the CPU. Indicates the CPU has halted microoperations.	2B5
S3WT0	State 3 and write. This signal indicates that the result of the floating-point operation is ready to be written into the register stacks.	9G9
S4MOF0	Shift right 4 bits and mantissa overflow. This signal ensures that bit 11 of the result gets set to a one after a mantissa overflow.	8N1
SBGT1	This signal indicates that the B operand is greater than the A operand in an add or subtract operation.	6G4
SBMBN0	Set B multiplexor B enable. This signal causes B multiplexor B to become enabled and B multiplexor A to become disabled. It is a pulse which occurs only before the first clock of a register-to-register operation.	8F2

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
SCLK1	Skew clock from the CPU. This CPU clock is used to maintain synchronization between the CPU and the HPFPP.	2B3
SCLR0A	Systems clear. This signal causes the HPFPP to become initialized to the idle state so that it is ready to accept a floating-point operation.	19E6
SEL1	HPFPP selected. This signal indicates that the HPFPP has been activated to perform a floating-point operation.	2L2
SFTEN0	Single-precision floating-point enable. This signal comes from the CPU to enable the HPFPP to begin a single-precision operation on the next CPU clock edge.	2B2
SHSL00:10	4-bit 4-way shifter select lines. These select lines are decoded from the multiply shift code lines. They indicate to the A latch shifter and the MQ shifter to shift right 1, 2, 3, or 4 bits.	8N4,8N3
SKSIG0	This signal disables the normalization sequence during a floating-point operation.	5B6
SLWCLK0	Slow clock. This signal causes all the HPFPP clocks to become stretched by about 30 ns. This allows time for the stacks to settle after a write and compensates for particularly slow data paths.	6N3
SMCOUT0	Shift carry-out into multiplicand. This signal maintains the most significant bits of the partial product during a multiply operation.	7N4
SNRLZ0/1	Normalized. This signal indicates that the data on the B bus is normalized and no mantissa overflow condition exists.	5K7

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
SP0/1	Single-precision. This signal indicates that the current floating-point operation is to be done in single-precision. These signals are not consistent when the HPFPP is in the idle state (state 0).	2L1,2L2
SPRD0	Round single-precision result. This signal forces a 1 to be added into bit 31 of the floating-point result.	7N2
SSC0/1	Start system clock. This signal is a short pulse which occurs when the HPFPP is activated. It initiates the control sequence by forcing the state controller out of state 0 and into state 1.	2L5
ST000/1	State 0. State 0 is the idle state of the HPFPP. When in state 0, all control functions are initialized and the HPFPP is ready to accept an operation.	3N2,3N1
ST010/1	State 1. State 1 is the start-up state of the HPFPP.	3N3
ST020/1	State 2. State 2 is entered only for multiplication or division. This is where these algorithms are performed.	3N4
ST030/1	State 3. State 3 is always the final state of a successful floating-point operation. In state 3, the HPFPP performs add and subtract operations, equalization, normalization, rounding, and storing of the final result.	3N5
ST040/1	State 4. State 4 is the fault state. If an arithmetic fault occurs, state 4 is entered to clear up the trouble. All the floating-point registers are restored to their states as they were before the operation began.	3N6
ST050/1	State 5. State 5 is entered when it is necessary to change the contents of the register stacks to carry out an operation.	3N7
SWN1	Write enable. Enables the stack write pulses.	9G8

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
UFL0/1	Underflow. This signal indicates that an exponent underflow has occurred. The action taken then depends on the state of PSW bit 19.	10K3
UFLT0/1	Underflow fault. This signal indicates that an exponent underflow has occurred and that it should be considered an arithmetic fault. The controller goes to state 4 and the registers are restored.	3K6
VFLG0	Overflow. This signal indicates that an exponent overflow has occurred. Exponent overflow is always considered to be an arithmetic fault. The controller goes to state 4 and the registers are restored.	10K4
VGT0/1	Very great. This signal indicates that the difference between the A operand and the B operand exponents is greater than the equalization capability of the HPFPP in an add or subtract operation.	6G3
WCLK1	Write clock. This clock is used to produce the stack write pulses.	4N5
WTRND0	Write after rounding. This signal inhibits the stack write pulse until after the result has been properly rounded.	3F3
XAEQB1	Exponent A equal B. This signal indicates that the exponent of the A operand is equal to the exponent of the B operand.	11N7
XAL000:70	Exponent ALU output. These bits are latched unshifted into the exponent A latch.	11N6,12N6
XC041	This signal is the ripple carry-line between the two 4-bit ALUs that make up the exponent ALU.	12N7

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
XCN1	Exponent carry-in. This signal provides the carry input to the exponent ALU.	9G2
XCOUT1	Exponent carry-out. This is the carry-out signal from the exponent ALU.	11N7
XCRY0/1	XCNTR carry-out. This signal indicates that the XCNTR on HPFPP-B has finished counting. This signifies the completion of equalization, multiplication, or division.	8F9
XEL0	This signal indicates an exponent overflow in state 1 of a multiply or divide operation.	11N4
XL001:71	Exponent A latch bits. These signals are the latched output from the exponent ALU.	11N3,12N3
XMDS1	Exponent ALU mode select. This signal, together with PLUS1, defines the function of the exponent ALU.	9G3
XMN0	Exponent multiplexor enable. This signal enables the exponent multiplexor which allows the contents of the A register stack or the exponent A latch to be gated onto the exponent bus.	9G5
XMQ30:31	Multiplexed MQ bits. These bits are latched from MQ bits 30:31 in single-precision or bits 62:63 in double-precision. These signals are used to control the multiplication algorithm.	7A5,7A8
XMQCLK0	Exponent MQ clock. This signal latches data into the exponent register from the C bus at the beginning of an HPFPP operation.	8N6
XMS1	Exponent multiplexor select line. When active, this signal gates the output of the A register stack onto the exponent bus.	9G4

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
XMX000:70	Exponent bus. These signals come from the exponent multiplexor and feed the A side of the exponent ALU.	11J6,12J6
XPA1/XPB1	Pull-ups. These lines are pull-up lines which are connected to unused gate inputs.	6G1,2L3
XSC0/1	Exponent special case. This signal becomes active when a special case of potential exponent overflow is detected.	10K8
ZER00/1	Zero. This signal indicates that the result of a floating-point operation is a true zero.	6N7

APPENDIX G
HPFPP-B MNEMONIC LIST

The following list provides a brief description of each mnemonic found in the HPFPP-P. The source of each signal, on Schematic Drawing 35-716D08, is also provided.

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
AAD001:31	A stack address lines. These address lines are decoded in the A register stack. AAD001 is decoded from SP1.	Sheet 2
ACLK1	This is the main system clock used throughout the HPFPP. It is decoded from EACLK0.	4E7
ACRY0	After XCRY. This signal is XCRY0/XCRY1 delayed by one clock.	3F6
AEQB1A	This signal indicates that bits 32:35 of the A operand are equal to bits 32:35 of the B operand.	6J3
AEQB1B	This signal indicates that bits 36:67 of the A operand are equal to bits 36:67 of the B operand.	14J9
AEQB1C	This signal indicates that the mantissa of the A operand is equal to the mantissa of the B operand on HPFPP-B (bits 32:67).	Sheet 14

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
AL32:68	These bits are the output of the A latch. They are the latched data which has come from the mantissa ALU and passed through the A latch shifter. These bits are fed back to the A multiplexor and B multiplexor so that they may be gated onto the A bus or B bus. They are also fed back to the A latch shifter so that the entire A latch may be shifted left one bit without passing through the ALU (required by the divide algorithm).	Sheets 6:15
ALCK0	A latch clock. The A latch clock is used to clock the A latch. It is decoded from ACLK1.	4E6
ALMAS1	A latch multiplexor A select line. When this signal is low, data from the mantissa ALU can be shifted left one bit. When this signal is high, data from the A latch can be shifted left one bit.	3M7
ALMBS1	A latch multiplexor B select line. When this signal is low, data from the mantissa ALU can pass to the A latch without being shifted. When this signal is high, data from the mantissa ALU can be shifted to the left four bits.	3M6

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
ALSRNO	A latch shift right enable. This signal enables the 4-bit 4-way shifter which allows data from the mantissa ALU to be shifted right 1, 2, 3, or 4 bits and then passed to the A latch.	3M8
AMINBO/1	This signal indicates that the mantissa ALU is to find the arithmetic difference between the A operand and the B operand.	5D7
AMNO	A multiplexor enable. This signal enables the A multiplexor, allowing either the contents of the addressed A stack register or the output of the A latch to be gated onto the A bus.	4E9
AMXSO/1	A multiplexor select line. When AMXS1 is low, the A multiplexor can gate the contents of the A latch onto the A bus. When AMXS1 is high, the A multiplexor can gate the output of the A stack onto the B bus.	4D4,4E4
AONLY1	This signal indicates that the contents of the A bus are to be passed directly through the mantissa ALU.	8A7
APLSBO	This signal indicates that the sum of the A bus and the B bus is to be produced by the mantissa ALU.	5A6
AS1	This signal is decoded from RD bits 09, 10, and 11. It indicates that the current HPFPP operation is either an add or a subtract.	2N7
AST321	Bit 32 of the A register stacks.	6D8
ASTKWN1	A stack write pulse enable. This signal enables the write pulse which strobes data into the A register stacks.	4A5

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
AWNO	A stack write enable. This signal strobes the data which resides on the B bus into the currently addressed register in the A register stack.	4E6
BAD001:31	B stack address lines. These address lines can be decoded from either YD or YS, depending on the state of BADMS1.	Sheet 2
BADMS1	B stack address multiplexor select line. When this signal is low, the B address lines are decoded from YS. When it is high, the B address lines are decoded from YD.	3E9
BGT0/1	B operand is greater than A operand. This signal can become active only during state 1 of an add or subtract operation.	13J2,3M2
BMAN0	B multiplexor A enable. This signal enables B multiplexor A, which gates either the contents of the A latch or the output of the currently addressed register in the B register stack onto the B bus.	4E1
BMAS1	B multiplexor A select line. When low, this signal can gate the contents of the A latch through to the B bus. When high, the output of the B register stack can be gated to the B bus.	4E3
BMBN0,1	B multiplexor B enable. This signal enables B multiplexor B, which gates either the contents of the MQ register or the data from the C bus through to the B bus.	4E1, 14K1

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
BMQCLK0	MQ register clock. This signal clocks that portion of the MQ register located on the HPFPP-B (bits 32:63).	4E6
BMX32:64	B bus. This B bus is internal to the HPFPP. It has no connection to the B bus in the CPU. The B bus is the main data path in the HPFPP. Data can be placed on the B bus from the C bus, MQ register, A latch register, or B register stack. The B bus feeds both the exponent and mantissa ALUs as well as the A and B register stacks. Bit 00 is the sign bit. Bits 01:07 are the exponent. Bits 08:64 are the mantissa or fraction part of the floating-point number. Only bits 32:64 reside on HPFPP-B.	Sheets 6:14
BONLY	This signal indicates that the contents of the B bus are to be passed directly through the mantissa ALU.	5A7
BSCLR0A	Buffered system clear. This signal places the HPFPP in the idle state (state 0) and initializes the processor so that it is ready to accept a floating-point operation from the CPU.	15H7
BST32:63	B stack output. This data from the B register stack can be gated onto the B bus through B multiplexor A. It can also be gated onto the C bus to return data to the CPU.	Sheets 6:13
BSTKWN1	B stack write pulse enable. This signal enables the write pulse, which strobes data into the B register stacks.	4A5

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
BWN0	B stack write enable. This signal strobes data on the B bus into the currently addressed register in the B register stack.	4E5
C000:310	C bus. The C bus is a bidirectional data bus which connects the HPFPP to the processor. All data which comes into the HPFPP from the C bus gets latched into the MQ register. All data which goes from the HPFPP to the CPU is gated onto the C bus from the B register stack.	Sheet 15
CBEN0	C bus enable. This signal switches the C bus transceivers from receive mode to transmit mode.	2N2
CGXX1	Carry generate lines. These signals are used in the carry-lookahead circuitry of the mantissa ALU.	Sheets 6:14
CPXX1	Carry propagate lines. These signals are used in the carry-lookahead circuitry of the mantissa ALU.	Sheets 6:14
CRYXX1	Carry inputs to the ALUs. These carry input signals are generated by the carry-lookahead circuitry of the mantissa ALU.	Sheets 14,15
CT010/1	Count equal to one. This signal becomes active when the count in the XCNTR is equal to one.	5R5
CTCLK0	Counter clock. This clock is used to clock the XCNTR and the XCRY flip-flop.	3E7
CTM10	Count equal to minus one. This signal becomes active when the count in the XCNTR is equal to minus one.	5R4

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
DO/1	Divide operation. These signals are decoded from RD bits 09, 10, and 11. They indicate that the current HPFPP operation is a divide.	2N7
DIST21	This signal indicates that the HPFPP is in state 2 during a divide operation.	2N8
DIST40	This signal indicates that the HPFPP is in state 4 during a divide operation.	4D4
DBB0	Disable B bus. This signal disables both B multiplexor A and B multiplexor B. This causes the B bus to be pulled up so that it locks like all zeros.	4A2
DCLK0	Delayed clock. This signal becomes set one clock after fast clock (FCLK0/FCLK1), and gets reset on the next clock. It is used by the divide algorithm to set up the first division step.	3J7
DCLR0A/B	Device clears. The device clears are decoded from state 0 or from systems clear. They reset the HPFPP so that it is ready to accept the next floating-point operation.	5D8
DDVQ0	Double-precision divide quotient. This signal is shifted into bit 64 of the MQ register during state 2 of a double-precision divide operation to produce the quotient.	3M6
DPRD0	Double-precision rounding bit. This signal jams bit 63 of the A bus to a one.	14N8

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
DVQ0	Single-precision divide quotient. This signal is shifted into bit 32 of the MQ register during state 2 of a single-precision divide operation to produce the quotient.	3M3
DFTEN0	Double-precision floating-point enable. This signal comes from the CPU to enable the HPFPP to begin a double-precision operation on the next CPU clock edge.	2A1
EACLK0	Early clock. This clock is decoded from WCLK1.	4A7
EQU0/1	Equalize. This signal indicates that one of the operands needs to be equalized or is being equalized.	3F4
F001:21	Function select bits. These bits are buffered RD bits 09, 10, and 11. When decoded, they indicate what specific function the HPFPP is to perform.	2F5
FCLK0	Fast clock. By remaining active in state 2 of a divide operation, this signal indicates that the divisor is larger than the dividend.	3J6
JACRY0	Jam the after carry flop. This signal activates the ACRY0/ACRY1 flip-flop.	3B6
JAM0	Jam. This signal becomes active only during the final rounding sequence. It disables the normal rounding procedure and, instead, forces the least significant bit of the result to a one.	14N4
JBMX0	When active, this signal causes B multiplexor B to become enabled and B multiplexor A to become disabled on the next clock.	4A1

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
JBMXAS1	When active, this signal causes B multiplexor A to select the B register stack as its input after the next clock.	4E2
JMQ28:31 JMQ60:63	Inputs to the MQ register. These lines feed bits 28:31 and bits 60:63 of the MQ register. They are also used by the multiplication algorithm to control the single- and double-precision multiply operations.	4G5,13N1
JVFLG0	Jam overflow flip-flop. This signal becomes active when an overflow condition is detected. The overflow flip-flop becomes active after the next clock.	Sheet 4
JVGT0/1	Jam very great flip-flop. This signal becomes active when the difference between the A operand and the B operand exponents is greater than the equalization capability of the HPFPP in an add or subtract operation.	3B5
JXCRY0	Jam XCNTR carry flip-flop. This signal becomes active when the XCNTR reaches a count of zero.	5R7
L0/1	Load function. This signal is decoded from RD bits 09, 10, and 11. It indicates that the current HPFPP operation is a floating-point load.	2N6
LIST30	Load and state 3.	3E9
LAEQB1B	Latched AEQB1B. This signal saves the information that bits 36:67 of the A operand were equal to bits 36:67 of the B operand.	14N9
LENB0	Enable A latch multiplexor B. This signal is one set of conditions which enables A latch multiplexor B.	3J5

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
LMNA0	A latch multiplexor A enable. This signal enables this multiplexor to place data on the A latch bus.	3M6
LMNB0	A latch multiplexor B enable. This signal enables this multiplexor to place data on the A latch bus.	3M5
LOAD1	Ready to load. This signal indicates that the HPFPP has obtained a result which must be normalized, rounded, and stored.	3M5
LSHEN0	Least significant half enable. This signal enables the C bus drivers on HPFPP-B.	2N2
M071	Carry bit for divide. This signal generates the quotient during a divide operation.	3J1
M1	Multiply. This signal is decoded from RD bits 09, 10, and 11. It indicates that the current HPFPP operation is a multiply.	2N7
MIST20/21A/21B	Multiply and state 2. This signal becomes active during state 2 of a multiply operation.	2N9
MAL280:680	Output from mantissa ALU. These signals are the output of the mantissa ALU. Bits 28:31 originate on HPFPP-A and bits 32:68 come from HPFPP-B. These signals are passed through the A latch shifter and latched in the A latch.	Sheets 6:15
MCOUT1	Mantissa carry-out. This signal indicates that a mantissa ALU function has resulted in a carry. This signal is generated by the carry-lookahead circuitry.	3J3

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
MFNO	Multiply finished. This signal indicates that a multiply operation is finished.	3M6
MMODE1	This signal determines the mode of operation for the mantissa ALU.	5D7
MOFF0/1	Mantissa overflow. This bit indicates that the previous mantissa ALU operation resulted in a mantissa carry-out.	3G5
MQ280:640	Multiplier-quotient register. These bits are the output from the multiplier-quotient register. They may be gated back through the MQ shifter or onto the B Bus.	Sheets 6:14
MQCEO	MQ register clear enable. This is a pulse which occurs only at the beginning of an HPFPP operation. The pulse initializes that portion of the MQ register which resides on HPFPP-B (bits 32:63). This pulse is disabled after the execution of the LW microinstruction.	3J2
MQCLRO	MQ register clear. This signal clears that portion of the MQ register on HPFPP (bits 32:63). It is decoded from MQCEO.	3M2
MS001:031	Mode select. These lines, in combination with MMODE1, determine the function to be performed by the mantissa ALU.	Sheet 5
MSC01:11	Multiply shift code. These signals indicate the shift distance (1, 2, 3, or 4 bits) necessary or the next step in the multiplication algorithm. These signals are decoded to address the 4-bit 4-way shifters in the A latch shifter and the MQ shifter.	4R6,4R5

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
MSH0	Gate most significant half to C bus. This signal is significant only during a read operation. It indicates that either a single-precision number or the most significant half of a double-precision number (bits 00:31) is to be gated onto the B bus.	2N1
MSUM1	Multiply sum. During a multiplication operation, this signal indicates that the current partial product was produced by a summing of operands rather than a difference of operands.	4G3
NRLZ1	Normalized. This signal indicates that the data on the B bus is normalized.	14J8
RO/1	Read operation. This signal is decoded from RD bits 09, 10, and 11. It indicates that the current operation is a read from a floating-point register.	2J6
RD1	Round. This signal enables the rounding mechanism in the HPFPP.	14J7
RD090:110	ROM data bits from the CPU. These bits are buffered into the HPFPP to form the function select bits. They specify the particular function to be performed by the HPFPP.	2A2
RD251	ROM data bit 25 from the CPU. This signal indicates that the current floating-point operation is a memory operation rather than a register-to-register operation.	2A3
REGDIS0	Register stacks disable. This signal disables the register stacks on the HPFPP in the event that a floating-point operation is to be aborted.	6A9

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
REGNA0	A register stacks enable.	6A8
REGNB0	B register stacks enable.	10A8
RR1	Register-to-register. This signal indicates that the current operation is a register-to-register operation rather than a memory operation. This signal is decoded from RD bit 25.	2C3
SBCLR0	This signal disables bit 68 of the A latch during single-precision operations.	5D9
SBMBN0	Set B multiplexor B enable. This signal causes B multiplexor B to become enabled and B multiplexor A to become disabled. It is a pulse which occurs only before the first clock of a register-to-register operation.	4A2
SBT0	Sticky bit. This signal controls the R*-rounding procedure.	14N5
SCLR0A	Systems clear. This signal causes the HPFPP to become initialized to the idle state so that it is ready to accept a floating-point operation.	15E6
SEL1	HPFPP selected. This signal indicates that the HPFPP has been activated to perform a floating-point operation.	2G5
SFTEN0	Single-precision floating-point enable. This signal comes from the CPU to enable the HPFPP to begin a single-precision operation on the next CPU clock edge.	2A2
SHSL00:10	4-bit 4-way shifter select lines. These select lines are decoded from the multiply shift code lines. They indicate to the A latch shifter and the MQ shifter to shift right 1, 2, 3, or 4 bits.	3M9,3M8

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
SKSIG0	This signal disables the normal normalization sequence during a floating-point operation.	4E4
SNRLZ0	Normalized. This signal indicates that the data on the B bus is normalized and no mantissa overflow condition exists.	3J6
SP0/1	Single-precision. This signal indicates that the current floating-point operation is to be done in single-precision. These signals are not consistent when the HPFPP is in the idle state (state 0).	2D1,2D2
ST001	State 0. State 0 is the idle state of the HPFPP. When in state 0, all control functions are initialized and the HPFPP is ready to accept an operation.	4A6
ST010/1	State 1. State 1 is the start-up state of the HPFPP.	4E9
ST020/1	State 2. State 2 is entered only for multiplication or division. This is where these algorithms are performed.	4E8
ST030/1	State 3. State 3 is always the final state of a successful floating-point operation. In state 3, the HPFPP performs add and subtract operations, equalization, normalization, rounding, and storing of the final result.	4E8
ST040	State 4. State 4 is the fault state. If an arithmetic fault occurs, state 4 is entered to clean up the trouble. All the floating-point registers are restored to their states as they were before the operation began.	3B7

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
ST050	State 5. State 5 is entered when it is necessary to change the contents of the register stacks to carry out an operation.	3B8
VGT0	Very great. This signal indicates that the difference between the A operand and the B operand exponents is greater than the equalization capability of the HPFPP in an add or subtract operation.	3B7
WCLK1	Write clock. This clock is used to produce the stack write pulses.	4A5
XAL020:070	Exponent ALU output. These bits are used to initialize the XCNTR for equalization.	Sheet 5
XCRY0/1	XCNTR carry-out. This signal indicates that the XCNTR has finished counting. This signifies the completion of equalization, multiplication, or division.	3E7
XCT000:050	Exponent count signals. These signals are the output of the exponent counter (XCNTR).	5R6,5R3
XCTA010/020	These signals are used to increment or decrement the XCNTR.	4R4,3M1
XMQ30:31	Multiplexed MQ bits. These bits are latched from MQ bits 30:31 in single-precision or bits 62:63 in double-precision. These signals are used to control the multiplication algorithm.	4R7

MNEMONIC	DESCRIPTION	SCHEMATIC LOCATION
XPX1	Pull-ups. These lines are pull-up lines which are connected to unused gate inputs.	3E6,15N4
YD081:111	Destination register. These signals indicate the number of the register into which the result of a floating-point operation is to be stored.	2A6,2A9
YS01:21	Source register. These signals indicate the number of the floating-point register which supplies the source operand in a register-to-register operation.	2A7

APPENDIX H
HPFPP INSTALLATION ON THE MODEL 3210 PROCESSOR

This appendix contains the information necessary to install the HPFPP on the Model 3210 Processor.

NOTE

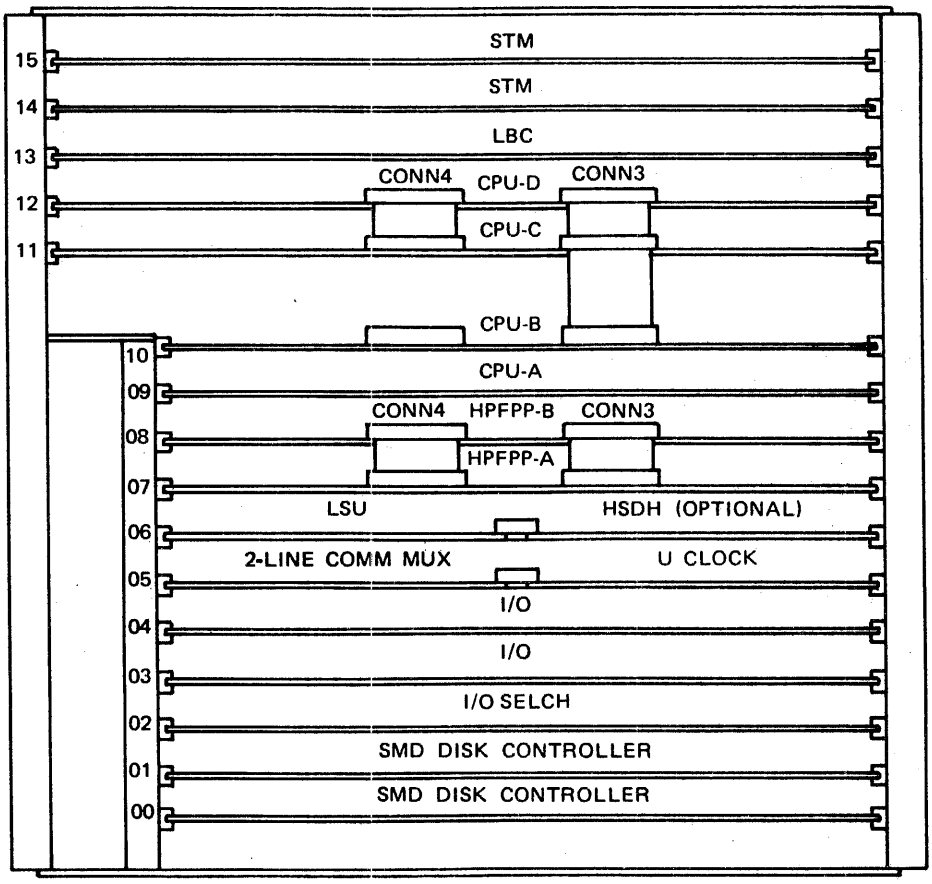
In the Model 3210, the HPFPP cannot be installed unless the chassis with a floating-point backpanel has already been installed at the factory. The floating-point backpanel cannot be installed in the field.

Installation of the HPFPP requires deletion of the strap at location 05M, C to G on the processor CPU-A board. This allows execution of the floating-point instructions.

The HPFPP components are installed in dedicated locations in the 3210 processor chassis as follows (See Figure H-1).

1. Install the HPFPP-A in Slot 07.
2. Install the HPFPP-B in Slot 08.
3. Install the C bus interface on the backpanel, Slot 8, Side 0.
4. Install cable from HPFPP-A, Connector 3 to HPFPP-B, Connector 3.
5. Install cable from HPFPP-A, Connector 4 to HPFPP-B, Connector 4.

2507-1



NOTE: THESE ARE TWO SEPARATE CHASSIS. SLOTS 7 AND 8 IN THE FLOATING-POINT CHASSIS ARE DEDICATED SLOTS AND CANNOT BE USED FOR OTHER BOARDS.

Figure H-1 3210 Board Installation and Cabling

APPENDIX I
HPFPP INSTALLATION ON THE MODEL 3220 PROCESSOR

This appendix contains the information necessary to install the HPFPP on the Model 3220 Processor.

Installation of the HPFPP requires deletion of the strap at location 05M, C to G on the processor CPU-A board. This allows execution of the floating-point instructions.

The HPFPP components are installed in dedicated locations in the 3220 processor chassis as follows (See Figure I-1).

1. Install the HPFPP-A in Slot 05.
2. Install the HPFPP-B in Slot 06.
3. Install the C bus interface on the backpanel, Slot 6, Side 0.
4. Install cable from HPFPP-A, Connector 3 to HPFPP-B, Connector 3.
5. Install cable from HPFPP-A, Connector 4 to HPFPP-B, Connector 4.

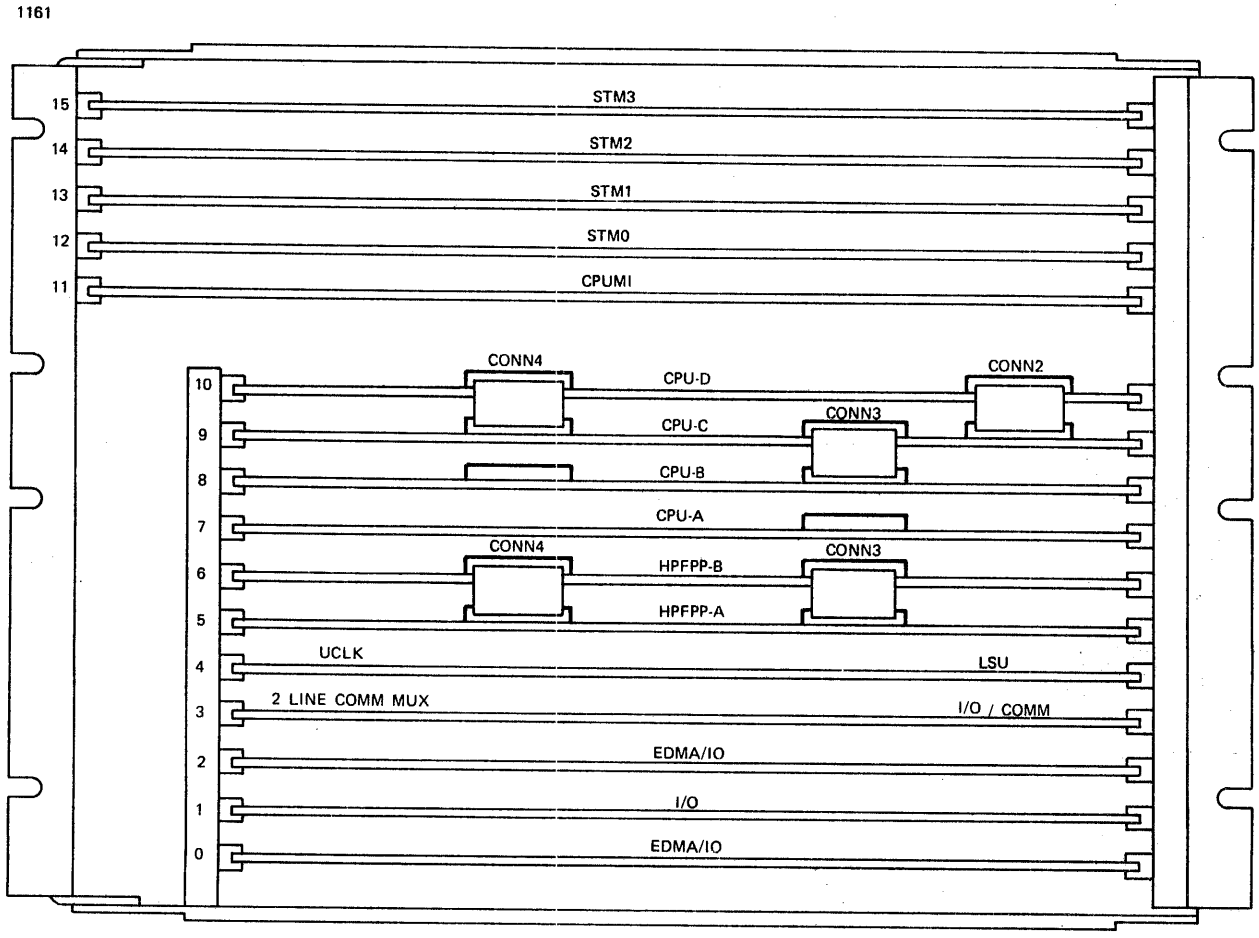


Figure I-1 3220 Board Installation and Cabling

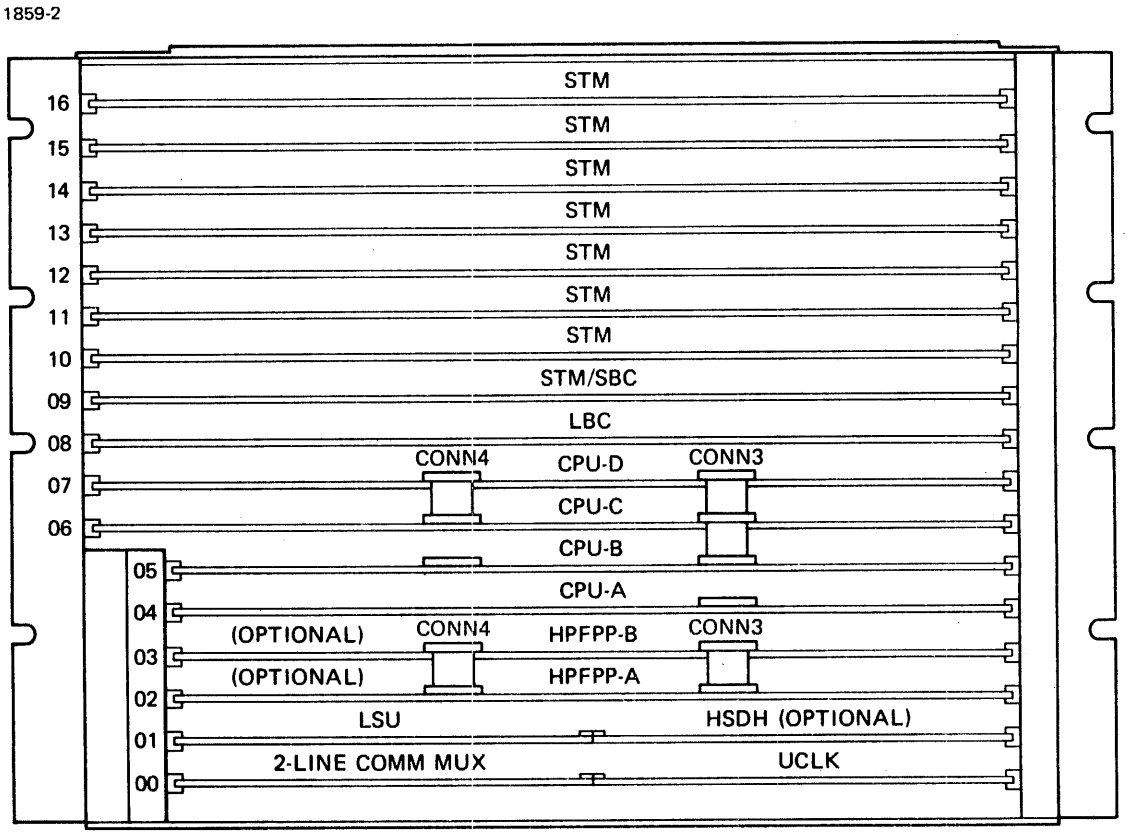
APPENDIX J
HPFPP INSTALLATION ON THE MODEL 3230 PROCESSOR

This appendix contains the information necessary to install the HPFPP on the Model 3230 Processor.

Installation of the HPFPP requires deletion of the strap at location 05M, C to G on the processor CPU-A board. This allows execution of the floating-point instructions.

The HPFPP components are installed in dedicated locations in the 3230 processor chassis as follows (See Figure J-1).

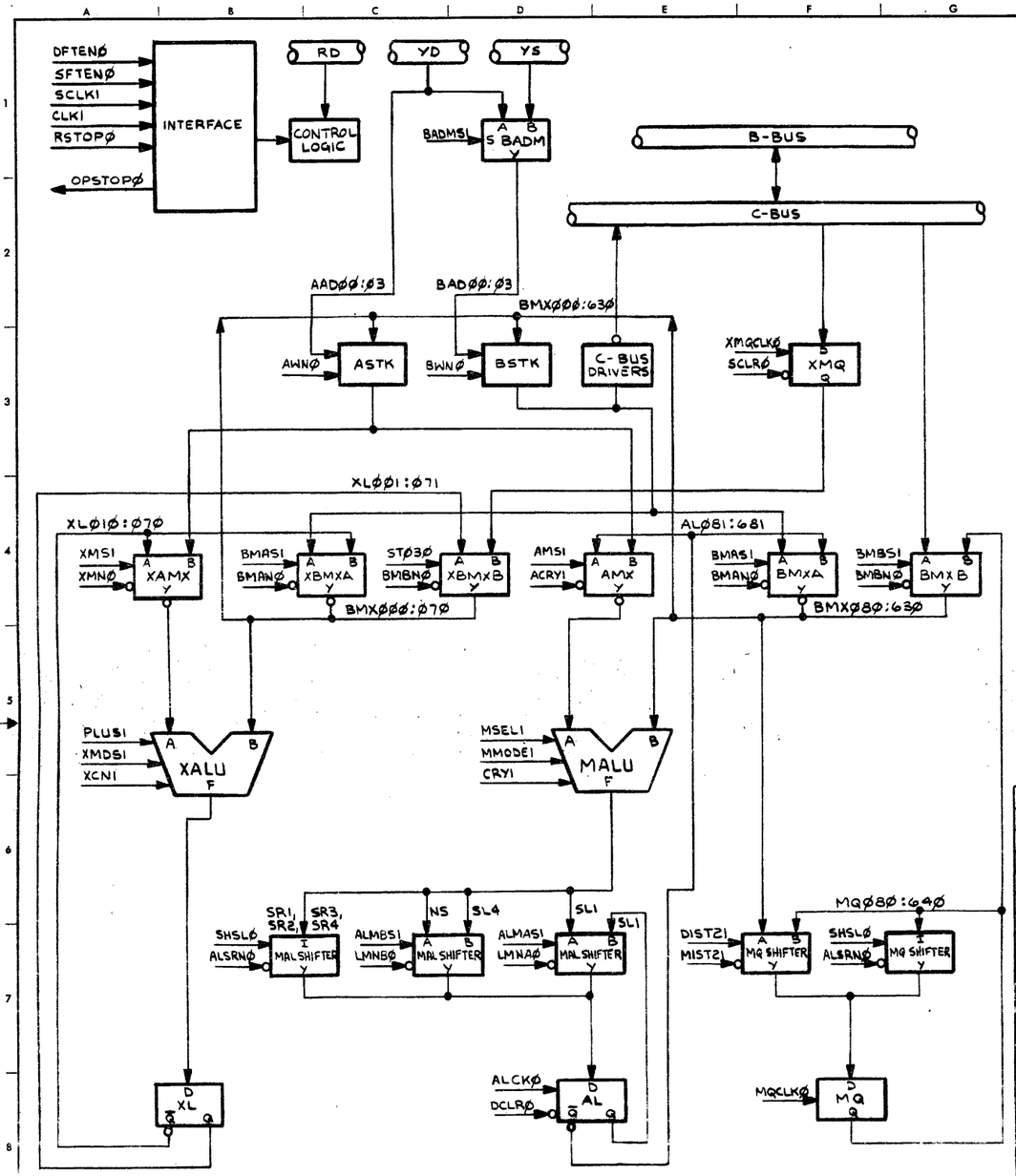
1. Install the HPFPP-A in Slot 02
2. Install the HPFPP-B in Slot 03
3. Install the C bus interface on the backpanel, Slot 3, Side 0
4. Install cable from HPFPP-A, Connector 3 to HPFPP-B, Connector 3
5. Install cable from HPFPP-A, Connector 4 to HPFPP-B, Connector 4



Front View of a 3230 Chassis

Figure J-1 3230 Board Installation and Cabling

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757



NOTE:
1. FOR NET#, MNEMONIC & LOCATION ON SCHEMATIC, SEE SHT. 20.

REF DESIGNATION	PART NUMBER	SPARE OUTPUT NUMBER
00M	19-055	11
01B	19-055	08,11
02B	19-055	08
06B	19-055	03
06D	19-055	08,11
00C	19-063	08/09
08A	19-063	08/09
05W	19-064	05/06
08M	19-064	07/09
09T	19-078	08
00S	19-125	03,08,11
01B	19-126	08
16F	19-126	06
04W	19-127	03,06
10E	19-127	08,11
00A	19-234	10,13
06W	19-234	10
07B	19-234	04

TABLE OF SPARES

CONN	TERM. NO.	CABLE CONNECTOR MAP ROW 2	CABLE CONNECTOR MAP ROW 1	BACK PANEL MAP ROW 1	BACK PANEL MAP ROW 2	TERM. NO.	CONN
5	16			PS	GND	41	
	15			GND	GND	40	
	14			AADØØ1	AADØ11	39	
	13			AADØ21	AADØ31	38	
	12			BADØØ1	BADØ11	37	
	11			BADØ21	BADØ31	36	
	10					35	
	09					34	
	08					33	
	07					32	
	06					31	
	05					30	
	04					29	
	03			CP32I	FPPFLTØ	28	
	02			RDI	NRLZI	27	
	01					26	
	00			MAL28Ø	MAL29Ø	25	
				MAL3ØØ	MAL31Ø	24	
				MAL32Ø	MAL33Ø	23	
				MAL34Ø	MAL35Ø	22	
4	24	GND	GND	GND	GND	21	1
	23	MCOUTI	CRY36I			20	
	22	CG32I				19	
	21	CGX2ØI	CPX2ØI			18	
	20	CRY2ØI	MOFFI			17	
	19	FCLKØ	DCLKØ	MQ28Ø	MQ29Ø	16	
	18	JACRYØ	JXCRYØ	MQ3ØØ	MQ31Ø	15	
	17	VGTØ	BGTØ	MQ32Ø		14	
	16	DBBØ	SBMBNØ			13	
	15	GND	AMXSØ	MSHØ	EQUØ	12	
	14	AEQBIA	AEQBIB	REGDISØ	RD25I	11	
	13			PSWI9I	DIST4Ø	10	
	12		SNRLZØ	AL32Ø	BMX32I	09	
	11	JBMXAS1	JBMXØ	MFNØ	RSTOPØ	08	
	10	GND	JVGTI			07	
	09	AMINBØ	APLSBØ			06	
	08	BONLYØ	AONLYI	SFTENØ	DFTENØ	05	
	07	XMQ3II	XMQ31Ø	RDØ9Ø	RDIØØ	04	
	06	XMQ3ØI	XMQ3ØØ	RDIIØ		03	
	05	BSTKWNI	ASTKWNI	LENBØ	DCLRI	02	
	04	GND		SKSIGØ	MQCEØ	01	
	03	ST03Ø	ST04Ø	GND	GND	00	
	02	ST03Ø	ST02Ø	PS	GND		
	01	ST01Ø	ST001				
	00	GND	GND	PS	GND	41	
				GND	GND	40	
				CLKI	OPSTOPØ	39	
					SCLKI	38	
				GND	GND	37	
	24	GND	GND	CØØØ	CØ1Ø	36	
	23	MØ7I	BMX641	CØ2Ø	CØ3Ø	35	
	22		JAMØ	CØ4Ø	CØ5Ø	34	
	21	MSUMI	GND	CØ6Ø	CØ7Ø	33	
	20			CØ8Ø	CØ9Ø	32	
	19			C1ØØ	C11Ø	31	
	18			C12Ø	C13Ø	30	
	17	SWNI		C14Ø	C15Ø	29	
	16			GND	GND	28	
	15					27	
	14					26	
	13					25	
3	12	MSCII	MSCØI			24	
	11					23	
	10					22	
	09	XAL Ø11				21	0
	08					20	
	07	WCLKI	EACLKØ			19	
	06	EXCLKØ	GND			18	
	05	XALØ7Ø	XALØ6Ø			17	
	04	XALØ5Ø	XALØ4Ø			16	
	03	XALØ3Ø	XALØ2Ø			15	
	02	JMQ31Ø	JMQ3ØØ			14	
	01	JMQ29Ø	JMQ28Ø			13	
	00	GND	GND			12	
				GND	GND	11	
				C16Ø	C17Ø	10	
				C18Ø	C19Ø	09	
	16			C2ØØ	C21Ø	08	
	15			C22Ø	C23Ø	07	
	14			C24Ø	C25Ø	06	
	13			C26Ø	C27Ø	05	
	12			C28Ø	C29Ø	04	
	11			C3ØØ	C31Ø	03	
	10				SCLRØA	02	
	09			GND	GND	01	
	08			PS	GND	00	
2	07						
	06						
	05						
	04						
	03						
	02						
	01						
	00						

THE REVISION LEVEL OF THIS SHEET IS CONSIDERED TO BE THE REVISION LEVEL OF THE DOCUMENT.

REVISIONS

PRE PRODUCTION APPROVAL	DEV	INIT.	DATE
		M	7/19

TABLE OF SPARE, 09K, 19-236, 08 WAS USED. SIGNAL REGDISØ ON CONN 1 ROW 1, PIN 12 WAS NOT SHOWN. TITLES ON SHEETS 1-20 WERE DFU-A.

DS RC ECP 3950 | 3-19-79 | R01

IN TABLE OF SPARES, 16B, 19-234, 10 WAS SPECIFIED. SHEETS 6, 10, & 20 WERE SPEC. AS R01.

DS WP 4020 | 5-3-79 | R02

RELEASED FOR PRODUCTION
ENG. _____ DATE 5/10/79

CABLE CONNECTOR MAP: CONN 4 PIN 14 WAS AEQBIC, CONN 3 PIN 9 WAS NOT SPEC. AREA B5 & D5 XALU & MALU WERE SHOWN INCORRECT. REVISED SHT. 3, 6, 8, 11 & 20. REMOVED SPARE OUTPUTS 11B-10, 00K-08, & 07B-04, AREA 58 35-715 WAS R03

KR 4096 | M | 7-17-79 | R03

REVISED SHT'S. 4 & 8 AREA 58 35-715 WAS R04

ET 4301 | M | 2-19-80 | R04

REVISED SHTS 1, 2, 9, 11 & 20 AREA N2: ADDED FPPFLTØ TO TERM 28. AREA R8, 35-715 WAS R05. RMVD. 07W08 (19-057) & 09M08 (19-058) FROM TABLE OF SPARES

JAT 4491 | MS | 10-7-80 | R05

REVISED SHT 1, IN AREA C7 & D7, ALMBSI & ABMASI AND LMNBØ & LMNAØ WERE REVERSED

VT LI9009-15 | R | 11-17-80 | R06

REVISED SHT 12.

JAT LI9013-05 | 4-15-81 | R07

DELETED "MODEL 3220" FROM TITLE BLOCK. RVS'D SHTS 1-20

JLV 5065 | R | 7-30-82 | R08

35-715 | R07
BOARD | REV LEVEL

BOARDS AGREEING WITH THIS SCHEMATIC MUST BE AT LEAST THE FOLLOWING REVISION LEVEL.

USED IN MANUAL: 29-705

REVISION	8	3	3	3	2	4	2	4	3	3	4	3	2	2	2	2	2	2	5
SHEET	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

SCALE——
TOLERANCE: XX ± .01, XXX ± .02, X ± .03, ANGLES ± .5°, UNLESS OTHERWISE SPECIFIED

C. MICHAELS	DRAFT	9-18-78
N. LEMONDE	MGR	5-11-79
R. CERO	CHK	5-11-79
N. SELHEIM	ENGR	5-11-79
R BARKER	QC	5-11-79
E. GREENSTEIN	SYSTEST	5-11-79

TITLE: FUNCTIONAL SCHEMATICS HPFPP-A

TASK NO. 03905 | SHEET OF 1 — 20

SYS NO. 35-715 R08 D08

BRUNING 44-231 24538

PERKIN ELMER
Computer Systems Division
Oceanport, N.J 07757

REVISIONS
GATES 00M & 10M WERE SHOWN AS A NAND GATES. IC 03W-02, 04, 06, 08, & 10 AND 07W-02 BALL WAS ON OUTPUT.
IC 08T & 05W-D5, Q, $\bar{Q}$ & DC WERE 5, 1, 0 & R RESPECTIVLY. TITLE WAS DFU-A.
DS | VER 3450 | 3-19-79 | R01
AREA M6: ADDED 9L1 TO RCCØ
JAT & 44911 | MS | 10-7-80 | R02
TITLE WAS MOD 3220
JLV | 5065 | R | 8-1-82 | R03 X

206-1 DFTENØ → 03W 19-057 SG → SPØ → 3A3, 6A1, 8A9

00M 19-055 SG

106-1 SFTENØ → 03W 19-057 SG

P5 → 45 02C 24 → 1K

3N1 STØØ1 → 10M 19-055 SG

238-0 SCLK1 → 07W 19-057 SG

08T 19-063 SDFF

3N3 STØ1ØA

139-Ø CLK1

2Ø9-1 RSTOPØ → 09K 19-230 SA

05W 19-064 SFF

19M7 BSCLRØB → 09T 19-078 SGO → OPSTOPØ (239-0

SP1 → 3A2, 6A1, 7K3

SEL1 → 5B4

XPB! → 8H7, 9H7

3N2 STØØØ → 10M 19-055 SG

DCLR1 (203-1

08S 19-124 SB → DCLRØA → 3A1, 8A8, 6H5, 7K4

08S 19-124 SB → DCLRØB → 9H7, 10K4, 13K9, 14K9, 15K9, 16K9, 17K9, 18K9, 6A6, 12L4, 11L4

07W 19-057 SG → SSC1 → 8A2, 8H6

08W 19-060 SG → SSCØ → 3N2

08W 19-060 SG → RCCØ → 9N2, 9L

00T 19-057 SG

105-1 RDØ9Ø → 03W 19-057 SG → FØØ1 → 3A4, 5B3, 9L6

205-1 RD1ØØ → 03W 19-057 SG → FØ11 → 3A4, 5B3, 9L6

06W 19-234 SN

104-1 RD11Ø → 03W 19-057 SG → FØ21 → 3A4, 5B3, 9L6

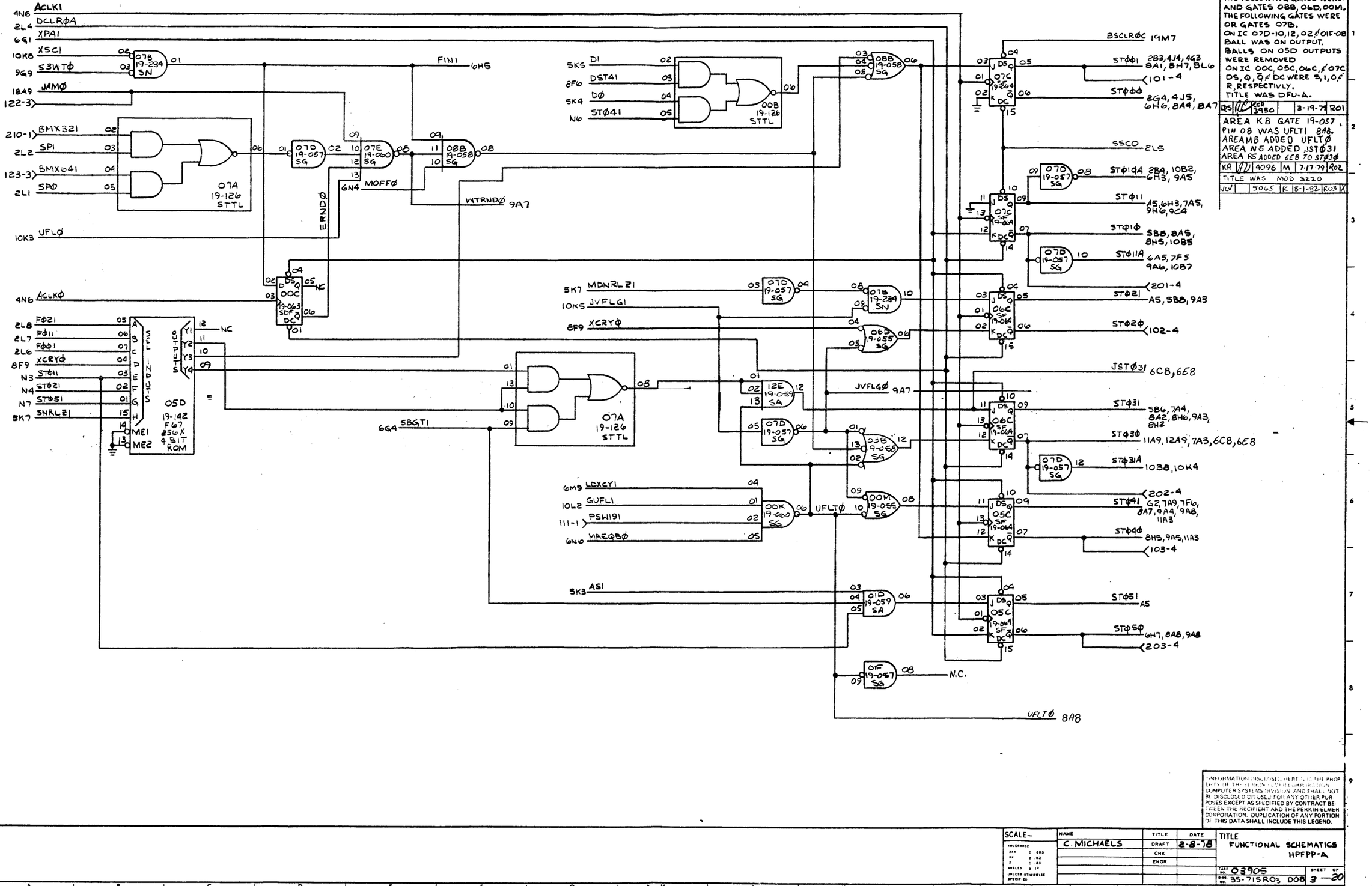
SCALE—	NAME	TITLE	DATE
TOLERANCE XXX ± .005, XX ± .02, X ± .09, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-8-78
		CHK	
		ENGR	

TITLE: FUNCTIONAL SCHEMATICS HPFPP-A

TASK NO. 03905 | DWG NO. 38-718 R03 D08 | SHEET OF 2—20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS				
THE FOLLOWING GATES WERE AND GATES 08B, 06D, 00M. THE FOLLOWING GATES WERE OR GATES 07B. ON IC 07D-10,12, 02 & 01F-08 BALL WAS ON OUTPUT. BALLS ON 05D OUTPUTS WERE REMOVED ON IC 00C, 05C, 06C, & 07C DS, Q, $\overline{Q}$ & DC WERE S, 1, 0 & R, RESPECTIVLY. TITLE WAS DFU-A.				
RS		3950		3-19-79 R01
AREA K8 GATE 19-057 PIN 08 WAS UFLTI 8A8. AREA M8 ADDED UFLT Ø AREA N5 ADDED JST Ø31 AREA R5 ADDED 6E8 TO ST Ø3 Ø				
KR		4096	M	7-17-79 R02
TITLE WAS MOD 3220				
JLV		5065	R	8-1-82 R03 X



SCALE–	NAME	TITLE	DATE
TOLERANCE XXX ± .003, XX ± .02, X ± .03, ANGLES ± 1° UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-8-78
		CHK	
		ENGR	

TITLE FUNCTIONAL SCHEMATICS HPFPP-A

TASK NO. 03905 SHEET OF 3 — 20

DWG NO. 35-715R03 DOB

PERKIN ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS			
ID TO16R AREA E2 WAS NOT SHOWN. THE FOLLOWING WERE SHOWN AS AND GATES-12A, 13B, 11A. 16B WAS SHOWN AS AN OR GATE. ON IC 11B-09 & 06 BALL WAS ON OUTPUTS TITLE WAS DFU-A.			
DS	ECR 3350	3-19-79	R01
AREA N5, WCLK1 ADDED REF. DES. 8A9. AREA N6, ACLK1 DELETED REF DES. 8A9			
ZT	43011M	2-19-80	R02
DELETED 'MODEL 3220' FROM TITLE			
HW	5065	R 8-1-82	R03 X

SINGLE STEP SWITCH
16R 33-034
14A 1K
12A 19-055 SG
12B 19-044 SF
13A 19-130 SB
13B 19-058 SG
16B 19-234 SN
11A 19-124 SB
11B 19-057 SG
13A 30-019 50 NS DELAY LINE
16A 33-035
14A 220
14A 330
14B 30-013 4.7 uH
14B 22-018 9-35 pF
XPA1 6G1
3N1 STΦΦ1
3N2 STΦΦ
206-3 EXCLKΦ
WCLK1 9A7, 8A9
207-3
EACLKΦ 8A9, 8H7, 8D8, 8A7
107-3
ACLK1 3A1, 6A4, 10K7
ACLKΦ 3A4, 6H5, 7K5, 8H9, 9H4, 11L4, 12L4
ALCKΦ 13K9, 14K9, 15K9, 16K9, 17K9, 18K9
ACLK1A 8H7, 9H6, 10H5
T.P.1 100-2
6N3 SLWCLKΦ

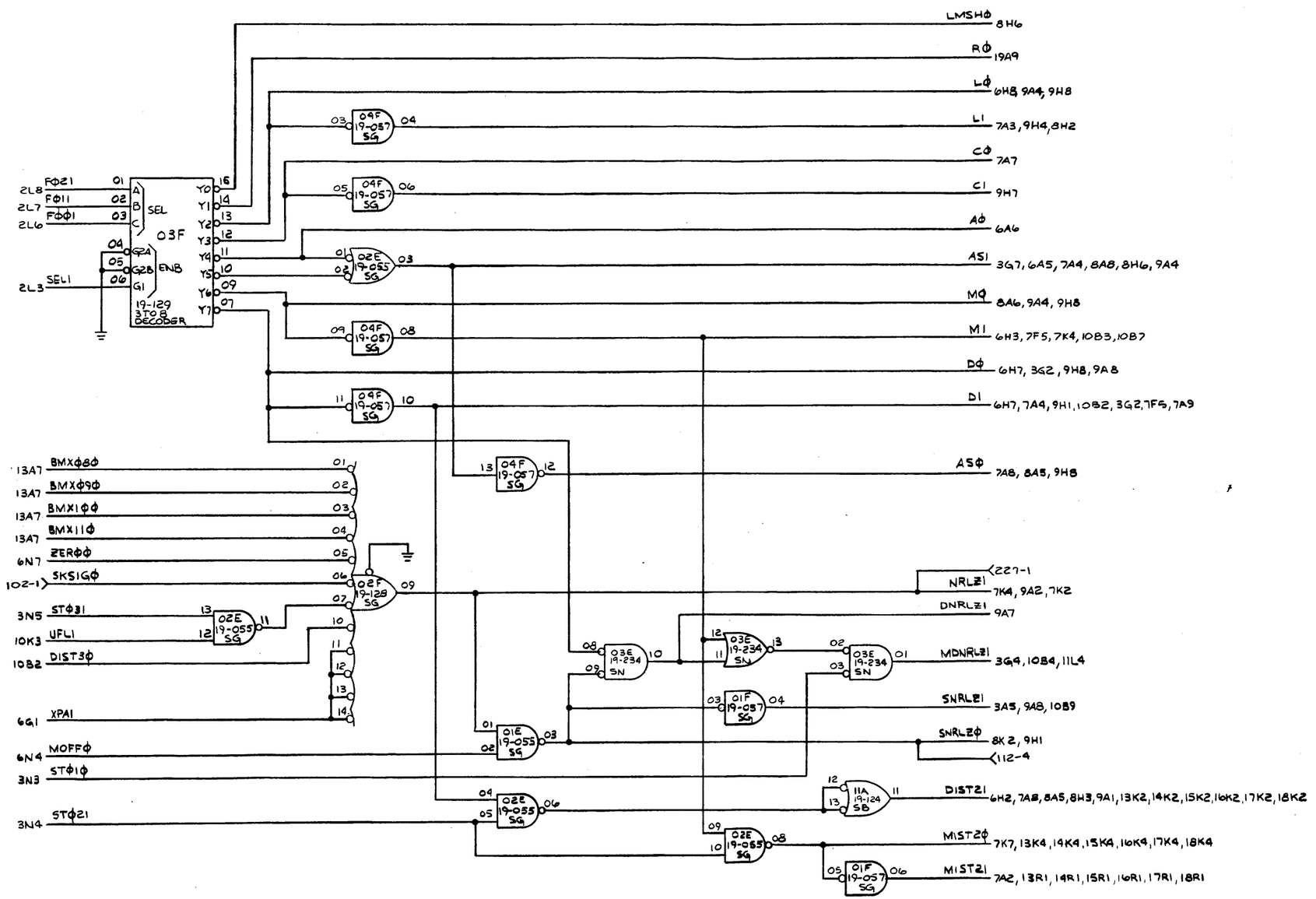
	SWITCH NO.							
	1	2	3	4	5	6	7	8
FAST CLOCK (95 ns)	ON	OFF	X	X	X	X	ON	OFF
NOM CLOCK (100 ns)	OFF	ON	X	X	X	X	ON	OFF
SINGLE-STEP	(SET FOR NOM OR FAST)						OFF	OFF
EXTERNAL CLOCK	X	X	X	X	X	X	X	ON

SCALE-	NAME	TITLE	DATE
TOLERANCE XXX ± .005, XX ± .02, X ± .03, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-8-78
		CHK	
		ENGR	

TITLE: FUNCTIONAL SCHEMATICS HPFPP-A
TASK NO. 03905
DWG NO. 35-715R03 D08
SHEET 4 OF 20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS				
THE FOLLOWING WERE SHOWN AS AND GATES; 02E & 02F. THE FOLLOWING WERE SHOWN AS OR GATES; 03E & 11A. ON IC 04F04, 06, 08, 10 & 01F04, 06 BALL WAS ON OUTPUT. TITLE WAS DFU-A				
DS	ECR 3950		3-26-79	R01
DELETED "MOD 3220" FROM TITLE				
JLV		5065	R	8-1-82	R02



SCALE–	NAME	TITLE	DATE
TOLERANCE .XXX ± .005, .XX ± .02, .X ± .03, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-8-78
		CHK	
		ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905
DWG. NO. 35-715R02 D08
SHEET OF 5 — 20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS			
THE FOLLOWING WERE SHOWN AS AND GATES; 03B06, 02B03, 01A03 & 02B11. THE FOLLOWING WERE SHOWN AS OR GATES; 00A01, 04. IN AREA J 3 GATE 09K08 WAS NOT SHOWN, SIG. LOADØ WENT TO 01A12, SIG. GDCLKØ WAS NOT SHOWN. ON IC 04B, 04C & 03C - DS, Q, Q̄, & DC WERE S, 1, 0, & R RESPECTIVLY. TITLE WAS DFU-A.			
DS	ECR 3950	3-26-79	R01
IN AREA B8 CROSS REF 10B2 ON ZERO1 WAS NOT SPEC.			
DS	ECR 4020	5-9-79	R02
AREA F8 ADDED 01B 19-126 STTL, AREA J6 ADDED R03A & R03B AREA H6 AEQBIA WAS AEQBIC			
KR	4096 M	7-17-79	R03
DELETED MODEL 3220 FROM TITLE			
JLV	5005 R	8-1-82	R04

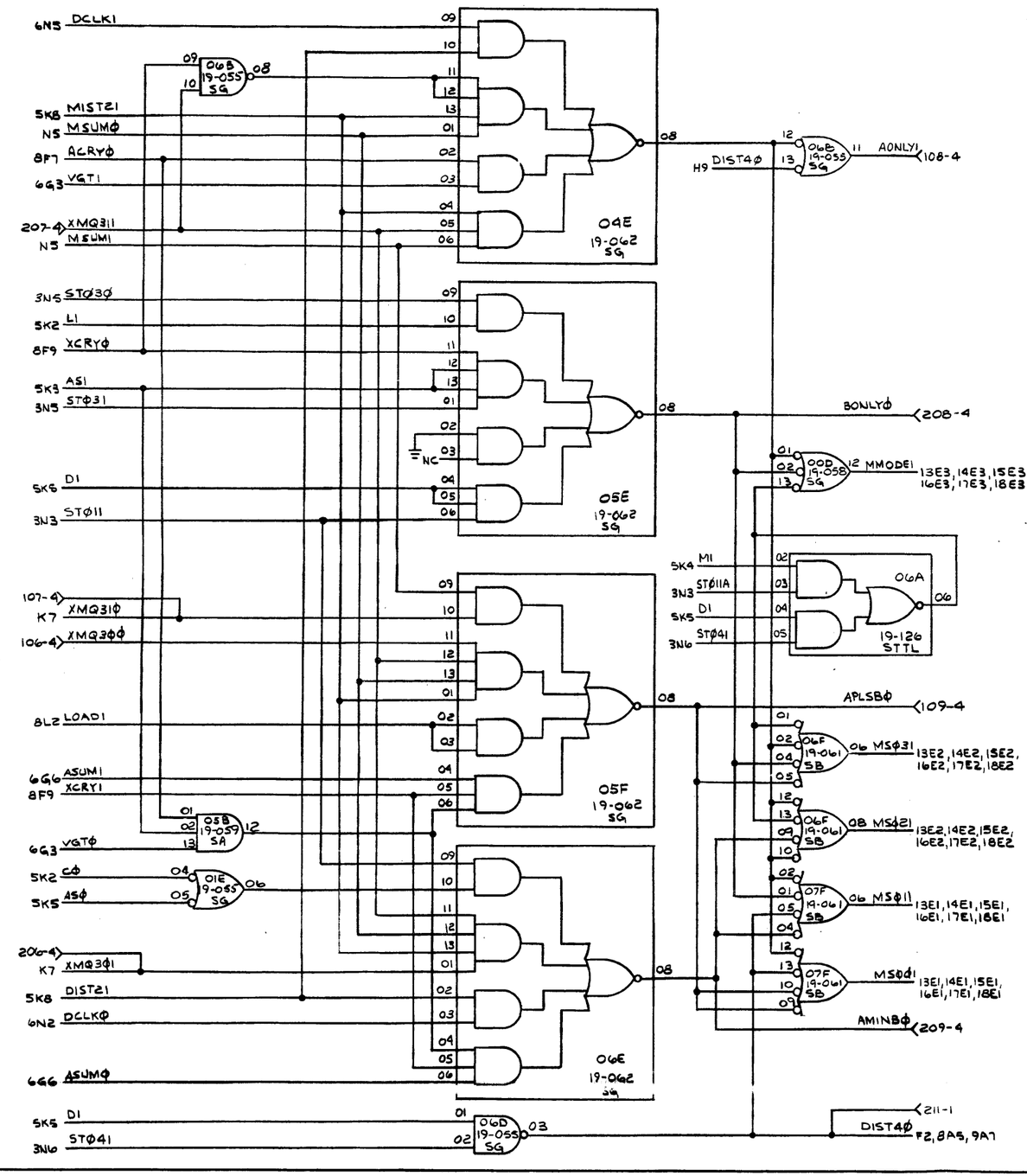
SCALE—	NAME	TITLE	DATE
TOLERANCE .XXX ± .005, .XX ± .02, .X ± .125, ANGLES ± 1° UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	
		CHK	
		ENGR	

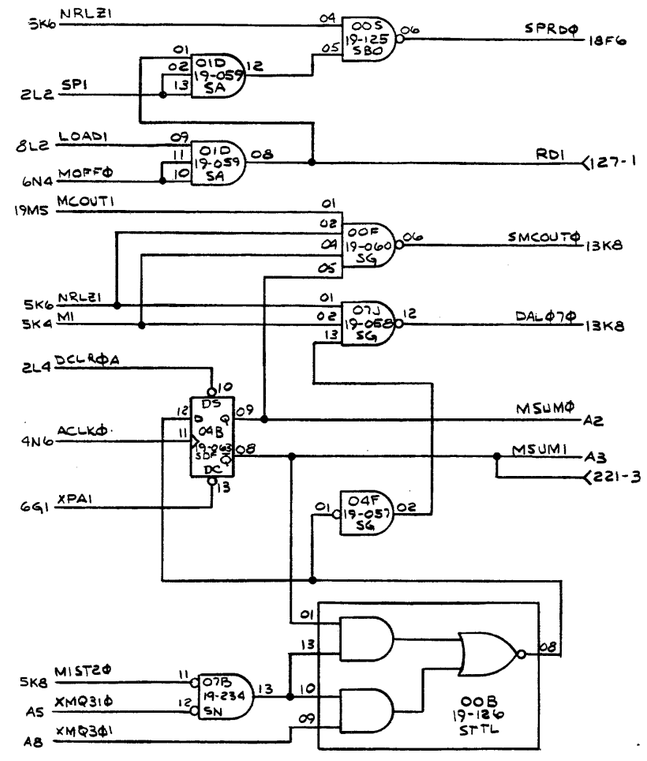
TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905 | SHEET 6 OF 20
DWG NO. 35-715 R04 D08

PERKIN ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS					
THE FOLLOWING WERE SHOWN AS AND GATES; 01E06, 06B11, 00D12, 06F06, 08 07F06, 08, & 07B13. ON IC 04B DS, Q, & DC WERE S, 1, 0, & R, RESPECT'LY ON IC 04F02 BALL WAS ON OUTPUT. TITLE WAS DFU-A					
DS		ECB 3950		3-26-79	R01
DELETED "MODEL 3220" FROM TITLE.					
JLV		5065	R	8-1-82	R02





SCALE–	NAME	TITLE	DATE
TOLERANCE .XXX ± .005 .XX ± .02 .X ± .09 ANGLES ± 1° UNLESS OTHERWISE SPECIFIED	C MICHAELS	DRAFT	2-9-78
		CHK	
		ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905
DWG NO. 35-715R02 D08
SHEET OF 7 — 20

PERKIN ELMER
Computer Systems Division
Oceanport, N.J. 07757

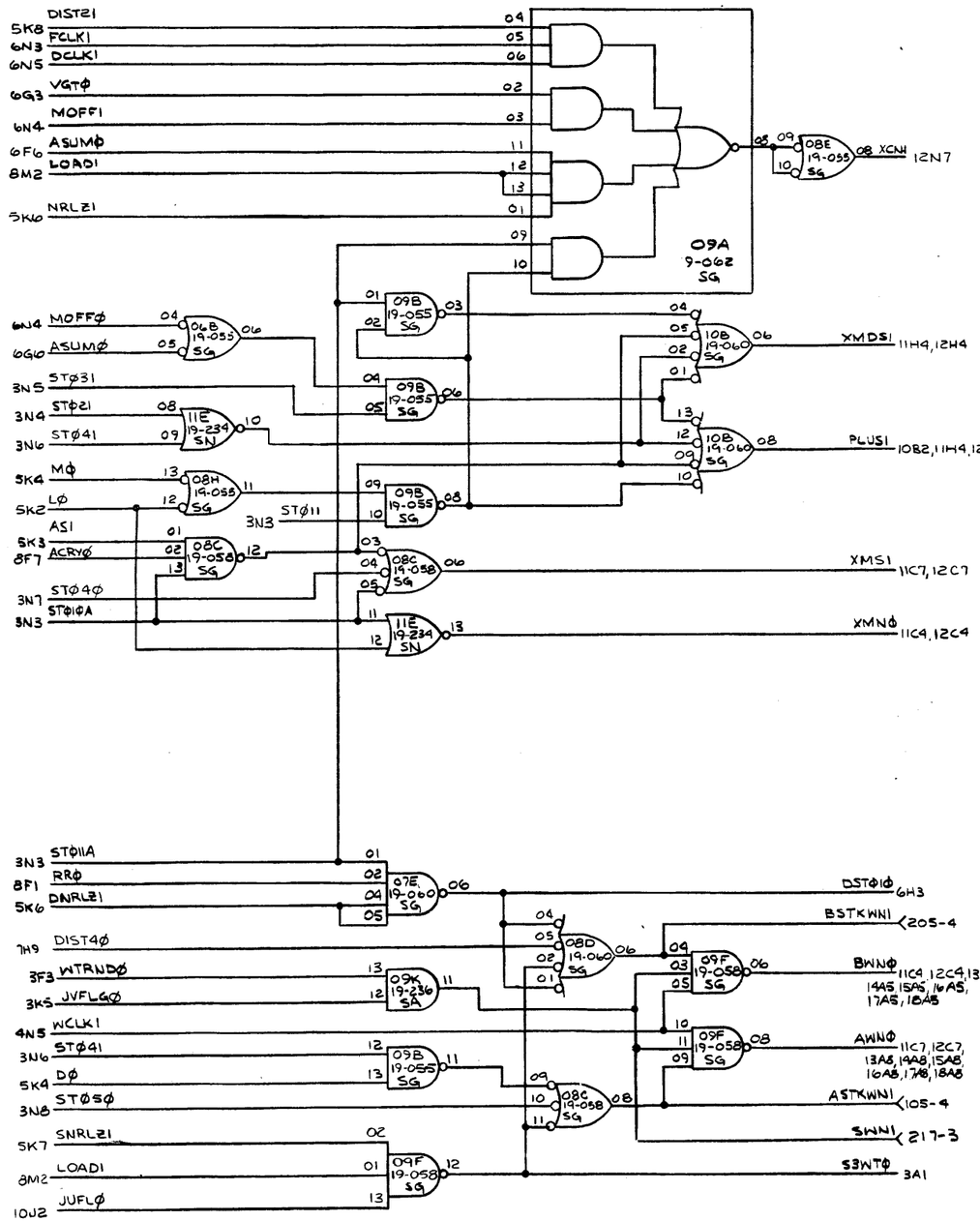
REVISIONS
THE FOLLOWING WERE SHOWN AS AND GATES 08J03, 05B08,06, 00F08, 08J06,08 08E11, 08H08, 09K03, 08J11 07K06,12, & 00D08. THE FOLLOWING WERE SHOWN AS OR GATES: 03E04, 07B04. SIGNAL GDCLK0 IN AREA A6 WAS NOT SHOWN. ON IC 07H, 08A, 03C, & 08T DS, Q, $\overline{Q}$, & DC WERE S, 1, 0, R - RESPECTIVLY. ON IC 08F08, & 01F02,10 BALL WAS SHOWN ON OUTPUT. TITLE WAS DFU-A
DS ECP 3950 3-26-79 R01
AREA B8 REMOVED 07B 19-234 SN, AREA C8 00K WAS 03A 19-058
KR SV 4096 M 7-17-79 R02
AREA A9 $\S$ B9, 00D05 WAS MNEMONIC - ACLK1
ET W 4301 M 2-19-80 R03
DELETED "MODEL 3220" FROM TITLE
JLV 5065 K 8-1-82 R04

SCALE–	NAME	TITLE	DATE	TITLE
TOLERANCE XXX ± .005, XX ± .02, X ± .08, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-9-78	FUNCTIONAL SCHEMATICS HPFPP-A
		CHK		
		ENGR		
				TASK NO. 03905 SHEET OF
				DWG NO. 35715 R04 D08 8 — 20

REVISIONS

THE FOLLOWING WERE SHOWN AS AND GATES; 06B04, 08H11, 08C06, 10B08,06 08E08, 08D06, 08C08, 07M06, 10M06, 08D08, 09M06, & 13B08.
THE FOLLOWING WERE SHOWN AS OR GATES; 06W04
ON 09S BALLS WERE SHOWN ON OUTPUTS.
ON 08M DS, Q, $\overline{Q}$, & DC WERE S, I, O, & R - RESPECTIVLY
TITLE WAS DFU-A

RS	ECR 3950		3-27-79	R01
AREA M152: ADDED I.C.'S 07M 5 09M FROM 10T - 06				2
JAT	W49	MS	10-7-80	2
DELETED "MODEL 3220" FROM TITLE				
4W	5065	R	8-1-82	R05



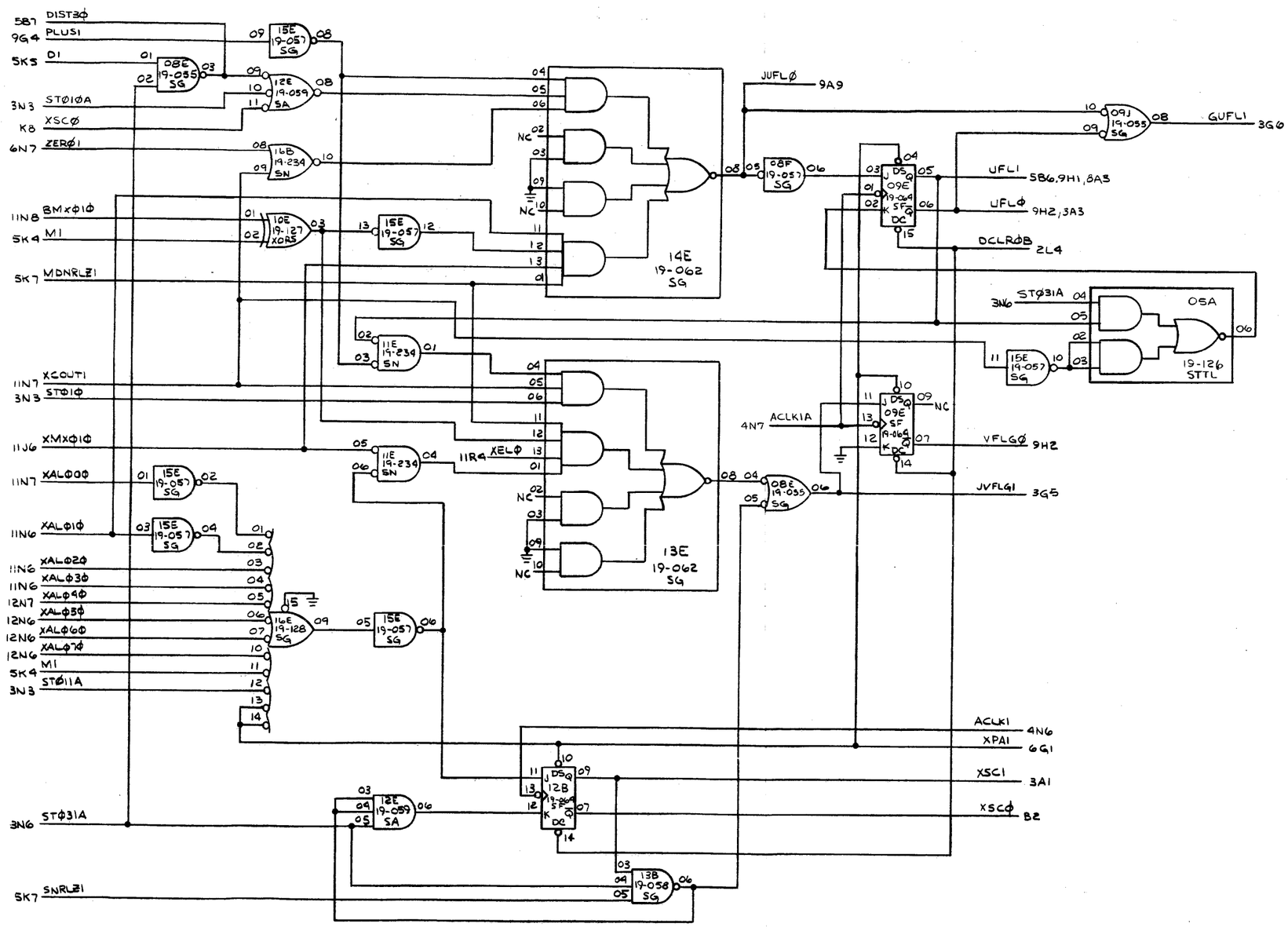
SCALE–	NAME	TITLE	DATE
TOLERANCE XXX ± .005, XX ± .02, X ± .03, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-9-78
		CHK	
		ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905
P.N. NO. 35-715 R03 DOC
SHEET 9 OF 20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

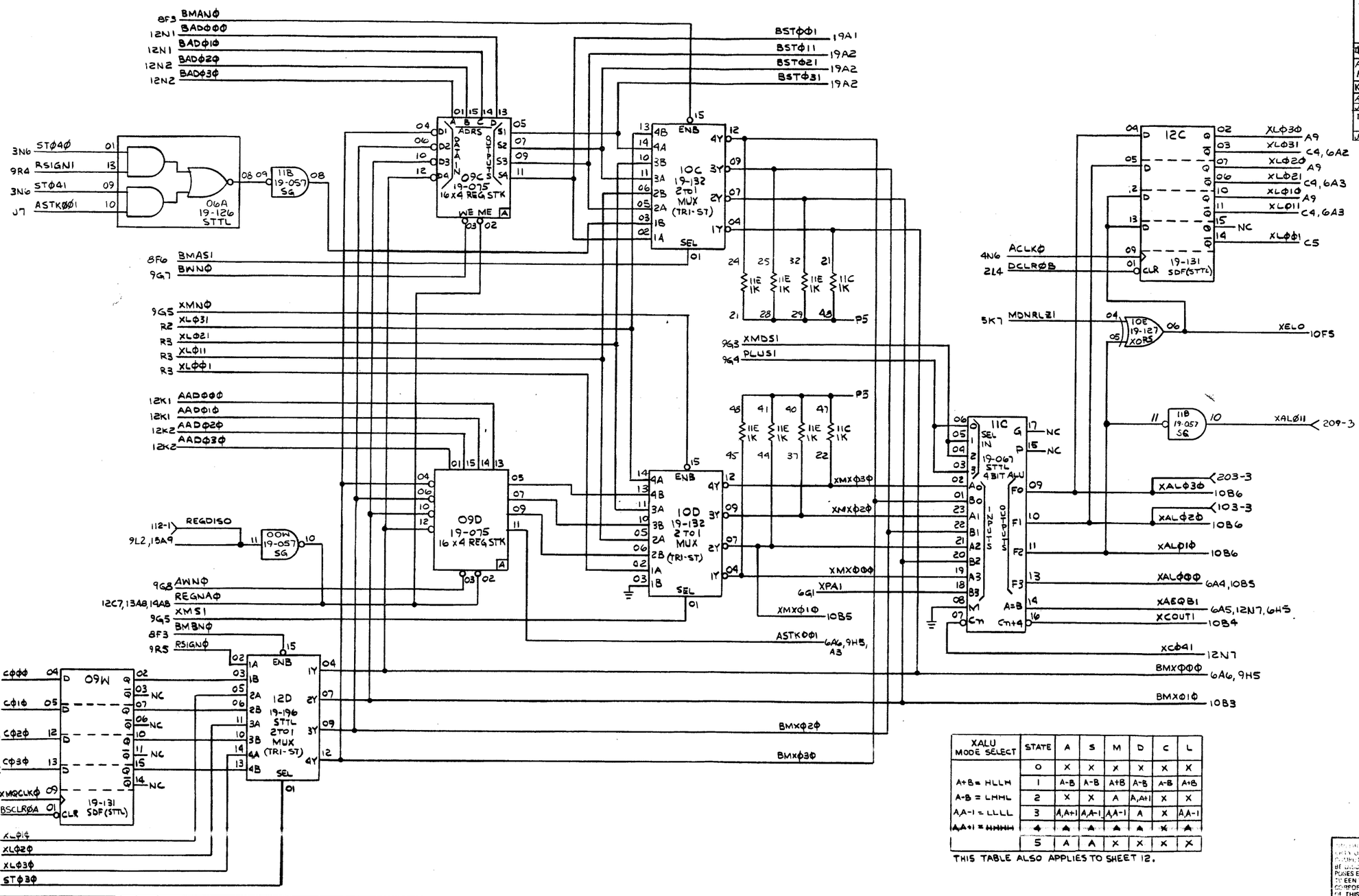
REVISIONS				
THE FOLLOWING WERE SHOWN AS AND GATES; 12E08, 16B09, 09J08, 08E06. THE FOLLOWING WERE SHOWN AS OR GATES; 11E01, 04. ON IC'S 15E12, 08E06 BUBBLE WAS SHOWN ON OUTPUTS. ON IC'S 09E, 12B DS, Q, 8E DC WERE 5, 1, 0 R- RESPECTIVELY. TITLE WAS DFU-A				
Ø5		3950		3-27-79 R01
AREA D3, GATE 16B10 WAS 15E10, CONN. TO 14E06. AREA B2, SIG. ZERØ1 WAS NOT SPEC. AREA K4 GATE 15E10 WAS NOT SHOWN.				
Ø5	D.P.	4020		5-4-79 R02
DELETED "MODEL 3220" FROM TITLE BLOCK				
JLV		5065	P	8-1-82 R03



SCALE–	NAME	TITLE	DATE	TITLE
TOLERANCE XXX ± .005, XX ± .02, X ± .03, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-9-78	FUNCTIONAL SCHEMATICS HPFPP-A
		CHK		
		ENGR		
				TAPE NO. 03905
				DWG NO. 35-715 R03 D08
				SHEET OF 10—20

PERKIN ELMER
Computer Systems Division
Oceanport, N.J. 07757.

REVISIONS				
ON IC'S 11B08, 09C, 09D BUBBLES WERE ON OUTPUTS AREA C6 DELETED 1K RESISTOR BETWEEN 02T10 & 02T07. 02T10 WAS CONN. TO P5, 02T07 WENT TO 00W11. SIGNAL REGDIS0 WAS NOT SHOWN. TITLE WAS DFU-A.				
RS	PC	3950	3-27-79	R01
AREA M5 ADDED GATE 11B 19-057 SG				
KR	RT	4096	M 7-17-79	R02
AREA C6, ADDED 9L2 TO REGDISQ				
JAM-#1	4491	MS1	10-7-80	R03
DELETED "MODEL 3220" FROM TITLE BLOCK				
JLV	5065	R	8-1-82	R04 X



XALU MODE SELECT	STATE	A	S	M	D	C	L
	0	X	X	X	X	X	X
A+B = HLLH	1	A-B	A-B	A+B	A-B	A-B	A+B
A-B = LHHL	2	X	X	A	A,A+1	X	X
A,A-1 = LLLL	3	A,A+1	A,A-1	A,A-1	A	X	A,A-1
A,A+1 = HHHH	4	▲	▲	▲	▲	X	▲
	5	A	A	X	X	X	X

THIS TABLE ALSO APPLIES TO SHEET 12.

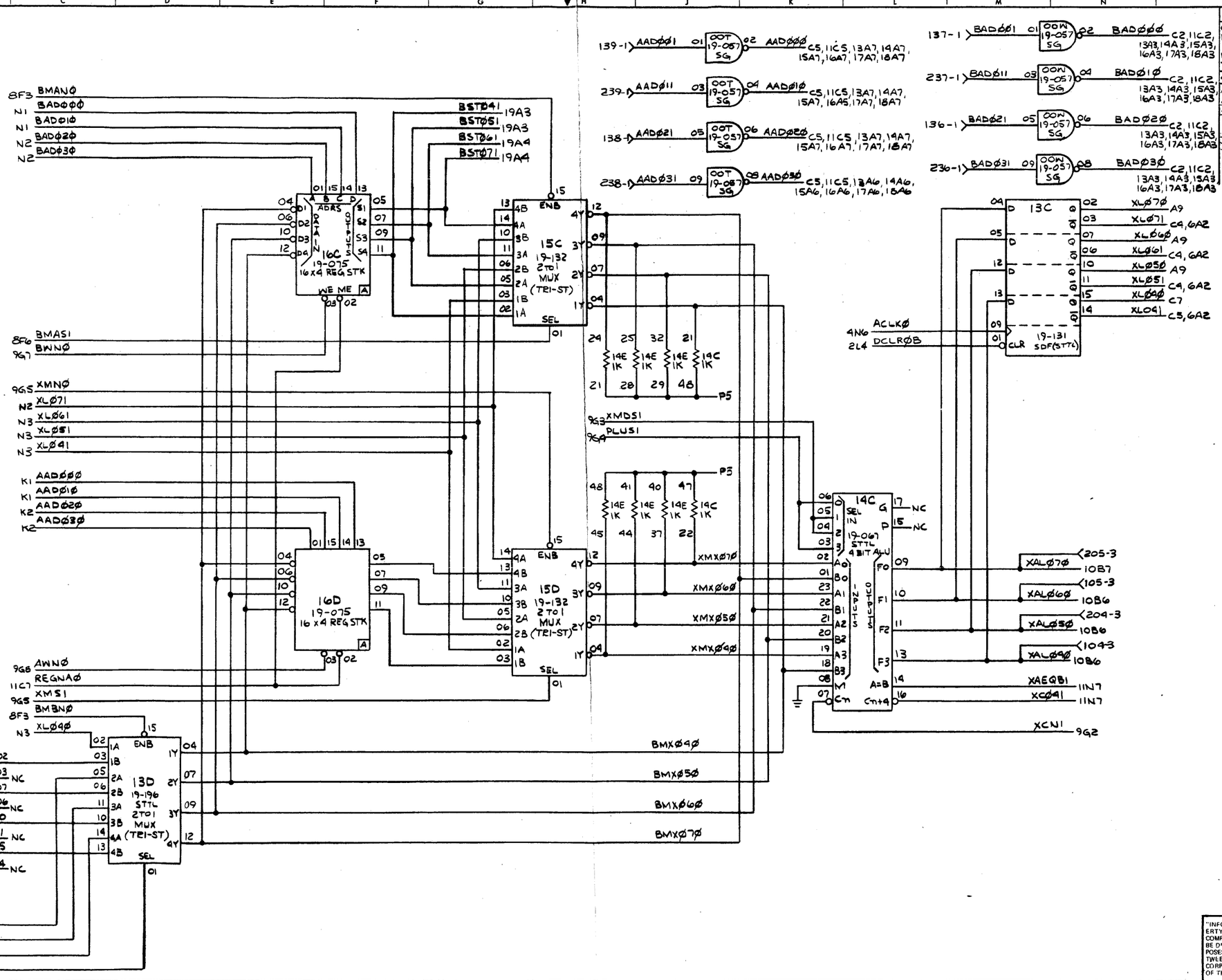
SCALE—	NAME	TITLE	DATE
TOLERANCE XXX ± .005, XX ± .02, X ± .03, ANGLES ± 1° UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-9-78
		CHK	
		ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905
DWG NO. 35-715 R04 D08
SHEET 11 OF 20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS				
ON IC 16C & 16D BUBBLES WERE SHOWN ON OUTPUT SIDE. ON IC 13 BUBBLES WERE NOT SPEC. DWG TITLE WAS DFU-A.				
RE	RP	3950	3-27-79	E01
AREA D8 : REMOVED BUBBLES FROM 13D-04,07,09&12				
JAT	KR	L19013-05	4-15-21	B02
DELETED "MODEL 3220" FROM TITLE BOX				
LW		5065	R 8-1-82	R03 X



NOTES: 1. FOR XALU MODE SELECT TABLE REFER TO PAGE 11.

SCALE—	NAME	TITLE	DATE	TITLE
TOLERANCE .XXX ± .005, .XX ± .02, .X ± .03, ANGLES ± 1° UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-9-78	FUNCTIONAL SCHEMATICS HPFPP-A
		CHK		
		ENGR		
				TASK NO 03905
				DWG. NO. 35-715 E03 D08
				SHEET OF 12 — 20

2 Crescent Place, Oceanport, New Jersey 07757

REVISIONS			
THE FOLLOWING WAS SHOWN AS A AND GATE; 07K08. ON IC'S 07S & 07T BUBBLES WERE SHOW ON OUTPUTS TITLE WAS DFU-A			
RBS	3950	3-27-79	R01
DELETED "MODEL 3220" FROM TITLE BOX			
JLW	5065	8-1-82	R02

FUNCTION	MS@01	MS@11	MS@21	MS@31	MMODE1
ADNLY1	1	1	1	1	1
BONLY1	Ø	1	Ø	1	1
APLSB1	1	Ø	0	1	Ø
AMINB1	Ø	1	1	Ø	Ø

4 BIT SHIFTER SHIFT CODE		
SHSL1Ø	SHSLØØ	
Ø	Ø	SR4
Ø	1	SR3
1	Ø	SR2
1	1	SR1

SCALE–	NAME	TITLE	DATE
TOLERANCE XXX ± .005, XX ± .02, X ± .03, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED	C.MICHAELS	DRAFT	2-9-78
		CHK	
		ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905
DWG NO. 35-715R02-008
SHEET OF 13–20

REVISIONS			
ON IC 01W & 02W BUBBLES WERE SHOWN ON OUTPUT SIDE. DWG. TITLE WAS DFU-A.			
JLV	3950	3-27-79	R01
DELETED "MODEL 3220" FROM TITLE BLOCK			
JLV	5065	R 8-1-82	R02

SCALE –

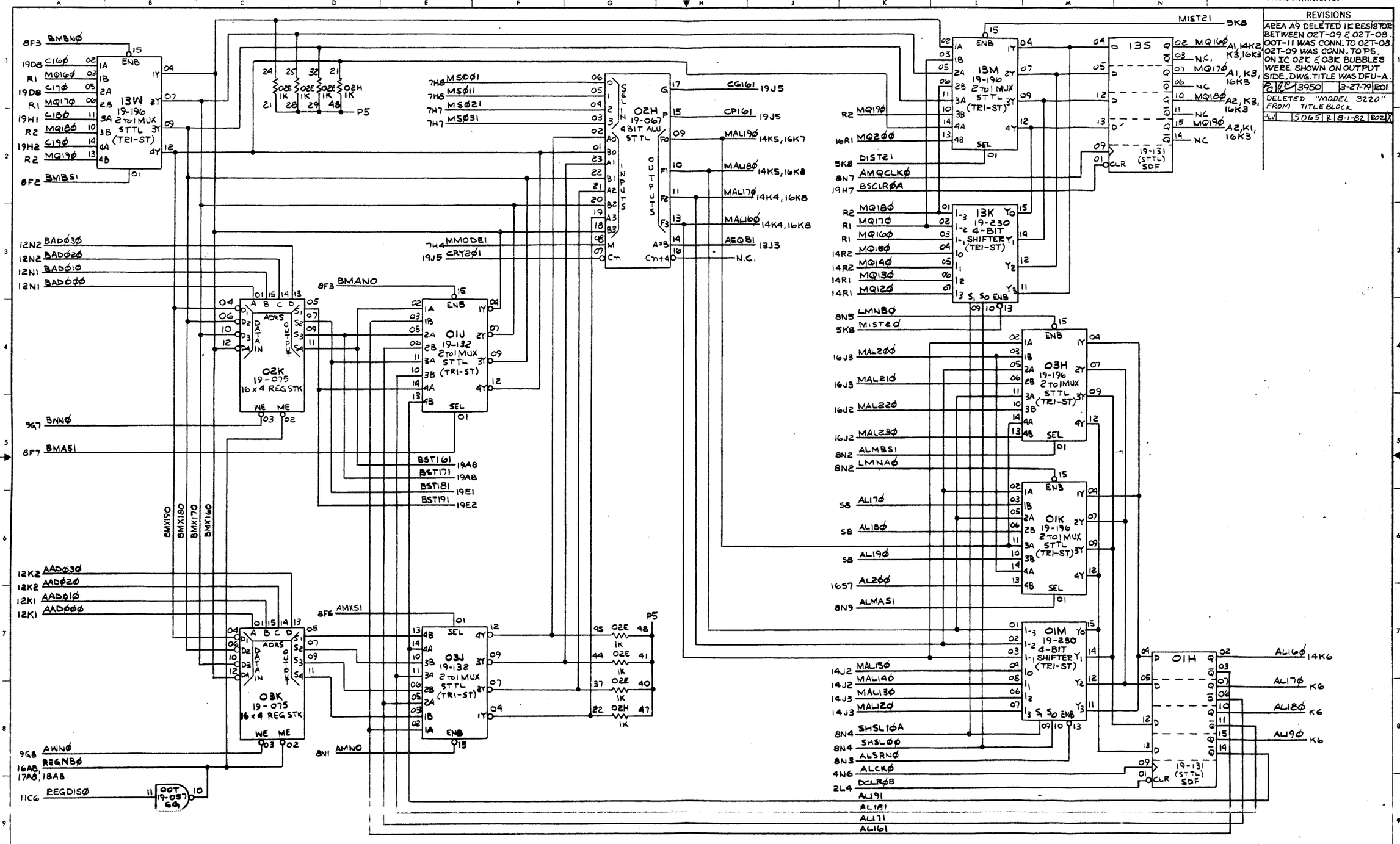
NAME	TITLE	DATE
C. MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905
DWG NO. 35-715 B02 D08
SHEET 14 OF 20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS			
AREA A9 DELETED IC RESISTOR BETWEEN 02T-09 & 02T-08. 00T-11 WAS CONN. TO 02T-08. 02T-09 WAS CONN. TO P5. ON IC 02E & 03K BUBBLES WERE SHOWN ON OUTPUT SIDE. DWG. TITLE WAS DFU-A.			
PEM	3950	3-27-79	E01
DELETED "MODEL 3220" FROM TITLE BLOCK.			
JLV	5065	R 8-1-82	R02 X



SCALE—

TOLERANCE
XXX ± .005
XX ± .02
X ± .03
ANGLES ± 1°
UNLESS OTHERWISE SPECIFIED

NAME	TITLE	DATE
C. MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905
DWG NO. 36-715 B02 D08
SHEET 15 OF 20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS			
ON IC 04K & 06M BUBBLES WERE SHOWN ON OUTPUT SIDE. DWG TITLE WAS DFU-A			
P	C	3950	3-27-79 R01
DELETED "MODEL 3220" FROM TITLE BLOCK			
KV	5065	R	8-1-82 R02

"INFORMATION DISCLOSED HEREIN IS THE PROPERTY OF THE PERKIN-ELMER CORPORATION, COMPUTER SYSTEMS DIVISION, AND SHALL NOT BE DISCLOSED OR USED FOR ANY OTHER PURPOSES EXCEPT AS SPECIFIED BY CONTRACT BETWEEN THE RECIPIENT AND THE PERKIN ELMER CORPORATION. DUPLICATION OF ANY PORTION OF THIS DATA SHALL INCLUDE THIS LEGEND.

SCALE—

NAME	TITLE	DATE
C. MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905
DWG NO. 33-715 R02 D08
SHEET OF 16—20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS			
ON IC 12J & 10J BUBBLES WERE SHOWN ON OUTPUTS ONLY TITLE WAS DFU-A			
EJ	3950	3-27-79	R01
DELETED "MODEL 3220" FROM TITLE BLOCK			
JLV	5065	8-1-82	R02

8F3 BMBNϕ → 15W 19-196 STTL 2 TO 1 MUX (TRI-ST) ENB, 1A 1B 2A 2B 3A 3B 4A 4B, 1Y 2Y 3Y 4Y, SEL
19H5 C29ϕ 19H5 MQ29ϕ 19H5 C25ϕ R1 MQ25ϕ 19H5 C26ϕ R2 MQ26ϕ 19H6 C27ϕ R2 MQ27ϕ
8F2 BMBS1
24 25 32 21 11K 1K 11K 1K 11K 1K 11M 1K 21 28 29 48 → P5
7H8 MS06ϕ1 7H8 MS01ϕ1 7H7 MS02ϕ1 7H7 MS03ϕ1
11H 19-067 4 BIT ALU STTL: A0 B0 A1 B1 A2 B2 A3 B3 M Cn; F0 F1 F2 F3 A=B Cn+4 P G
CG29ϕ1 19J4; CP29ϕ1 19J5; MAL27ϕ 16K5, 18K7; MAL26ϕ 16K5, 18K8; MAL25ϕ 16K4, 18K8; MAL24ϕ 16K4, 18K8; AEQB1 13J3; NC
7H4 MMODE1 19M4 CRY28ϕ1
12N2 BADϕ3ϕ 12N2 BADϕ2ϕ 12N1 BADϕ1ϕ 12N1 BAD$\phi\phi\phi$
8F3 BMANϕ
12J 19-075 16 x 4 REGSTK: A B C D DATA IN, S1 S2 S3 S4 DATA OUT, WE ME
12H 19-132 2 TO 1 MUX STTL (TRI-ST) ENB, 1A 1B 2A 2B 3A 3B 4A 4B, 1Y 2Y 3Y 4Y, SEL
9G7 BWNϕ
8F7 BMAS1
BST24ϕ1 19E5; BST25ϕ1 19E5; BST26ϕ1 19E5; BST27ϕ1 19E6
BMX27ϕ BMX26ϕ BMX25ϕ BMX24ϕ
12K2 AADϕ3ϕ 12K2 AADϕ2ϕ 12K1 AADϕ1ϕ 12K1 AAD$\phi\phi\phi$
10J 19-075 16 x 4 REGSTK: A B C D DATA IN, S1 S2 S3 S4 DATA OUT, WE ME
8F6 AMXS1
10H 19-132 2 TO 1 MUX STTL (TRI-ST) ENB, 1A 1B 2A 2B 3A 3B 4A 4B, 1Y 2Y 3Y 4Y, SEL
45 11K 1K 48; 44 11K 1K 41; 37 11K 1K 40; 22 11H 1K 47 → P5
9G8 AWNϕ
15A8 REGN3ϕ
8N1 AMNϕ
MIST21 5K8
15M 19-196 2 TO 1 MUX STTL (TRI-ST) ENB, 1A 2A 2B 3A 3B 4A 4B, 1Y 2Y 3Y 4Y, SEL
R2 MQ27ϕ 18R1 MQ28ϕ 5K8 DIST21 8N7 AMQCLKϕ 19H7 BSCLRϕA
15S D Q: 09 MQ24ϕ A1, 16K2, K3, 18K3; 05 NC; 12 MQ25ϕ A1, K3, 18K3; 06 NC; 13 MQ26ϕ A2, K3, 18K3; 10 NC; 11 MQ27ϕ A2, K2, 18K3; 14 NC
19-131 (STTL) SDF CLR
15K 19-230 4-BIT SHIFTER (TRI-ST): I-3 I-2 I-1 I0 I1 I2 I3 S1 S0 ENB; Y0 Y1 Y2 Y3
R2 MQ26ϕ R1 MQ25ϕ R1 MQ24ϕ 16R2 MQ23ϕ 16R2 MQ22ϕ 16R1 MQ21ϕ 16R1 MQ20ϕ
8N5 LMNBϕ 5K8 MIST2ϕ
12F 19-196 2 TO 1 MUX STTL (TRI-ST) ENB, 1A 1B 2A 2B 3A 3B 4A 4B, 1Y 2Y 3Y 4Y, SEL
18J3 MAL28ϕ 18J3 MAL29ϕ 18J2 MAL30ϕ 18J2 MAL31ϕ 8N2 ALMBS1 8N2 LMNAϕ
10F 19-196 2 TO 1 MUX STTL (TRI-ST) ENB, 1A 1B 2A 2B 3A 3B 4A 4B, 1Y 2Y 3Y 4Y, SEL
5B AL25ϕ 5B AL26ϕ 5B AL27ϕ 18S7 AL28ϕ 8N9 ALMAS1
09H 19-230 4-BIT SHIFTER (TRI-ST): I-3 I-2 I-1 I0 I1 I2 I3 S1 S0 ENB; Y0 Y1 Y2 Y3
16J2 MAL23ϕ 16J2 MAL22ϕ 16J2 MAL21ϕ 16J3 MAL20ϕ 8N3 SHSLIϕB 8N9 SHSL$\phi\phi$B 8N3 ALSRNϕ 4N6 ALCKϕ 2L4 DCLRϕB
11F D Q: 02 AL24ϕ 16K6; 03 AL25ϕ K6; 06 AL26ϕ K6; 11 AL27ϕ K6; 15; 14
19-131 (STTL) SDF CLR
AL271 AL261 AL251 AL291

"INFORMATION DISCLOSED HEREIN IS THE PROPERTY OF THE PERKIN-ELMER CORPORATION, COMPUTER SYSTEMS DIVISION, AND SHALL NOT BE DISCLOSED OR USED FOR ANY OTHER PURPOSES EXCEPT AS SPECIFIED BY CONTRACT BETWEEN THE RECIPIENT AND THE PERKIN ELMER CORPORATION. DUPLICATION OF ANY PORTION OF THIS DATA SHALL INCLUDE THIS LEGEND

SCALE— TOLERANCE .XXX ± .005 .XX ± .02 .X ± .03 ANGLES ± 1° UNLESS OTHERWISE SPECIFIED

NAME	TITLE	DATE
C.MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE: FUNCTIONAL SCHEMATICS HPFPP-A

CAGE NO. 03905 DWG NO. 35-715R02 D08 SHEET OF 17—20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS			
ON IC 16H & 16J BUBBLES WERE SHOWN ON OUTPUTS ONLY TITLE WAS DFU-A			
EJ	PE	3950	3-27-79 R01
DELETED "MODEL 3220" FROM TITLE BLOCK			
JM	5065	R	8-1-82 R02

8F3 BMBNØ
19H7 C2BØ
R1 MQ28Ø
19H8 C29Ø
R1 MQ29Ø
19H8 C3ØØ
R2 MQ3ØØ
19H8 C31Ø
R2 MQ31Ø
8F2 BMBS1
16W 19-196 2 TO 1 MUX STTL (TRI-ST)
24 25 32 31 14K 1K 14K 1K 14K 1K 14H 1K 21 28 29 45 P5
7H8 MSØØ1
7H8 MSØ11
7H7 MSØ21
7H7 MSØ31
14H 19-067 4 BIT ALU STTL
7H9 MMØDE1
19M4 CRY321
12N2 BADØ3Ø
12N2 BADØ2Ø
12N1 BADØ1Ø
12N1 BADØØØ
16J 19-075 16 x 4 REG STK ADRS DATA IN DATA OUT WE ME
8F3 BMANØ
15J 19-132 2 TO 1 MUX STTL (TRI-ST)
9G7 BWNØ
8F7 BMAS1
BST281 19E7
BST291 19E7
BST301 19E8
BST311 19E8
BMX31Ø BMX3ØØ BMX29Ø BMX28Ø
7N2 SPRDØ
12K2 AADØ3Ø
12K2 AADØ2Ø
12K1 AADØ1Ø
12K1 AADØØØ
16H 19-075 16 x 4 REG STK ADRS DATA IN DATA OUT WE ME
8F6 AMXS1
15H 19-132 2 TO 1 MUX STTL (TRI-ST)
P5 45 14K 330 48 44 14K 1K 41 37 14K 1K 40 22 14H 1K 47
9G8 AWNØ
15A8 REGNBØ
8N1 AMWØ
3A2 JAMØ
2L1 SPØ
16F 19-126 STTL
CG28I 19J4
CP28I 19J4
MAL31Ø 17K5
MAL3ØØ 17K5
MAL29Ø 17K4
MAL28Ø 17K4
AEQ8I 13J3
NC
MAL28Ø 125-1
MAL29Ø 225-1
MAL3Ø 124-1
MAL31Ø 224-1
MIST21 5K8
R2 MQ31Ø
115-1 MQ32Ø
5K8 DIST21
8N7 AMQCLKØ
19H7 BSCLRØA
16M 19-196 2 TO 1 MUX STTL (TRI-ST)
16S 19-131 (STTL) SDF
02 MQ28Ø 17K2, A1, K3
03 NC 117-1
07 MQ29Ø A1, K3
06 NC 217-1
10 MQ3ØØ A2, K3
11 NC 116-1
15 MQ31Ø A2, K3
14 NC 216-1
R2 MQ36Ø
R1 MQ29Ø
R1 MQ28Ø
17R2 MQ27Ø
17R2 MQ26Ø
17R1 MQ25Ø
17R1 MQ24Ø
16K 19-230 4-BIT SHIFTER (TRI-ST)
JMQ31Ø 202-3
JMQ3ØØ 102-3
JMQ29Ø 201-3
JMQ28Ø 101-3
8N5 LMNBØ
5K8 MIST2Ø
123-1 MAL32Ø
223-1 MAL33Ø
122-1 MAL34Ø
222-1 MAL35Ø
8N2 ALMBS1
8N2 LMNAØ
13F 19-196 2 TO 1 MUX STTL (TRI-ST)
5B AL29Ø
5B AL3ØØ
5B AL31Ø
110-1 AL32Ø
8N9 ALMAS1
13H 19-196 2 TO 1 MUX STTL (TRI-ST)
13J 19-230 4 BIT SHIFTER (TRI-ST)
17J2 MAL27Ø
17J2 MAL26Ø
17J3 MAL25Ø
17J3 MAL24Ø
8N3 SHSLIØB
8N4 SHSL6Ø
8N3 ALSRNØ
4N6 ALCKØ
2L9 DCLRØB
AL31Ø
AL3Ø1
AL291
AL281
14F 19-131 (STTL) SDF
AL28Ø 17K7
AL29Ø K6
AL3ØØ K6
AL31Ø K6
NC

SCALE—
TOLERANCE XXX ± .005, XX ± .02, X ± .03, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED

NAME	TITLE	DATE
C. MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-A

TASK NO. 03905
DWG NO. 85-715R02 D08
SHEET 18 OF 20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS			
02C-44 VALUE WAS 220 PF.			
IC 06W WAS AN OR GATE			
IC 00H PINS 05&06			
WERE NO CONNECTION			
CAP BETWEEN 02C 44			
&25 WAS 220 PF			
IC 07W12 BUBBLE WAS			
SHOWN AS OUTPUT			
TITLE WAS DFU-A			
EJ	3950	3-27-79	R01
DELETED "MODEL 3220" FROM TITLE BLOCK			
JLV	5065	8-1-82	R02

11J1 BSTØØ1 — 14, 15 — 11T 19-134 A — 13 C000 11A7 ‹136-0; 11 C010 11A8 ‹236-0; 03 C020 11A8 ‹135-0; 05 C030 11A8 ‹235-0; 07 C040 12A7 ‹134-0; 09 C050 12A8 ‹234-0
11J2 BSTØ11 — 12
11J2 BSTØ21 — 02, 01
11J2 BSTØ31 — 04
12G1 BSTØ41 — 06
12G2 BSTØ51 — 10

12G2 BSTØ61 — 14, 15 — 12T 19-134 A — 13 C060 12A8 ‹133-0; 11 C070 12A8 ‹233-0; 03 C080 13A1 ‹132-0; 05 C090 13A1 ‹232-0; 07 C100 13A2 ‹131-0; 09 C110 13A2 ‹231-0
12G2 BSTØ71 — 12
13E5 BSTØ81 — 02, 01
13E5 BSTØ91 — 04
13E6 BSTØ101 — 06
13E6 BSTØ111 — 10

14E5 BSTØ121 — 14, 15 — 13T 19-134 A — 13 C120 14A1 ‹130-0; 11 C130 14A1 ‹230-0; 03 C140 14A2 ‹129-0; 05 C150 14A2 ‹229-0; 07 C160 15A1 ‹110-0; 09 C170 15A1 ‹210-0
14E5 BSTØ131 — 12
14E6 BSTØ141 — 02, 01
14E6 BSTØ151 — 04
15E5 BSTØ161 — 06
15E5 BSTØ171 — 10

15E6 BSTØ181 — 14, 15 — 19T 19-134 A — 13 C180 15A2 ‹109-0; 11 C190 15A2 ‹209-0; 03 C200 16A1 ‹108-0; 05 C210 16A1 ‹208-0; 07 C220 16A2 ‹107-0; 09 C230 16A2 ‹207-0
15E6 BSTØ191 — 12
16E5 BSTØ201 — 02, 01
16E5 BSTØ211 — 04
16E6 BSTØ221 — 06
16E6 BSTØ231 — 10

NC, NC, NC — 14, 15, 12 — 15T 19-134 A — 13 NC; 11 NC; 03 C240 17A1 ‹106-0; 05 C250 17A1 ‹206-0; 07 C260 17A2 ‹105-0; 09 C270 17A2 ‹205-0
17E5 BSTØ241 — 02, 01
17E5 BSTØ251 — 09
17E6 BSTØ261 — 06
17E6 BSTØ271 — 10

202-0 SCLRØA — 14, 15, 12 — 16T 19-134 A — 13 BSCLRØA 11A9, 12A9, 13K2, 14K2, 15K2, 16K2, 17K2, 18K2; 03 C280 9R3, 18A1 ‹104-0; 05 C290 9R3, 18A1 ‹204-0; 07 C300 9R3, 18A2 ‹103-0; 09 C310 9R4, 18A2 ‹203-0
18E5 BSTØ281 — 02, 01
18E5 BSTØ291 — 09
18E6 BSTØ301 — 06
18E6 BSTØ311 — 10

14 00E 03 — 100Ω — 44 02C 05 — 330pF
13 00E 04 — 100Ω — 43 02C 06 — 220pF
12 00E 05 — 100Ω — 42 02C 07 — 220pF
11 00E 06 — 100Ω — 41 02C 08 — 220pF

123-4 CRY361; 222-4 CG321; 128-1 CP321; 18J2 CG281; 18J2 CP281; 17J2 CG241; 17J2 CP241; 16J2 CG201; 16J2 CP201; 15E5 — 00J 19-068 LOOK AHEAD CAR. GEN — CPX201 121-4; CRY321 18E3; CRY281 17E3; CRY241 16E3; CGX201 221-4

220-4 CRY201; 15J2 CG161; 15J2 CP161; 14J2 CG121; 14J2 CP121; 13J2 CG081; 13J2 CP081 — 00H 19-068 LOOK AHEAD CAR. GEN. — NC; CRY161 14E3; CRY121 13E3; MCOUT1 6H2, 7K3, 8H8 ‹223-4; MCOUT0 6H4, 6A5 (08F 19-057 SG)

BSCLRØA 11A9, 12A9, 13K2, 14K2, 15K2, 16K2, 17K2, 18K2 — 07W 19-057 SG — SCLR1A — 07W 19-057 SG — BSCLRØB 9H8, 2G2
00W 19-057 SG — BSCLRØC 8H6, 3N1

5K2 — MSHØ 113-1 — 06W 19-239 SN — 07W 19-057 SG

SCALE–	NAME	TITLE	DATE
TOLERANCE XXX ± .005, XX ± .02, X ± .09, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED	C. MICHAELS	DRAFT	2-9-78
		CHK	
		ENGR	

TITLE FUNCTIONAL SCHEMATICS HPFPP-A
TASK NO. 03905
DWG. NO. 35-715 R02 D08
SHEET 19 OF 20

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

CROSS REFERENCE NET#, MNEMONIC, SHEET#

NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.
0001	13B12	04	0063	AWNØ	09	0125	Ø8B12	03	0187	11MØ4	13	0249	Ø1AØ6	06	0311	AL221	16	0373	BMX11Ø	13	0435	CØ	05	0497	DIST21	05	0559	L1	05	0621	MQ21Ø	16	0683	PSØ5F		0745	RCCØ	02	0807	XALØ5Ø	12
0002	13AØ8	04	0064	BWNØ	09	0126	Ø8C12	09	0188	11MØ7	13	0250	16DØ5	12	0312	AL230	16	0374	BMX12Ø	14	0436	CØØØ	19	0498	DIST3Ø	10	0560	LDXCY1	06	0622	MQ22Ø	16	0684	PSØ5H	16	0746	Ø3F1Ø	05	0808	XALØ6Ø	12
0003	15AØ3	04	0065	ALMAS1	08	0127	Ø8D08	09	0189	11MØ9	13	0251	16DØ7	12	0313	AL231	16	0375	BMX13Ø	14	0437	CØ1Ø	19	0499	DIST4Ø	07	0561	LENBØ	08	0623	MQ23Ø	16	0685	PSØ5S	13	0747	RD1	07	0809	XALØ7Ø	12
0004	15AØ4	04	0066	ALMBS1	08	0128	Ø8E11	08	0190	11MI2	13	0252	16DØ9	12	0314	AL240	17	0376	BMX14Ø	14	0438	CØ2Ø	19	0500	DALØ7Ø	07	0562	LMNAØ	08	0624	MQ24Ø	17	0686	PSØ6A		0748	Ø3H04	15	0810	XCØ41	12
0005	15AØ7	04	0067	ALSRNØ	08	0129	Ø8F06	10	0191	12AØ3	04	0253	16D11	12	0315	AL241	17	0377	BMX15Ø	14	0439	CØ3Ø	19	0501	DBBØ	08	0563	LMNBØ	08	0625	MQ25Ø	17	0687	PSØ6B		0749	Ø3H07	15	0811	XCN1	09
0006	15A14	04	0068	AMNØ	08	0130	Ø8H11	09	0192	12AØ6	04	0254	16EØ9	10	0316	AL250	17	0378	BMX16Ø	15	0440	CØ4Ø	19	0502	DCLKØ	06	0564	LMSHØ	05	0626	MQ26Ø	17	0688	PSØ6D		0750	Ø3H09	15	0812	XCOUT1	11
0007	16AØ1	04	0069	AMXS1	08	0131	Ø8KØ3	08	0193	12AØ8	04	0255	Ø5WØ9	02	0317	AL251	17	0379	BMX17Ø	15	0441	CØ5Ø	19	0503	DCLK1	06	0565	LOAD1	08	0627	MQ27Ø	17	0689	PSØ6E		0751	REGNAØ	11	0813	XCRYØ	08
0008	16AØ7	04	0070	BMANØ	08	0132	Ø8K06	08	0194	12A11	04	0256	Ø6B06	09	0318	AL260	17	0380	BMX18Ø	15	0442	CØ6Ø	19	0504	DCLRØA	02	0566	MØ	05	0628	MQ28Ø	18	0690	PSØ6F		0752	REGNBØ	15	0814	XCRY1	08
0009	16AØ8	04	0071	BMAS1	08	0133	Ø8K11	08	0195	12B06	04	0257	16F08	18	0319	AL261	17	0381	BMX19Ø	15	0443	CØ7Ø	19	0505	DCLR0B	02	0567	MØ71	08	0629	MQ29Ø	18	0691	PSØ6W		0753	RRØ	08	0815	XELØ	11
0010	13AØ6	04	0072	BMBNØ	08	0134	Ø8M06	09	0196	12EØ6	10	0258	16H05	18	0320	AL270	17	0382	BMX20Ø	16	0444	CØ8Ø	19	0506	DCLR1	02	0568	M1	05	0630	MQ30Ø	18	0692	PSØ7A		0754	RR1	08	0816	XLØØ1	11
0011	16BØ1	04	0073	BMBS1	08	0135	Ø8T08	08	0197	12EØ8	10	0259	16H07	18	0321	AL271	17	0383	BMX21Ø	16	0445	CØ9Ø	19	0507	03E13	05	0569	MIST2Ø	05	0631	MQ31Ø	18	0693	PSØ7B		0755	RSIGNØ	09	0817	XL01Ø	11
0012	16B13	04	0074	JMQ28Ø	18	0136	Ø9A08	09	0198	JST03I	03	0260	16H09	18	0322	AL280	18	0384	BMX22Ø	16	0446	C1	05	0508	DNRLZ1	05	0570	MIST21	05	0632	MQ32Ø	18	0694	PSØ7D		0756	RSIGN1	09	0818	XL011	11
0013	WCLK1	04	0075	JMQ29Ø	18	0137	Ø9B03	09	0199	12F04	17	0261	16H11	18	0323	AL281	18	0385	BMX23Ø	16	0447	C1ØØ	19	0509	DSTØ1Ø	09	0571	MAEQBØ	06	0633	MQCEØ	08	0695	PSØ7E		0757	S3WTØ	09	0819	XL02Ø	11
0014	EACLKØ	04	0076	JMQ3ØØ	18	0138	Ø9B06	09	0200	12F07	17	0262	16RØ1	04	0324	AL290	18	0386	BMX24Ø	17	0448	C11Ø	19	0510	DST41	08	0572	MALØ8Ø	13	0634	MSØØ1	07	0696	PSØ7F		0758	S4MOFØ	08	0820	XL021	11
0015	ACLK1	04	0077	JMQ31Ø	18	0139	Ø9B08	09	0201	12F09	17	0263	16R02	04	0325	AL291	18	0387	BMX25Ø	17	0449	C12Ø	19	0511	Ø1BØ6	06	0573	MALØ9Ø	13	0635	MSØ11	07	0697	PSØ7H		0759	SBGT1	06	0821	XL03Ø	11
0016	ACLK1A	04	0078	MAL28Ø	18	0140	Ø9B11	09	0202	12F12	17	0264	16T13	19	0326	AL300	18	0388	BMX26Ø	17	0450	C13Ø	19	0512	EQUØ	08	0574	MALIØØ	13	0636	MSØ21	07	0698	PSØ7J		0760	SBMBNØ	08	0822	XL031	11
0017	ACLKØ	04	0079	MAL29Ø	18	0141	Ø9D05	11	0203	12M04	14	0265	AØ	05	0327	AL301	18	0389	BMX27Ø	17	0451	C14Ø	19	0513	ERNDØ	03	0575	MALI1Ø	13	0637	MSØ31	07	0699	PSØ7K		0761	Ø3D11	08	0823	XL040	12
0018	ALCKØ	04	0080	MAL3ØØ	18	0142	Ø9D07	11	0204	12M07	14	0266	Ø3H12	15	0328	AL310	18	0390	BMX28Ø	18	0452	CGØ81	13	0514	EXCLKØ	04	0576	MALI2Ø	14	0638	MSC01	08	0700	PSØ7M		0762	SCLRIA	19	0824	XL041	12
0019	CRY361	19	0081	MAL31Ø	18	0143	Ø9D09	11	0205	12M09	14	0267	AADØØ1	12	0329	AL320	18	0391	BMX29Ø	18	0453	CG121	14	0515	FØØ1	02	0577	MALI3Ø	14	0639	MSC11	08	0701	PSØ7N		0763	SELI	02	0825	XL050	12
0020	CRY2Ø1	19	0082	NRLZ1	05	0144	Ø9J03	09	0206	12M12	14	0268	Ø3JØ4	15	0330	Ø1E06	07	0392	BMX3ØØ	18	0454	CG161	15	0516	FØ11	02	0578	MALI4Ø	14	0640	MSHØ	19	0702	PSØ8A		0764	Ø3E04	08	0826	XL051	12
0021	CG321	19	0083	SNRLZØ	05	0145	Ø9J06	09	0207	ØØM08	03	0269	AADØ11	12	0331	Ø4S04	13	0393	BMX31Ø	18	0455	CG2Ø1	16	0517	FØ21	02	0579	MALI5Ø	14	0641	MSUMØ	07	0703	PSØ8B		0765	Ø6J07	16	0827	XL060	12
0022	CP321	19	0084	SNRLZ1	05	0146	Ø9K06	02	0208	ØØA04	06	0270	Ø3J07	15	0332	Ø4S07	13	0394	BMX321	03	0456	CG241	17	0518	FCLKØ	06	0580	MALI6Ø	15	0642	MSUM1	07	0704	PSØ8C		0766	Ø6J09	16	0828	XL061	12
0023	CPX2Ø1	19	0085	SHSLØØ	08	0147	Ø9M06	09	0209	13B06	10	0271	AADØ21	12	0333	Ø4S09	13	0395	BMX641	03	0457	CG281	18	0519	FCLK1	06	0581	MALI7Ø	15	0643	Ø6B08	07	0705	PSØ8D		0767	Ø6J12	16	0829	XL070	12
0024	CGX2Ø1	19	0086	SHSLIØA	08	0148	Ø8T06	02	0210	13B08	09	027?	Ø3J09	15	0334	AMINBØ	07	0396	SWN1	09	0458	CP081	13	0520	FIN1	03	0582	MALI8Ø	15	0644	Ø4D06	06	0706	PSØ8E		0768	SKSIGØ	05	0830	XL071	12
0025	C15Ø	19	0087	SHSLIØB	08	0149	Ø9M12	09	0211	ØØAØ1	06	0273	AADØ31	12	0335	Ø4S12	13	0397	BONLYØ	07	0459	CP121	14	0521	GNDØ1C	06	0583	MALI9Ø	15	0645	PSØØA		0707	PSØ8F		0769	SLWCLKØ	06	0831	XMDS1	09
0026	C16Ø	19	0088	XMQ3ØØ	07	0150	Ø9S11	09	0212	13E08	10	0274	ABØ6Ø	13	0336	Ø4D05	06	0398	BSCLRØA	19	0460	CP161	15	0522	GNDØ2F	05	0584	MAL20Ø	16	0646	PSØØB		0708	PSØ8H		0770	SMCOUTØ	07	0832	XMNØ	09
0027	C17Ø	19	0089	XMQ3Ø1	07	0151	Ø9S12	09	0213	13F04	18	0275	Ø1D12	07	0337	AMXS0	08	0399	BSCLRØB	19	0461	CP2Ø1	16	0523	GNDØ3F	05	0585	MAL21Ø	16	0647	PSØØC		0709	PSØ8J		0771	Ø6D06	03	0833	Ø6K04	16
0028	C18Ø	19	0090	XMQ31Ø	07	0152	Ø9W02	11	0214	13F07	18	0276	Ø1C08	06	0338	Ø4N08	06	0400	BSCLRØC	19	0462	CP241	17	0524	GNDØ3D	03	0586	MAL22Ø	16	0648	PSØØD		0710	PSØ8K		0772	Ø6J04	16	0834	Ø6K07	16
0029	C19Ø	19	0091	XMQ311	07	0153	Ø9W07	11	0215	13F09	18	0277	Ø1D06	03	0339	AONLY1	07	0401	BSTØØ1	11	0463	CP281	18	0525	GNDØ3E	07	0587	MAL23Ø	16	0649	PSØØF		0711	PSØ8S		0773	SPØ	02	0835	Ø6K09	16
0030	C2ØØ	19	0092	Ø6M05	16	0154	Ø9W1Ø	11	0216	13F12	18	0278	ACRYØ	08	0340	APLSBØ	07	0402	BSTØ11	11	0464	CRY121	19	0526	GNDØ5W	02	0588	MAL24Ø	17	0650	PSØØK		0712	PSØ8T		0774	SP1	02	0836	Ø6K12	16
0031	C21Ø	19	0093	Ø6M07	16	0155	Ø9W15	11	0217	13K11	15	0279	ACRY1	08	0341	ASØ	05	0403	BSTØ21	11	0465	CRY161	19	0527	GNDØ7C	03	0589	MAL25Ø	17	0651	PSØØM		0713	PSØ8W		0775	SPRDØ	07	0837	XMQCLKØ	08
0032	C22Ø	19	0094	Ø6M09	16	0156	1ØAØ9	06	0218	13K12	15	0280	AEQB1	13	0342	AS1	05	0404	BSTØ31	11	0466	CRY241	19	0528	GNDØ9E	10	0590	MAL26Ø	17	0652	PSØØS		0714	PSØ9A		0776	SSCØ	02	0838	XMS1	09
0033	C23Ø	19	0095	Ø6M11	16	0157	1ØA1Ø	06	0219	13K14	15	0281	AEQB1A	06	0343	ASF1	08	0405	BSTØ41	12	0467	CRY281	19	0529	GNDØ9S	09	0591	MAL27Ø	17	0653	PSØØT		0715	PSØ9B		0777	SSC1	02	0839	XMXØØØ	11
0034	C24Ø	19	0096	Ø6TØ7	13	0158	1ØEØ3	10	0220	13K15	15	0282	AEQBCØ	06	0344	ASTKØØ1	11	0406	BSTØ51	12	0468	CRY321	19	0530	GND1ØD	11	0592	Ø8T05	02	0654	PSØØW		0716	PSØ9F		0778	STØØØ	03	0840	XMXØ1Ø	11
0035	C25Ø	19	0097	Ø6TØ9	13	0159	1ØH04	17	0221	14B16	14	0283	ALØØ1	13	0345	ASTKNN1	09	0407	BSTØ61	12	0469	Ø1WØ9	14	0531	GND1ØT	09	0593	Ø5WØ7	02	0655	PSØ1A		0717	PSØ9J		0779	STØØ1	03	0841	XMXØ2Ø	11
0036	C26Ø	19	0098	Ø6T12	13	0160	1ØH07	17	0222	14K11	16	0284	ALØ9Ø	13	0346	ASUMØ	06	0408	BSTØ71	12	0470	Ø1W11	14	0532	GND11C	11	0594	Ø6A06	07	0656	PSØ1B		0718	PSØ9K		0780	STØ1Ø	03	0842	XMXØ3Ø	11
0037	C27Ø	19	0099	Ø6W01	02	0161	1ØH09	17	0223	14K12	16	0285	ALØ91	13	0347	ASUM1	06	0409	BSTØ81	13	0471	ØZBØ3	06	0533	GND12B	04	0595	Ø6A08	11	0657	PSØ1C		0719	PSØ9M		0781	STØ1ØA	03	0843	XMXØ4Ø	12
0038	C28Ø	19	0100	Ø6W04	09	0162	1ØH12	17	0224	14K14	16	0286	AL1ØØ	13	0348	Ø4E08	07	0410	BSTØ91	13	0472	ØZDØ3	08	0534	GND13E	10	0596	MAL32Ø	18	0658	PSØ1D		0720	PSØ9T		0782	STØ11	03	0844	XMXØ5Ø	12
0039	C29Ø	19	0101	RSTOPØ	02	0163	1ØJ05	17	0225	14K15	16	0287	AL1Ø1	13	0349	Ø3J12	15	0411	BST1Ø1	13	0473	GDCLKØ	08	0535	GND14A	04	0597	MAL33Ø	18	0659	PSØ1E		0721	PS1ØB		0783	STØ11A	03	0845	XMXØ6Ø	12
0040	C3ØØ	19	0102	Ø6W13	19	0164	1ØJ07	17	0226	ØØBØ6	03	0288	AL11Ø	13	0350	BADØØ1	12	0412	BST111	13	0474	ØZD11	08	0536	GND14B	04	0598	MAL34Ø	18	0660	PSØ1F		0722	PS1ØE		0784	STØ2Ø	03	0846	XMXØ7Ø	12
0041	C31Ø	19	0103	Ø7AØ6	03	0165	1ØJ09	17	0227	ØØBØ8	07	0289	AL111	13	0351	Ø3K05	15	0413	BST121	14	0475	Ø1SØ9	14	0537	GND14C	12	0599	MAL35Ø	18	0661	PSØ2B		0723	PS1ØM		0785	STØ21	03	0847	XPA1	06
0042	CLK1	02	0104	Ø7AØ8	03	0166	1ØJ11	17	0228	Ø1F02	08	0290	AL112Ø	14	0352	BADØ11	12	0414	BST131	14	0476	Ø1WØ7	14	0538	GND14E	10	0600	Ø3W12	08	0662	PSØ2C	06	0724	PS11A		0786	STØ3Ø	03	0848	XPB1	02
0043	SCLK1	02	0105	XALØ11	11	0167	1ØKØ2	09	0229	Ø1F12	08	0291	AL121	14	0353	Ø3KØ7	15	0415	BST141	14	0477	Ø3DØ3	08	0539	GND15A	04	0601	Ø4C05	06	0663	PSØ2D		0725	PS11B		0787	STØ31	03	0849	XSCØ	10
0044	SFTENØ	02	0106	Ø7B1Ø	03	0168	1ØKØ3	09	0230	15EØ2	10	0292	AL130	14	0354	BADO21	12	0416	BST151	14	0478	GDCLKØ		0540	GND16B	04	0602	MDNRLZ1	05	0664	PSØ2E	15	0726	PS11C		0788	STØ31A	03	0850	XSC1	10
0045	DFTENØ	02	0107	Ø7B13	07	0169	1ØKØ6	09	0231	15EØ4	10	0293	AL131	14	0355	Ø3K09	15	0417	BST161	15	0479	SCLRØA	19	0541	GND16E	10	0603	MFNØ	08	0665	PSØ2H	15	0727	PS11E	11	0789	STØ4Ø	03	0851	ZERØØ	06
0046	RDØ9Ø	02	0108	Ø7D02	03	0170	1ØK1Ø	09	0232	15EØ6	10	0294	AL14Ø	14	0356	BADØ31	12	0418	BST171	15	0480	ØZEØØ	05	0542	GND16T	19	0604	MMØDE1	07	0666	PSØ2S	14	0728	PS11H	17	0790	STØ41	03	0852	ZERØ1	06
0047	RD1ØØ	02	0109	Ø7D04	03	0171	1ØK15	09	0233	15EØ8	10	0295	AL141	14	0357	BGTØ	06	0419	BST181	15	0481	ØZE11	08	0543	GUFL1	10	0605	MOFFØ	06	0667			0729	PS12A		0791	STØ5Ø	03	0853	PSØ5K	13
0048	RD11Ø	02	0110	WTRNDØ	03	0172	1ØM03	09	0234	15E10	10	0296	AL15Ø	14	0358	Ø4W11	06	0420	BST191	15	0482	Ø2MØ4	14	0544	JACRYØ	08	0606	MOFF1	06	0668	PSØ3A		0730	PS12E		0792	STØ51	03	0854	PSØ2K	14
0049	RD251	08	0111	Ø7H05	08	0173	1ØM11	02	0235	15E12	10	0297	AL161	14	0359	ØSA06	10	0421	BST2Ø1	16	0483	Ø2MØ7	14	0545	JAMØ	03	0607	MQØ8Ø	13	0669	PSØ3B		0731	PS13A		0793	UFLØ	10	0855	PS11K	17
0050	AADØØØ	12	0112	LOADØ	08	0174	1ØSØ3	09	0236	15F08	08	0298	AL16Ø	15	0360	ØSBØ6	08	0422	BST211	16	0484	Ø2MØ9	14	0546	JBMXØ	08	0608	MQØ81	13	0670	PSØ3C		0732	PS13B		0794	UFL1	10	0856	PS14K	18
0051	AADØ1Ø	12	0113	Ø7J08	08	0175	1ØSØ7	09	0237	15H04	18	0299	AL161	15	0361	ØSBØ8	08	0423	BST221	16	0485	Ø2M12	14	0547	JBMXAS1	08	0609	MQØ9Ø	13	0671	PSØ3E		0733	PS13E		0795	UFLTØ	03	0857	PSØØE	19
0052	AADØ2Ø	12	0114	Ø7K08	13	0176	1ØS1Ø	09	0238	15H07	18	0300	AL17Ø	15	0362	BMXØØØ	11	0424	BST231	16	0486	Ø1S12	14	0548	ØSB12	07	0610	MQ1ØØ	13	0672	PSØ3W		0734	PS14A	04	0796	AEQB1B	06	0858	ØØEØ3	19
0053	AADØ3Ø	12	0115	Ø7M06	09	0177	1ØS14	09	0239	15H09	18	0301	AL171	15	0363	BMXØ1Ø	11	0425	BST241	17	0487	Ø1WØ5	14	0549	ØSDØ9	03	0611	MQ11Ø	13	0673	PSØ4A		0735	PS14C	12	0797	VFLGØ	10	0859	ØØE04	19
0054	BADØØØ	12	0116	Ø7M08	08	0178	1ØWØ2	12	0240	15K11	17	0302	AL18Ø	15	0364	BMXØ2Ø	11	0426	BST251	17	0488	Ø1B08	06	0550	ØSD1Ø	03	0612	MQ12Ø	14	0674	PSØ4B		0736	PS14E	12	0798	VGTO	06	0860	ØØE05	19
0055	BADØ1Ø	12	0117	Ø7T05	13	0179	1ØWØ7	12	0241	15K12	17	0303	AL181	15	0365	BMXØ3Ø	11	0427	BST261	17	0489			0551	ØSD11	03	0613	MQ13Ø	14	0675	PSØ4C		0737	PS14H	18	0799	VGT1	06	0861	ØØE06	19
0056	BADØ2Ø	12	0118	Ø7T07	13	0180	1ØW1Ø	12	0242	15K14	17	0304	AL19Ø	15	0366	BMXØ4Ø	12	0428	BST271	17	0490	Ø1S07	14	0552	JUFLØ	10	0614	MQ14Ø	14	0676	PSØ4D		0738	PS15E		0800	Ø1AØ8	06	0862	GNDØ2C	19
0057	BADØ3Ø	12	0119	Ø7T09	13	0181	1ØW15	12	0243	15K15	17	0305	AL191	15	0367	BMXØ5Ø	12	0429	BST281	18	0491	Ø3B03	06	0553	JVFLGØ	03	0615	MQ15Ø	14	0677	PSØ4E		0739	PS15F		0801	XAEQB1	06	0863	GNDØØH	19
0058	JXCRYØ	08	0120	Ø7T11	13	0182	11B04	04	0244	ØØDØ6	08	0306	AL2ØØ	16	0368	BMXØ6Ø	12	0430	BST291	18	0492	Ø3B08	06	0554	JVFLG1	10	0616	MQ16Ø	15	0678	PSØ4F		0740	PS16B		0802	XALØØØ	11	0864	Ø9K08	06
0059	MCOUTØ	19	0121	Ø7W04	19	0183	11B08	11	0245	ØØDØ8	08	0307	AL2Ø1	16	0369	BMXØ7Ø	12	0431	BST3Ø1	18	0493	Ø3B11	06	0555	JVGTØ	06	0617	MQ17Ø	15	0679	PSØ4W		0741	PS16F		0803	XALØ1Ø	11	0865	REG DISØ	11
0060	MCOUT1	19	0122	Ø7W0Ø	02	0184	11EØ1	10	0246	ØØF08	08	0308	AL21Ø	16	0370	BMXØ8Ø	13	0432	BST311	18	0494	Ø1SØ4	14	0556	JVGT1	06	0618	MQ18Ø	15	0680	PSØ5A		0742	PLUS1	09	0804	XALØ2Ø	11	0866	16B10	10
0061	OPSTOPØ	02	0123	Ø8BØ6	03	0185	11EØ4	10	0247	ØØT12	02	0309	AL211	16	0371	BMXØ9Ø	13	0433	BSTKWN1	09	0495	DØ	05	0557	Ø3K11	15	0619	MQ19Ø	15	0681	PSØ5B		0743	PSW191	03	0805	XALØ3Ø	11	0867	FPPFLTØ	09
0062	AMQCLKØ	08	0124	Ø8BØ8	03	0186	11EIØ	09	0248	16B04	04	0310	AL22Ø	16	0372	BMX1ØØ	13	0434	Ø4FØ2	07	0496	D1	05	0558	LØ	05	0620	MQ2ØØ	16	0682	PSØ5E	16	0744	RØ	05	0806	XALØ4Ø	12	0868	RCC1	09

REVISIONS

NET 0473 WAS Ø2DØ8
ADDED NETS 0863,0864, 0865
NET 0488 WAS Ø2TØ7 ON SHEET 11 IS OMITTED
NET 0489 WAS Ø2TØ8 ON SHEET 15 IS OMITTED
NET 0667 WAS PSØ2T AND 15 OMITTED
TITLE WAS HOURGLASS DFU-A

EJ	RC	3950		3-27-79	R01

NET 866 WAS NOT SPEC.

RLS	J.P.	4020		5-4-79	R02

NET 105 WAS ØTBØ4-08
NET 198 WAS 12E12-03
NET 281 WAS AEQBIC-06
NET 796 WAS UFLT1-03
ADDED NET 0488

KR	AJ	4096	M	7-17-79	R03

ADDED NETS 0867,0868

JAH	44491	MS	10-7-80	R04	

DELETED "MODEL 3220" FROM TITLE BLOCK.

JW	5065	R	8-1-82	R05	X

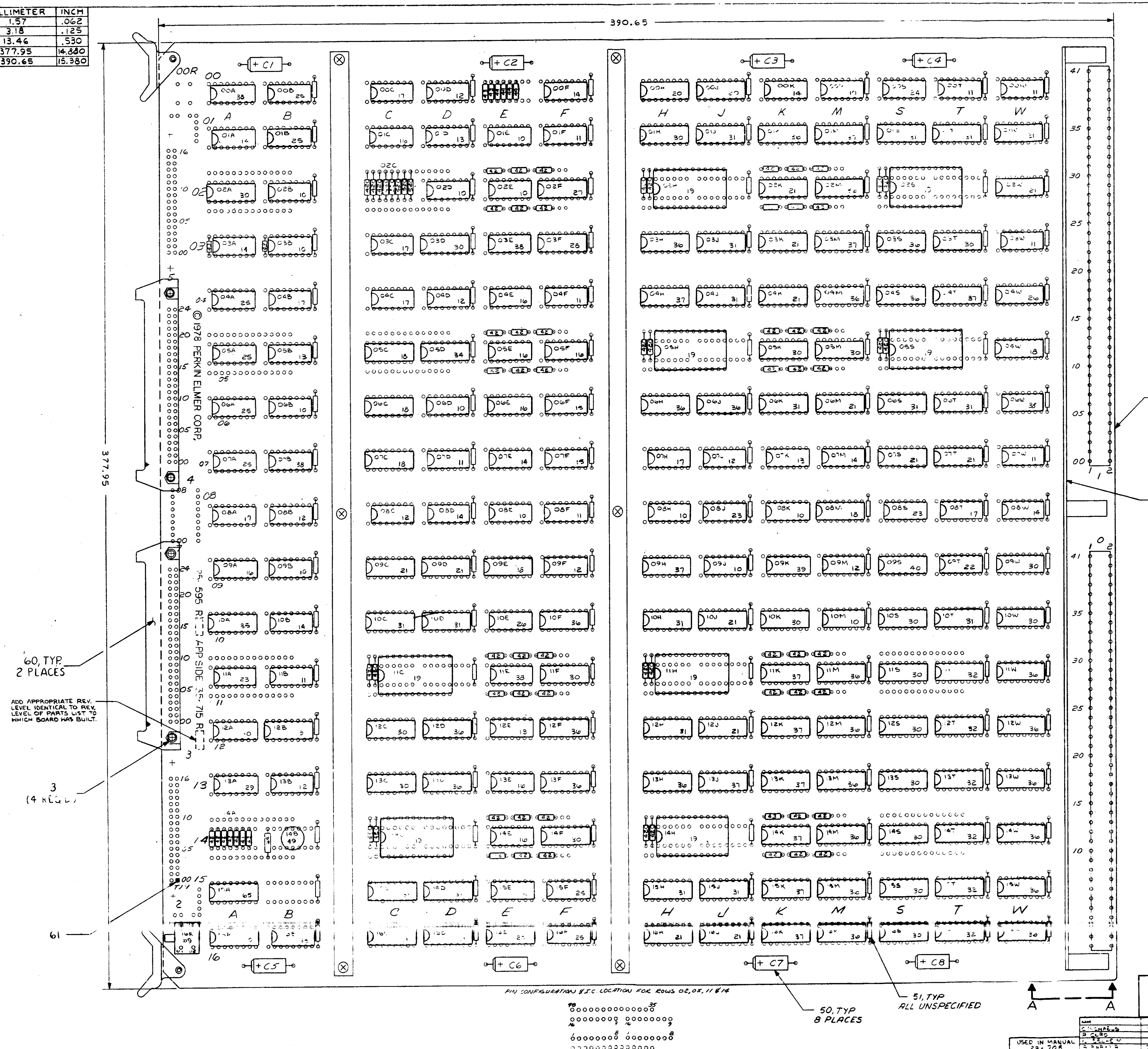
SCALE –
TOLERANCE
XXX ± .005
.XX ± .02
.X ± .03
ANGLES ± 1°
UNLESS OTHERWISE SPECIFIED

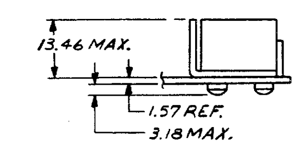
NAME	TITLE	DATE
C. MICHAELS	DRAFT	12-5-78
	CHK	
	ENGR	

TITLE: FUNCTIONAL SCHEMATICS HPFPP-A

TASK NO. 03908
DWG NO. 55-715R05 D08
SHEET OF 20 – 20

MILLIMETER	INCH
1.57	.062
3.18	.125
13.46	.530
377.95	14.880
390.65	15.380





PARTIAL VIEW A-A
TYP 3 PLACES

NOTES:
1. FOR MOUNTING OF STANDARD HARDWARE SEE 16-642 D12.
2. DIMENSIONS ARE IN MILLIMETERS.
3. BEND PINS CLOSEST TO EDGE OF BOARD INWARD PRIOR TO SOLDERING.

COMPONENT	REF. DESIGNATION

REVISIONS

PRE PRODUCTION APPROVAL	DEV / PROD	INIT / DATE

TITLE WAS DFU-A.
ITEM 64 AT 14B WAS CONN BETWEEN 14B [illegible] PIN 16.
ITEM 53 IN POS 02C-05 WAS ... N ITEM 52.
TP1 CONN 2 WAS SHOWN IN ROW 2.
REMOVED 2 RESISTOR AT 02T-07 & 08 (ITEM 42)

	ECR 3950		3-19-79	R01

RELEASED FOR PRODUCTION
MFG ENG ____ DATE ____

IC 03A WAS ITEM 12
ADDED ITEM 43 TO 03A & 03B

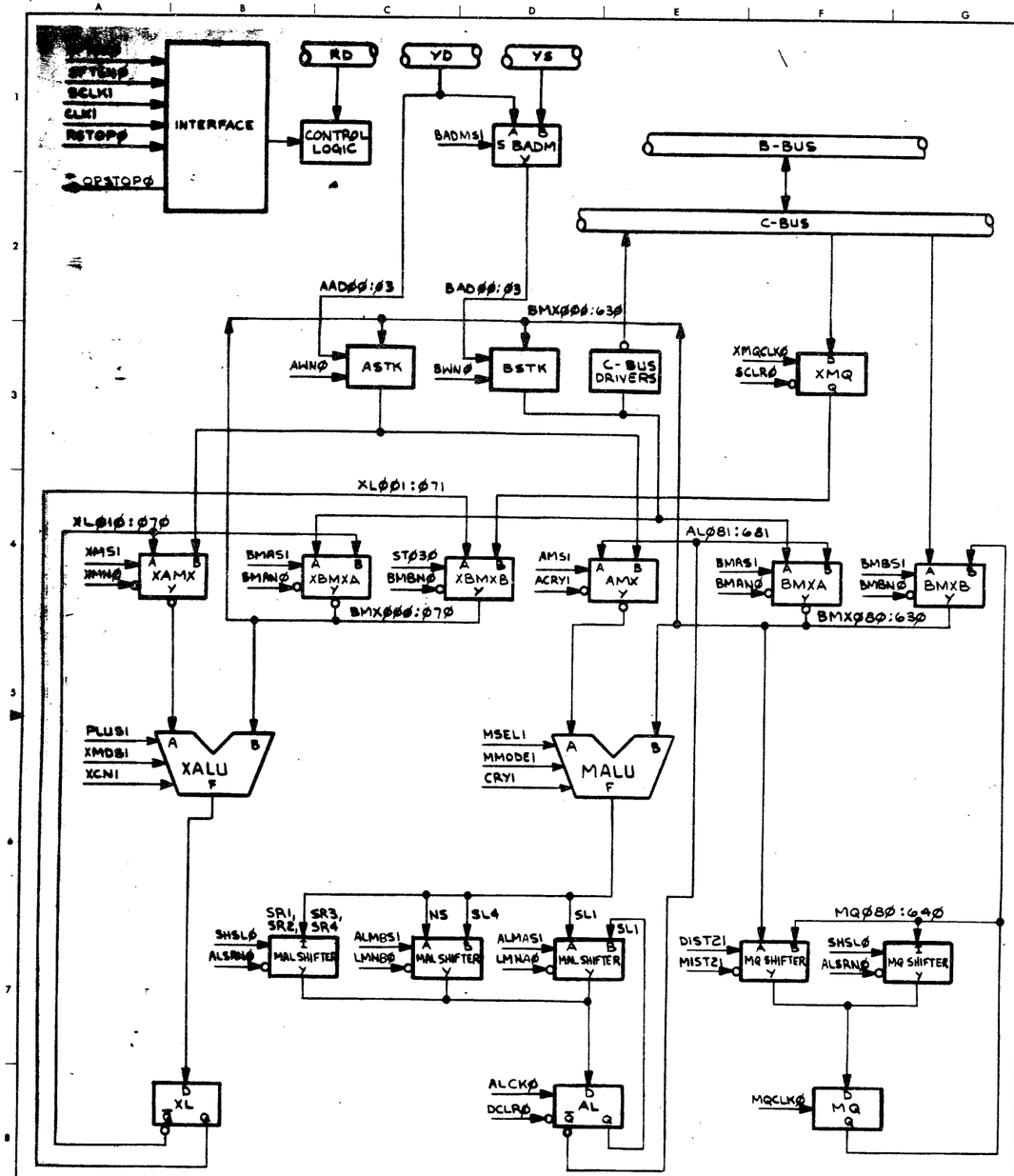
ECR		4096	M	7-13-79	R02

PERKIN ELMER
Computer Systems Division
Oceanport, N J 07757

USED IN MANUAL 29-705

NAME	TITLE	DATE
C. MICHAELS	DRAFT	9-22-78
P. CERO	CHK	5-11-79
[illegible]	ENGR	5-11-79
[illegible]		5-11-79
D. B[illegible]NN	DIR ENG	5-11-79

TITLE MULTIWIRE ASSEMBLY MULTI[illegible] HPFPP-A
03905
35-715 R02 E03
SHEET 1 OF 1



NOTE:
1. FOR NET# MNEMONIC & LOCATION ON SCHEMATIC, SEE SHT. 16.

REF. DESIGNATION	PART NUMBER	SPARE OUTPUT NUMBER
00S	19-055	03,06
07K	19-055	03,06
08W	19-055	03,11
00E	19-057	02,04,08,12
01S	19-057	06,12
04E	19-057	04,06
12T	19-057	06,12
16F	19-057	06,08,10,12
06K	19-058	08
08C	19-236	03,08,11
04F	19-059	08
01E	19-063	08/09
08C	19-064	07/09
08T	19-124	08
08D	19-125	03,06
01F	19-126	08
10E	19-126	08
03	19-37	08

TABLE OF SPARES

REVISION	7	4	3	3	3	3	2	2	2	2	2	2	2	3	2	5
SHEET	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

BRUNING 44-231 24838

USED IN MANUAL 29-705

CONN	TERM. NO.	CABLE CONNECTOR MAP ROW 2	ROW 1	BACK PANEL MAP ROW 1	ROW 2	TERM. NO.	CONN
5	16			P5	GND	41	
	15			GND	GND	40	
	14			AAD00I	AAD01I	39	
	13			AAD02I	AAD03I	38	
	12			BAD00I	BAD01I	37	
	11			BAD02I	BAD03I	36	
	10					35	
	09					34	
	08					33	
	07					32	
	06					31	
	05					30	
	04					29	
	03					28	
	02			CP32I		27	
	01			RDI	NRL2I	26	
	00					25	
				MAL28Ø	MAL29Ø	24	
				MAL30Ø	MAL31Ø	23	
				MAL32Ø	MAL33Ø	22	
				MAL34Ø	MAL35Ø	21	
4	24	GND	GND	GND	GND	20	1
	23	MCOUTI	CRY3I	YS0I	YS1I	19	
	22	CG32I		YS2I		18	
	21	CGX2GI	CPX2GI			17	
	20	CRV2GI	MOFFI			16	
	19	FCLKØ	DCLKØ	MQ28Ø	MQ29Ø	15	
	18	JACRYØ	JXCRYØ	MQ30Ø	MQ31Ø	14	
	17	JSTØ	BGTØ	MQ32Ø	MACØ	13	
	16	DSSØ	SBMBNØ	ALGNØ	FNCEØ	12	
	15	GND	AMXSØ	MSHØ	EQUØ	11	
	14	AEQBIA	AEQBIB	REGDISØ	RD2SI	10	
	13				DISTØ	09	
	12		SNRL2Ø	AL32Ø	EVX32I	08	
	11	JBMXASI	JBMXØ	MFNØ		07	
	10	GND	JVGTI	YD28I	YD29I	06	
	09	AMINBØ	APLSBØ	YDIØI	YDIII	05	
	08	BONLYØ	AONLYI	SFTENØ	DFTENØ	04	
	07	XMQ3 1	XMQ31Ø	RD09Ø	RD10Ø	03	
	06	XMQ36I	XMQ3ØØ	RDIIØ	UNNLDØ	02	
	05	BSTKWNI	ASTKWNI	LENBØ	DCURI	01	
	04	GND		SKSIGØ	MQCEØ	00	
	03	ST05Ø	ST04Ø	GND	GND		
	02	ST03Ø	ST02Ø	P5	GND		
	01	ST01Ø	ST00I				
	00	GND	GND				
				P5	GND	41	
				GND	GND	40	
						39	
						38	
				GND	GND	37	
3	24	GND	GND	C000	C010	36	
	23	MDTI	BMX94I	C020	C030	35	
	22		JAMØ	C040	C050	34	
	21	MSUMI	GND	C060	C070	33	
	20			C080	C090	32	
	19			C100	C110	31	
	18			C120	C130	30	
	17	SWNI		C140	C150	29	
	16			GND	GND	28	
	15					27	
	14					26	
	13					25	
	12	VSCII	MSCØI			24	
	11					23	
	10					22	
	09	XALØ1I				21	0
	08					20	
	07	ACLKI	EACLKØ			19	
	06		GND			18	
	05	XALØ7Ø	XALØ6Ø			17	
	04	XALØ5Ø	XALØ4Ø			16	
	03	XALØ3Ø	XALØ2Ø			15	
	02	JMQ31Ø	JVQ3ØØ			14	
	01	JMQ29Ø	JMQ28Ø			13	
	00	GND	GND			12	
				GND	GND	11	
	16			CI60	CI70	10	
	15			CI80	CI90	09	
	14			C200	C210	08	
	13			C220	C230	07	
	12			C240	C250	06	
	11			C260	C270	05	
	10			C280	C290	04	
	09			C300	C310	03	
	08			CBENØ	SCLRØA	02	
2	07			GND	GND	01	
	05			P5	GND	00	
	04						
	03						
	02						
	01						
	00						

THE REVISION LEVEL OF THIS SHEET IS CONSIDERED TO BE THE REVISION LEVEL OF THE DOCUMENT.

PERKIN ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS

PRE PRODUCTION APPROVAL	DEV	INIT.	DATE
			1/5/79

TABLE OF SPARES, GATE 08S, 19-059, PIN 08 WAS SHOWN.
SIGNALS ALGNØ, REGDISØ, MACØ, & FNCEØ WERE NOT SHOWN ON BACK PANEL MAP.
TITLES WERE DFU-B ON SHT 1 THRU 16.
IN AREA 58 & 88 REV. LEVEL OF 35-716 WAS R01.
SHEETS 2 THRU 16 WERE REV. LEVEL R00.

DS	JP	3951	4-5-79	R01

RELEASED FOR PRODUCTION
ENG. DATE 3/19/79

TABLE OF SPARES 08C 19-236 03,08,11 & 04F 19-059 08 WERE NOT SPEC, 06K 19-058 08 WAS 01K 19-058 06,08.
CABLE CONNECTOR MAP CONN 4 PIN 14 AEQBIB WAS ADDED. AEQBIA WAS AEQBIC. CONN 3 PIN 09 WAS NOT SPEC. XALUE MALU WERE SHOWN INCORRECT. REVISED SHT'S 2, 4, 5, 6, 14 & 16 AREA 59 BD REV LEVEL WAS R02

KR	JV	4096	M	7/16/79 R02

REVISED SHTS 2, 3, 11 & AREA 58, 35-716 WAS R03.

BP	JV	4146	M	9-20-79 R03

REVISED SHT 16, AREA 58 35-716 WAS R04

JV	JV	4162	M	11-2-79 R04

AREA 58 35-716 WAS R05

KR	JV	4181	R	11-7-79 R05

REVISED SHT 1, AREA C7 & D7, ALMBSI & ALMASI, AND LMNBØ & LMNAØ WERE REVERSED

VT	JV	L19009-15	R	11-17-80 R06

DELETED "MODEL 3220" FROM TITLE BLOCK REVISED SHTS 1-16

JW	JV	5065	R	8-1-82 R07

35-716	R06
BOARD	REV. LEVEL

BOARDS AGREEING WITH THIS SCHEMATIC WILL BE AT LEAST THE FOLLOWING REVISION LEV...

C. MICHAELS	DRAFT	10-3-78	
N. LEMONDE	MGR	5-11-79	
R. CERO	"	5-11-79	
N. SELHEIM	ENGR	5-11-79	
R.A. BARKER	Q.C.	5-11-79	
E. GREENSTEIN	SYSTEST	5-11-79	

SCALE —

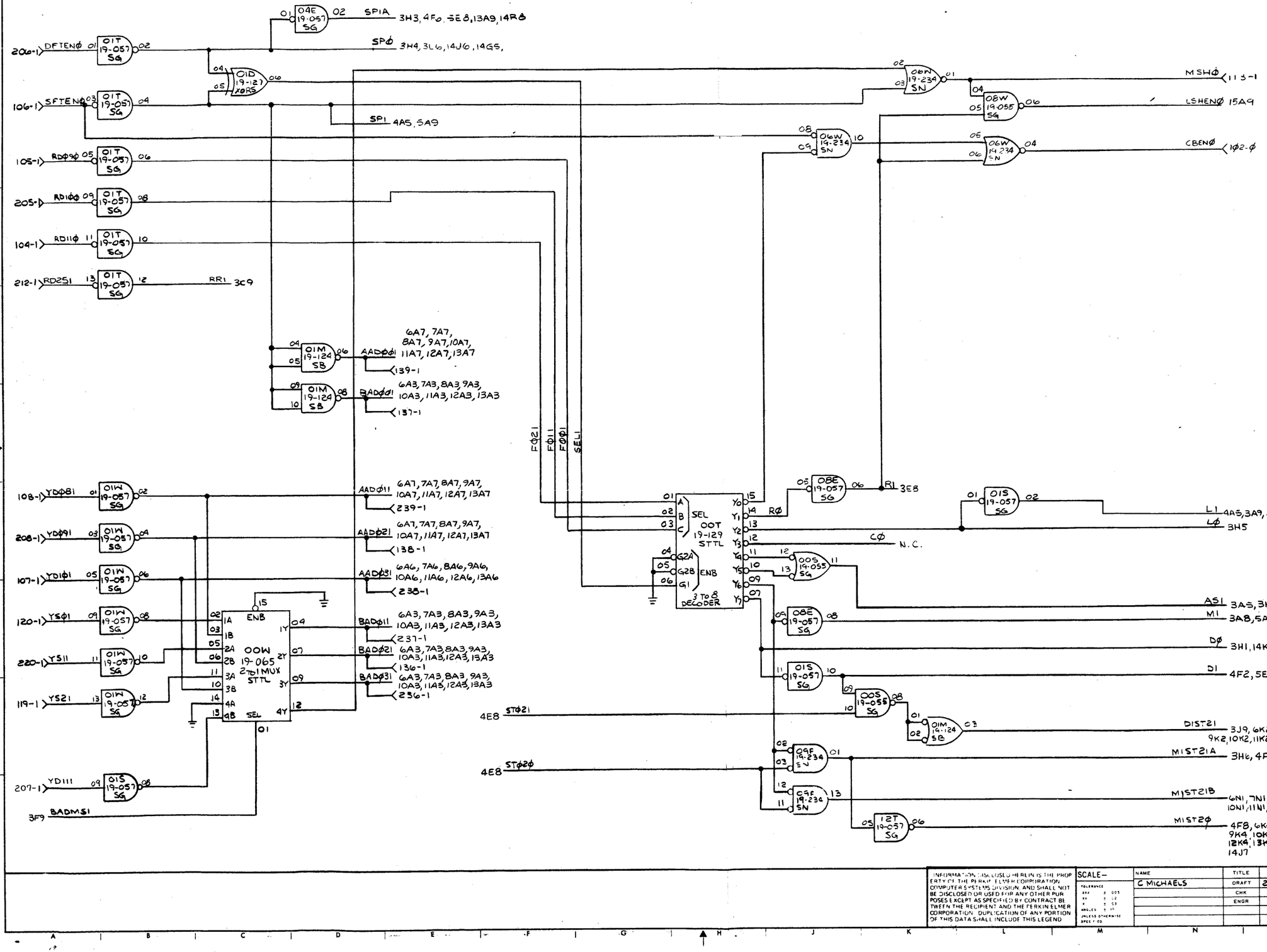
TITLE FUNCTIONAL SCHEMATICS HPFPP-B MULTIWIRE

03905		
35-716	R07	D08

SHEET 1 OF 16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS				
THE FOLLOWING WERE SHOWN AS AND GATES: 00511,01M03. THE FOLLOWING WERE SHOWN AS OR GATES: 06F10, 09F01 09F13. ON IC'S 04E02, 01T04, 04, 08 10, 12, 01502, 08E06, 08E08 01510 BUBBLES WERE ON OUTPUT, TITLE WAS DFU-B				
BS	HP	3951		3-30-79	R01
AREA R7 ADDED 5J9 TO ASI				
KR	AV	4096	M	7-26-79	R02
AREA N6, ADDED CROSS REF 3A4 TO GATE 01502.				
BP	[illegible]	4146	M	9-20-79	R03
DELETED "MODEL 3220" FROM TITLE BLOCK				
JW		5065	R	8-1-82	R04



SCALE—

TOLERANCE
XXX ± .005
XX ± .02
X ± .03
ANGLES ± 1°
UNLESS OTHERWISE SPECIFIED

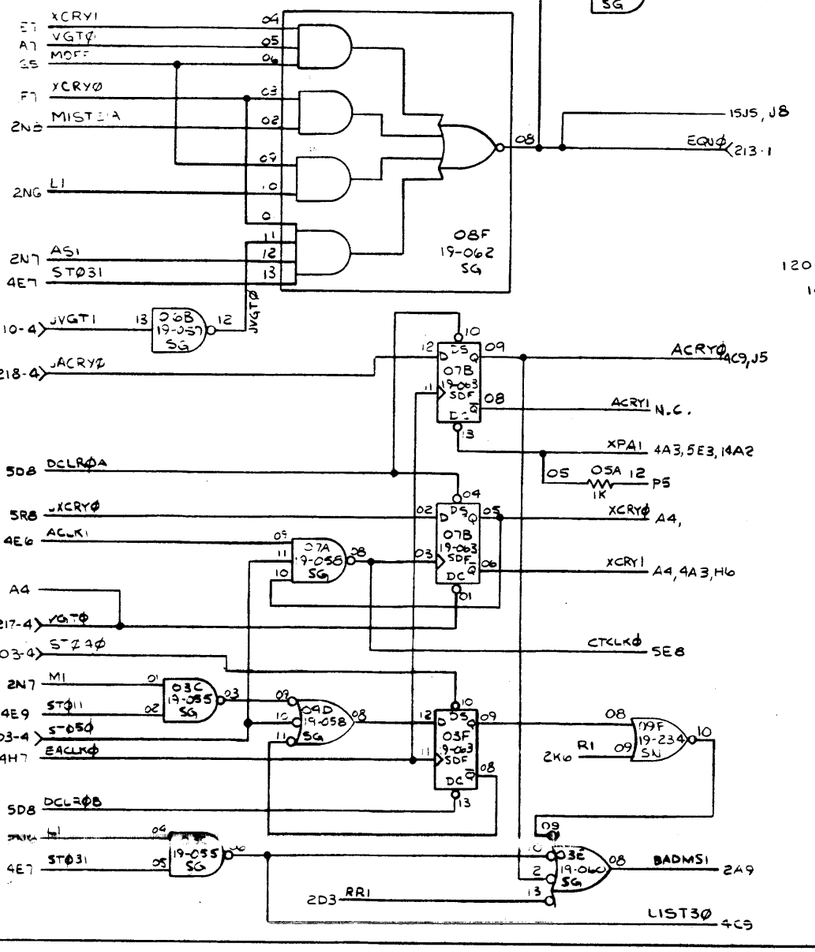
NAME	TITLE	DATE
C MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

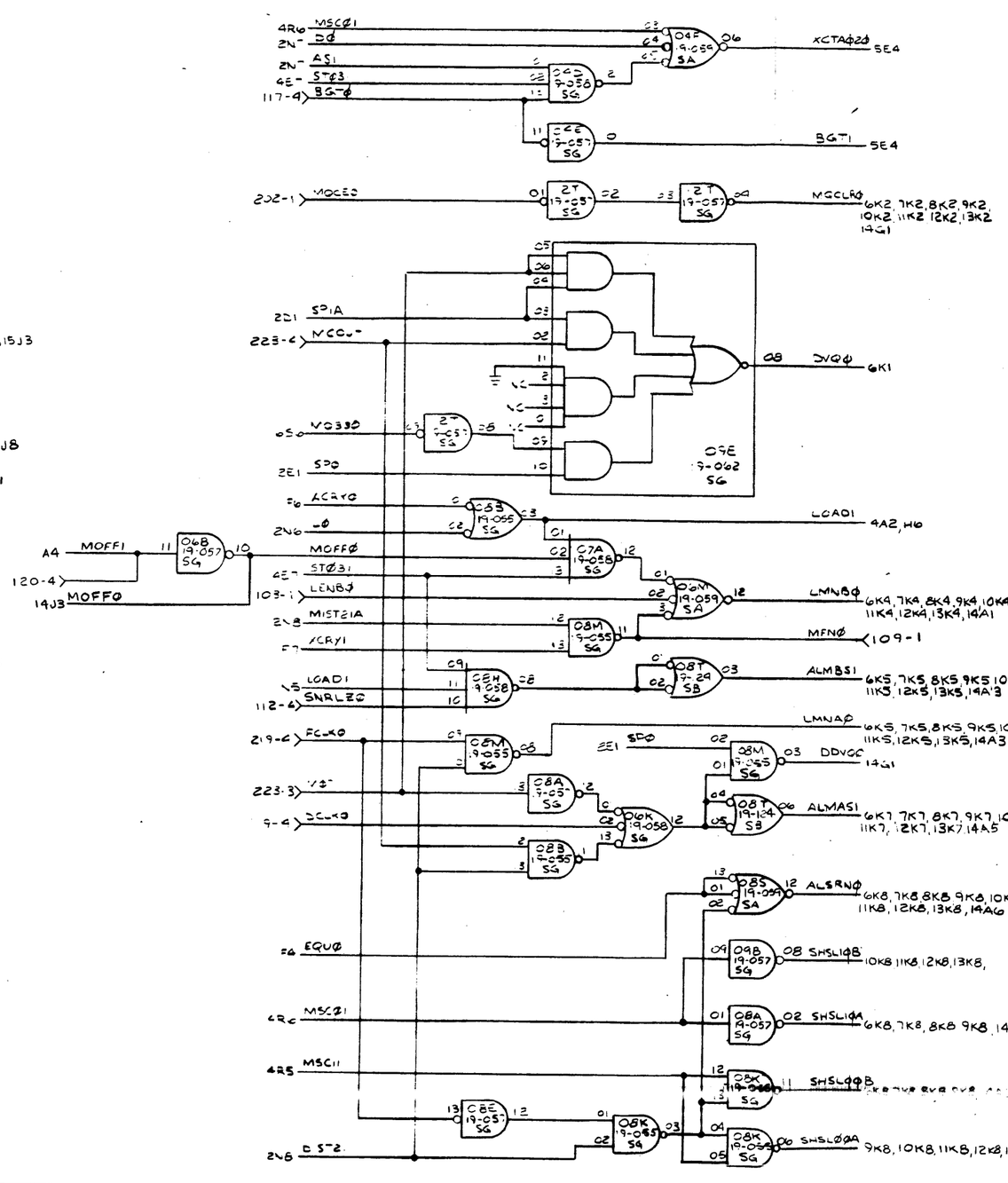
TITLE
FUNCTIONAL SCHEMATICS
HPFPP-B

TASK NO. 03905
DWG NO. 35-716R04 D08
SHEET OF 2—16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS				
THE FOLLOWING WERE SHOWN AS AND GATES 04D08, 03E08, 04F06, 08B03 06M12, 08T03, 06K12, 08T06 & 08S12. ON IC'S 07B, 03F DS, Q, & DC WERE S, 1, 0 R RESPECTIVLY ON IC'S 04E08, 10, 12T02, 08 & 08E12 BUBBLE WERE ON OUTPUTS TITLE WAS DFU-B				
DS	JJP	3951		3-30-79 R01
AREA A4, 08F09 WAS N.C. 08F10 WAS CONNECTED TO GND.				
BP		4146	M	9-20-79 R02
DELETED "MODEL 3220" FROM TITLE BLOCK				
JLV		5065	R	8-1-82 R03





SCALE –	NAME C. MICHAELS	TITLE DRAFT	DATE 2-11-78	TITLE FUNCTIONAL SCHEMATICS HPFPP-B
		CHK		
		ENGR		
				DWG NO 03905 SHEET OF
				CMR NO 35-716R03 D08 3 — 16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS

THE FOLLOW WERE SHOWN AS AND GATES; 06M06, 08 06A06, 08T11, 01M11, 07K08 & 09A08.
THE FOLLOWING WAS SHOWN AS AN OR GATE; 14B01 ON IC 07M DS, Q, Q & DC WERE S, 1, 0 & R - RESPECTIVLY. ON 16B - BUBBLES WERE ON OUTPUT ONLY.
ON GATES 12T10, 09B02 08A04, 06 & 08 BUBBLES WERE ON OUTPUTS.
TITLE WAS DFU-B

DS	3951		4-2-79	ROI
AREA F9 ADDED 5J9 TO ST#11				
KR	4096	M	7:26:79	R02
DELETED "MODEL 3220" FROM TITLE BLOCK				
JLV	5065	R	8-1-82	R05

A6 ST#Ø1 — 08E 19-057 SG — 06M 19-059 SA — BMBNØ → 6A1, 7A1, 8A1, 9A1, 10A1, 11A1, 12A1, 13A1, 14J2
111-4 JBMXØ — 07M 19-058 SDF DCR — BMANØ → 6A5, 7A5, 8A5, 9A5, 10A5, 11A5, 12A5, 13A5, 14A8
116-4 SBMBNØ
216-4 DBBØ — 06M 19-059 SA
3N5 LOAD1 — 08A 19-057 SG — 06A 19-059 SG — JBMXAS1 → 211-4
3A4 XCRY1 — 03C 19-055 SG
E8 ST#21
3F6 XPA1
07M 19-063 SDF DCR — NC
08T 19-124 SB — BMAS1 → 6D3, 7D3, 8D3, 9D3, 10D3, 11D3, 12D3, 13D3, 14A9
·5D8 DCLRØA
211-1 DIST4Ø — 01M 19-124 SB — AMXS1 → 6D7, 7D7, 8D7, 9D7, 10D7, 11D7, 12D7, 13D7, 14A8
113-4 AMXSØ
SKS16Ø → 102-1
08W 19-055 SG
204-1 WNNLDØ — 12T 19-057 SG
07K 19-055 SG
217-3 SWN1 — 07K 19-055 SG
2N6 L1 — 08J 19-060 SG — BWNØ → 6A5, 7A5, 8A5, 9A5, 10A5, 11A5, 12A5, 13A5
2E2 SP1
205-4 BSTKWN1
207-3 NCLK1
105-4 ASTKWN1 — 08J 19-060 SG — AWNØ → 6A8, 7A8, 8A8, 9A8, 10A8, 11A8, 12A8, 13A8
101-4 ST#Ø1 — A1
09E 19-234 SN — BMQCLKØ → 6K2, 7K2, 8K2, 9K2, 10K2, 11K2, 12K2, 13K2, 14G2
08K 19-055 SG — ALCKØ → 6J8, 7J8, 8J8, 9J8, 10J8, 11J8, 12J8, 13J8, 14A6
ACLK1 → 3A7, 14H5
09B 19-057 SG
107-3 EACLKØ
EACLKØ → 3A8
08B 19-055 SG — ALCKØB → 14L9, 15M4
08A 19-057 SG
202-4 ST#3Ø — ST#31 → G2, 3A5, 3A9, 3H2, 3H5
ST#3Ø → 15J2
08A 19-057 SG
102-4 ST#2Ø — ST#E1 → A3, 2F8,
ST#2Ø → 2F8
08A 19-057 SG
201-4 ST#1Ø — ST#11 → 3A8, 5A4, 5J9
ST#1Ø → 5E1
3F9 LIST3Ø — 08M 19-056 SG — AMNØ → 6D8 7D8, 8D8, 9D8, 10D8, 11D8, 12D8, 13D8, 14A7, 14K3
3F6 ACRYØ

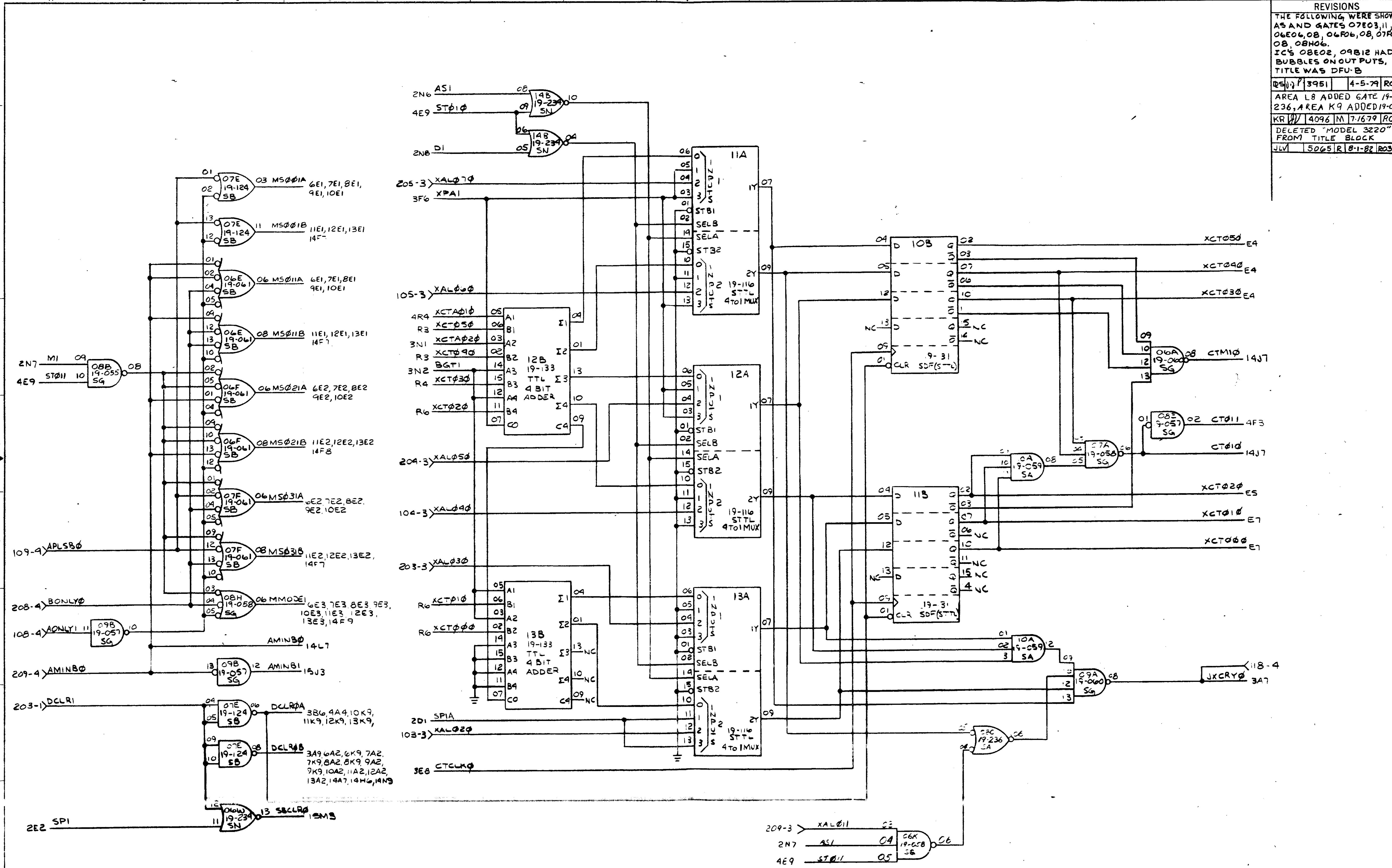
2N8 DI — 14B 19-234 SN
E7 ST#31
2N8 MISTE1A
5R5 ST#11
221-3 MSUM1
15B 19-062 SG
09A 19-060 SG — 09B 19-057 SG — XCTAØ → 5E4
15A 19-060 SG
10A 19-059 SA — MSC11 → 212-3, 3H9, 15J4
15A 19-060 SG
13N1 JMQ6ØØ — 16B 19-066 STTL 2 to 1 MUX — 16A
101-3 JMQ28Ø
13N1 JMQ61Ø
201-3 JMQ29Ø
13N2 JMQ62Ø
102-3 JMQ3ØØ
13N2 JMQ63Ø
202-3 JMQ31Ø
2D1 SP1A
15H7 BSCLRØ4
19-131 SDF(STTL) CLR
14A 19-062 SG
14B 19-234 SN — 09B 19-057 SG — MSC01 → 112-3, 3H1, 3H8, 15J5
XMQ3Ø1 → 206-4
XMQ3ØØ → 106-4
XMQ311 → 207-4
XMQ31Ø → 107-4
2N9 MISTEØ

SCALE –
TOLERANCE: XXX ± .005; XX ± .02; X ± .03; ANGLES ± 1°; UNLESS OTHERWISE SPECIFIED

NAME	TITLE	DATE
C MICHAELS	DRAFT	2-11-78
	CHK	
	ENGR	

TITLE: FUNCTIONAL SCHEMATICS HPFPP-B
TASK NO. 03905
DWG NO. 35-716R03 D08
SHEET OF 4 – 16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757



REVISIONS				
THE FOLLOWING WERE SHOWN AS AND GATES 07E03, 11, 06E06, 08, 06F06, 08, 07F06, 08, 08H06. IC'S 08E02, 09B12 HAD BUBBLES ON OUT PUTS, TITLE WAS DFU-B				
	3951		4-5-79	R01
AREA L8 ADDED GATE 19-236, AREA K9 ADDED 19-058				
KR	4096	M	7-16-79	R02
DELETED "MODEL 3220" FROM TITLE BLOCK				
JLV	5065	R	8-1-82	R03

INFORMATION DISCLOSED HEREIN IS THE PROPERTY OF THE PERKIN-ELMER CORPORATION, COMPUTER SYSTEMS DIVISION, AND SHALL NOT BE DISCLOSED OR USED FOR ANY OTHER PURPOSES EXCEPT AS SPECIFIED BY CONTRACT BETWEEN THE RECIPIENT AND THE PERKIN-ELMER CORPORATION. DUPLICATION OF ANY PORTION OF THIS DATA SHALL INCLUDE THIS LEGEND.

SCALE-	NAME		DATE	TITLE
	C. MICHAELS	DRAFT	2-13-78	FUNCTIONAL SCHEMATICS HPFPP-B
		CHK		
		ENGR		

03905
35-716R03 D08 — SHEET 5 OF 16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS
IN AREA A9 RES, 05A, 1K WAS DELETED, 05A 10 WENT TO P5, 05A WENT TO 00E05, IC 08508 AND SIGNALS REGDIS∅, FNC∅, ALGN∅ & MAC∅ WERE NOT SPECIFIED. ON IC'S 07S, 07T, 01S04 10F08, 06, 04, 02 BUBBLES WERE ON OUTPUTS ONLY. TITLE WAS DFU-B
∅5 ∅1 P 3951 4-5-79 R01
AREA J3 AE∅B1A 219-4 WAS AE∅B1A 19J8
KR /// 4096 M 7/16/79 R02
DELETED "MODEL 3220" FROM TITLE BLOCK
JLV 5065 E 8-1-82 R03

MULTIPLICATION SHIFT CODE		
SHSLI∅	SHSL∅∅	
0	0	SR4
0	1	SR3
1	0	SR2
1	1	SR1

SCALE –
TOLERANCE: XXX ± .005; XX ± .02; X ± .03; ANGLES ± 1°; UNLESS OTHERWISE SPECIFIED

NAME	TITLE	DATE
C. MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-B

TASK NO. 03905
DWG NO. 35-716R03 D08
SHEET OF 6 — 16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS					
ON IC'S 04W, 05W, M704, 02 10F12, 16 BUBBLE WERE ON OUTPUT TITLE WAS DFU-B					
35	01	3951		4-5-79	R01
DELETED "MODEL 3220" FROM TITLE BOX					
JLV		5065	R	8-1-82	R02

"INFORMATION DISCLOSED HEREIN IS THE PROPERTY OF THE PERKIN ELMER CORPORATION: COMPUTER SYSTEMS DIVISION, AND SHALL NOT BE DISCLOSED OR USED FOR ANY OTHER PURPOSES EXCEPT AS SPECIFIED BY CONTRACT BETWEEN THE RECIPIENT AND THE PERKIN ELMER CORPORATION. DUPLICATION OF ANY PORTION OF THIS DATA SHALL INCLUDE THIS LEGEND

SCALE—

NAME	TITLE	DATE
C.MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-B

TAPE NO.	03905
DWG NO.	35-716 R02 D08
SHEET	7 OF 16

REVISIONS				
ON IC'S 00K, 01K, & 14T12, 10, 08, 06 BUBBLES WERE ON OUTPUTS, TITLE WAS DFU-B				
DS	N.P.	3951	4-5-79	R01
DELETED "MODEL 3220" FROM TITLE BLOCK				
JV	5065	R	8-1-82	R02

MIST21B 2N9

4E1 BMBW0 — ENB 15 — 11W 19-196 STTL 2TO1MUX (TRI-ST) — SEL 01
15D5 CØBØ 02 — MQ4ØØ 03 — CØ9Ø 05 — MQ41Ø 06 — 15D5 C1ØØ 11 — MQ42Ø 10 — 15D6 C11Ø 14 — MQ43Ø 13
1A 1B 2A 2B 3A 3B 4A 4B — 1Y 04 — 2Y 07 — 3Y 09 — 4Y 12
5D8 DCLRØB

21 25 29 21 — 02E 1K — 02H 1K — 24 28 32 48 — P5

5D3 MSØØ1A — 5D4 MSØ11A — 5D5 MSØ21A — 5D6 MSØ31A
SUM INPUTS — 02H 19-067 4 BIT ALU STTL — FUNCTION INPUTS
A0 A1 A2 A3 B0 B1 B2 B3 M Cn — G P F0 F1 F2 F3 A=B Cn+4
06 05 04 03 02 01 23 22 21 20 19 18 08 07 — 17 15 09 10 11 13 14 16
CG9Ø1 15J9 — CP4Ø1 15J9 — MAL43Ø 7K5, 9K7 — MAL42Ø 7K5, 9K8 — MAL41Ø 7K5, 9K8 — MAL4ØØ 7K4, 9K8 — AEQBIB 14J9 — NC

5D7 MMODE1 — 15N8 CRY4Ø1

4E3 BMAS1 — 01 — SEL — 01H 19-132 2TO1MUX STTL (TRI-ST) — ENB 15
4B 4A 3B 3A 2B 2A 1B 1A — 4Y 3Y 2Y 1Y — 12 09 07 04
BST43I 15A6 — BST42I 15A6 — BST41I 15A5 — BST4ØI 15A5

2E8 BADØ31 — 2E7 BADØ21 — 2E7 BADØ11 — 2E5 BADØØ1
00K 19-075 16x4 REGSTK — ADRS — DATA IN — DATA OUT — A B C D — D1 D2 D3 D4 — S1 S2 S3 S4 — 01 15 14 13 — 05 07 09 11 — WE ME — 03 02

4E5 BWNØ — 4E1 BMANØ

BMX43Ø BMX42Ø BMX41Ø BMX4ØØ

2E6 AADØ31 — 2E6 AADØ21 — 2E6 AADØ11 — 2E4 AADØØ1
01K 19-075 16x4 REGSTK — ADRS — DATA IN — DATA OUT — A B C D — D1 D2 D3 D4 — S1 S2 S3 S4 — 01 15 14 13 — 05 07 09 11 — WE ME — 03 02

4E4 AMXS1 — 01 — SEL — 01J 19-132 2TO1MUX STTL (TRI-ST) — ENB 15
4B 4A 3B 3A 2B 2A 1B 1A — 4Y 3Y 2Y 1Y — 12 09 07 04
P5 — 22 02H 47 1K — 40 02E 37 1K — 44 02E 41 1K — 48 02E 45 1K

4E9 AMNØ

4E6 AWNØ
6A8 REGNAØ

MQ43Ø — MQ4ØØ — DIST21 — 2N8 — 4E6 BMØCLKØ — 3N2 MQCLRØ

01 — 1-3 — 1-2 — 1-1 — 1-0 — 11K 19-230 4-BIT SHIFTER (TRI-ST) — S1 S0 ENB 13 — Y0 Y1 Y2 Y3 — 15 14 12 11
MQ42Ø — MQ41Ø — MQ4ØØ — MQ39Ø — MQ38Ø — MQ37Ø — MQ36Ø

3N5 LMNBØ — 2N9 MIST2Ø

ENB 15 — 03H 19-196 2TO1MUX STTL (TRI-ST) — SEL 01
1A 1B 2A 2B 3A 3B 4A 4B — 1Y 2Y 3Y 4Y — 04 07 09 12
9H3 MAL44Ø — 9H3 MAL45Ø — 9H2 MAL46Ø — 9H2 MAL47Ø — 3N6 ALMBS1 — 3N6 LMNAØ

ENB 15 — 03J 19-196 2TO1MUX STTL (TRI-ST) — SEL 01
1A 1B 2A 2B 3A 3B 4A 4B — 1Y 2Y 3Y 4Y — 04 07 09 12
5B AL41Ø — 5B AL42Ø — 5B AL43Ø — 9S7 AL44Ø — 3N7 ALMAS1

01 — 1-3 — 1-2 — 1-1 — 1-0 — 02K 19-230 4-BIT SHIFTER (TRI-ST) — S1 S0 ENB 13 — Y0 Y1 Y2 Y3 — 15 14 12 11
7H2 MAL39Ø — 7H2 MAL38Ø — 7H3 MAL37Ø — 7H3 MAL36Ø

3N8 SHSLIØA — 3N9 SHSLØØB — 3N7 ALSRNØ — 4E6 ALCKØ — 5D8 DCLRØB

14T 19-057 SG — 13 12 — 11 10 — 09 08 — 05 06

115 — 19-131 (STTL) SDF — CLR 01 — D 09 12 05 04 — Q 01 14 10 07 06 02 03
15 NC — 14 MQ43Ø A2, K2, 9K3 — 10 NC — 11 MQ42Ø A2, K3, 9K3 — 07 NC — 06 MQ41Ø A1, K3, 9K3 — 02 NC — 03 MQ4ØØ A1, 7K2, K3, 9K3

03K — 19-131 (STTL) SDF — CLR 09 01 — D 05 12 13 — Q 02 03 07 06 10 11 15 14
AL4ØØ 7K7 — AL41Ø K6 — AL42Ø K6 — AL43Ø K6

AL43I
AL42I
AL41I
AL4ØI

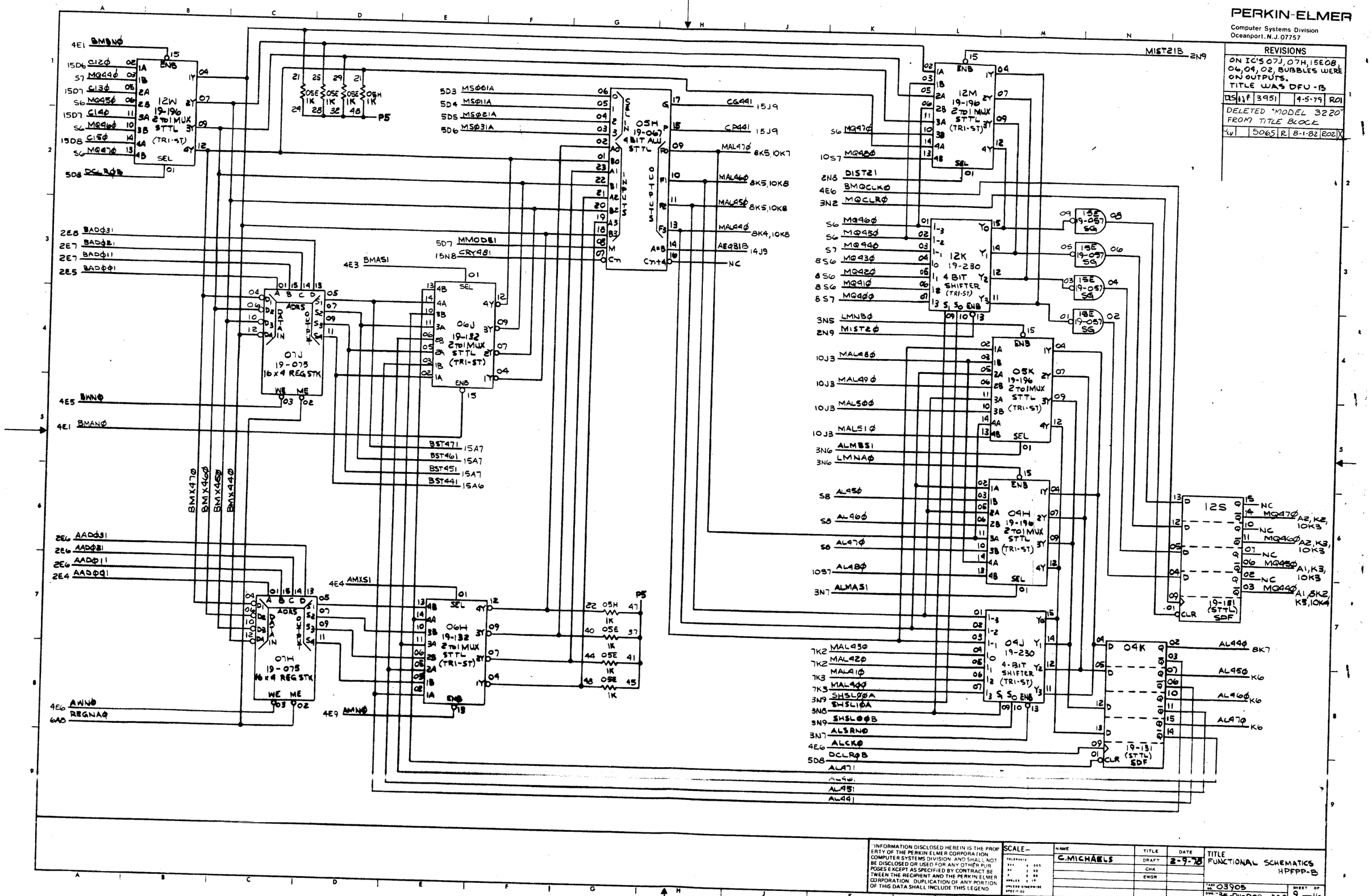
SCALE — TOLERANCE: XXX ± .005; XX ± .02; X ± .03; ANGLES ± 1°; UNLESS OTHERWISE SPECIFIED.

NAME	TITLE	DATE
C. MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE: FUNCTIONAL SCHEMATICS HPFPP-B
TASK NO. 03905
DWG NO. 35-716R02 D08
SHEET OF 8 — 16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS				
ON IC'S 07J, 07H, 15E08 06,04,02, BUBBLES WERE ON OUTPUTS. TITLE WAS DFU-B				
135	3951		4-5-79	R01
DELETED "MODEL 3220" FROM TITLE BLOCK				
4W	5065	R	8-1-82	R02 X



SCALE–	NAME	TITLE	DATE
TOLERANCE .XX ± .005, .XX ± .02, X ± 0 00, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED	C.MICHAELS	DRAFT	2-9-78
		CHK	
		ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-B

TABS NO. 03905
DWG NO. 35-716R02 D08
SHEET 9 OF 16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS					
ON IC'S 09H, 09J, 15F04, 02 12, 10 BUBBLES WERE ON OUTPUTS. IN AREA A9 RES. 05A 1K WAS DELETED, 05A13 WAS CONN TO P5, 05A04 WAS CONN TO 08E03.					
DS	JJP	3951		4-5-79	R01
DELETED "MODEL 3220" FROM TITLE BLOCK					
KW		5065	R	8-1-82	B02

MIST2IB 2N9

4E1 BMBNØ — 15 ENB — 13W 19-196 STTL 2 TO 1 MUX (TRI-ST) — SEL 01 — DCLRØB 5D8

15D8 S16Ø, 15D8 S17Ø, S6 MQ49Ø, 15H2 S18Ø, S6 MQ5ØØ, 15H2 S19Ø, S6 MQ51Ø

21 11E 1K, 25 11E 1K, 29 11E 1K, 21 11H 1K, 24 28 32 48 — P5

5D3 MSØØ1A, 5D4 MSØ11A, 5D5 MSØØ21A, 5D6 MSØ31A

11H 19-067 4 BIT ALU STTL (MSI) — A0 B0 A1 B1 A2 B2 A3 B3 M Cn — F0 F1 F2 F3 G P A=B Cn+4

17 CGA81 15J8; 15 CP481 15J8; 09 MAL51Ø 9K4, 11K7; 10 MAL5ØØ 9K5, 11K8; 11 MAL49Ø 9K5, 11K8; 13 MAL48Ø 9K5, 11K8; 14 AEQBIB 14J9; 16 NC

5D7 MMODØ81, 15N7 CRYS21

4E3 BMAS1 — 01 SEL — 10H 19-132 2 TO 1 MUX STTL (TRI-ST) — ENB 15

13 4B, 14 4A, 10 3B, 11 3A, 06 2B, 05 2A, 03 1B, 02 1A — 12 4Y, 09 3Y, 07 2Y, 04 1Y

BST511 15E2; BST501 15E1; BST491 15A8; BST481 15A8

2E8 BADØ31, 2E7 BADØ21, 2E7 BADØ11, 2E5 BADØØ1

09H 19-075 16 x 4 REGSTK — A B C D — D1 D2 D3 D4 DATA IN — ADRS — S1 S2 S3 S4 OUTPUTS — WE 03, ME 02

4E5 BWNØ; 4E1 BMANØ

BMX51Ø, BMX5ØØ, BMX49Ø, BMX48Ø

2E6 AADØ31, 2E6 AADØ21, 2E6 AADØ11, 2E4 AADØØ1

09J 19-075 16 x 4 REGSTK — A B C D — D1 D2 D3 D4 DATA IN — ADRS — S1 S2 S3 S4 OUTPUTS — WE 03, ME 02

4E4 AMXS1 — 01 SEL — 10J 19-132 2 TO 1 MUX STTL (TRI-ST) — ENB 15

13 4B, 14 4A, 10 3B, 11 3A, 06 2B, 05 2A, 03 1B, 02 1A — 12 4Y, 09 3Y, 07 2Y, 04 1Y

P5: 22 11H 1K 47; 40 11E 1K 37; 44 11E 1K 41; 48 11E 1K 45

4E9 AMNØ; 4E6 AWNØ; REGNBØ 11AB, 12AB, 13AB; 6A9 REGDISØ — 08E 19-057 SG

15 ENB — 13M 19-196 2 TO 1 MUX STTL (TRI-ST) — SEL 01 — DIST21 2N8, BMQCLKØ 4E6, MQCLRØ 9N2

S6 MQ51Ø, 11S7 MQ52Ø — 02 1A, 03 1B, 05 2A, 06 2B, 11 3A, 10 3B, 14 4A, 13 4B — 04 1Y, 07 2Y, 09 3Y, 12 4Y

S6 MQ5ØØ, S6 MQ49Ø, S7 MQ48Ø, 9S6 MQ47Ø, 9S6 MQ46Ø, 9S6 MQ45Ø, 9S7 MQ44Ø — 13K 19-230 4-BIT SHIFTER (TRI-ST) — I-3 I-2 I-1 I0 I1 I2 — Y0 Y1 Y2 Y3 — S1 S0 ENB 09 10 13

3N5 LMNBØ, 2N9 MIST2Ø — 15 ENB — 12H 19-196 2 TO 1 MUX STTL (TRI-ST) — SEL 01

11J3 MAL52Ø; 11J3 MAL53Ø; 11J3 MAL54Ø; 11J2 MAL55Ø; 3N6 ALMBS1, 3N6 LMNAØ

15 ENB — 11F 19-196 2 TO 1 MUX STTL (TRI-ST) — SEL 01

S8 AL49Ø; S8 AL5ØØ; S8 AL51Ø; 11S7 AL52Ø; 3N7 ALMAS1

15F 19-057 SG; 15F 19-057 SG; 13E 19-057 SG; 13E 19-057 SG

13S — D Q — 19-131 (STTL) SDF — CLR

15 NC; 14 MQS1Ø A2, K2, 11K3; 10 NC; 11 MQS5ØØ A2, K3, 11K3; 07 NC; 06 MQ49Ø A1, K3, 11K3; 02 NC; 03 MQ48Ø A1, 9K2, K5, 11K4

I-3 I-2 I-1 I0 I1 I2 — 12J 19-230 4-BIT SHIFTER (TRI-ST) — Y0 Y1 Y2 Y3 — S1 S0 ENB 09 10 13

9J2 MAL47Ø, 9J2 MAL46Ø, 9J3 MAL45Ø, 9J3 MAL44Ø

3N8 SHSLIØB, 3N9 SHSLØØA, 3N7 ALSRNØ, 4E6 ALCKØ, 5D8 DCLRØA

12F — D Q — 19-131 (STTL) SDF — CLR

AL48Ø 9K7; AL49Ø K6; AL5ØØ K6; AL51Ø K6

AL511, AL501, AL491, AL481

SCALE — TOLERANCE .XXX ± .005, .XX ± .02, .X ± .03, ANGLES ± 1°, UNLESS OTHERWISE SPECIFIED

NAME	TITLE	DATE
C. MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE FUNCTIONAL SCHEMATICS HPFPP-B

TASK NO. 05905 SYM NO. 35-716 R02 D08 SHEET OF 10—16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS				
ON IC'S 16H, 16J, 15F12, 10 08, 06 BUBBLES WERE ON OUTPUTS. TITLE WAS DFU-B				
105	NJP	3951		4-5-79	R01
DELETE "MODEL 3220" FROM TITLE BLOCK				
JW	5065	R	8-1-82	B02	X

MIST21B 2N9

4E1 BMBN0 — 14W 19-196 2 TO 1 MUX STTL (TRI-ST) — 5D8 DCLR0B

15H3 CG00 — 14M 19-196 2 TO 1 MUX STTL (TRI-ST)

5T MQ52Ø — 14K 19-230 4-BIT SHIFTER (TRI-ST)

15H3 C21Ø — 14H 19-067 4 BIT ALU STTL

15H3 C22Ø — 14F 19-196 2 TO 1 MUX STTL (TRI-ST)

5G MQ54Ø — 15H 19-132 2 TO 1 MUX STTL (TRI-ST)

15H4 CE3Ø — 15J 19-132 2 TO 1 MUX STTL (TRI-ST)

5G MQ55Ø — 13H 19-196 2 TO 1 MUX STTL (TRI-ST)

21 25 29 21 — 14E 1K — 24 28 32 48 — P5

5D3 MS00 1B
5D4 MS01 1B
5D5 MS02 1B
5D6 MS03 1B

CG521 15J7
CP521 15J7
MAL55Ø 10K3, 12K7
MAL54Ø 10K5, 12K7
MAL53Ø 10K9, 12K8
MAL52Ø 10K4, 12K8
AEQB1B 14J9
NC

5D7 MMOD81
15N6 CRY561

MQ55Ø
12S7 MQ56Ø
2N8 DIST21
4E6 BMQCLK0
3N2 MQCLR0
MQ54Ø
MQ53Ø
MQ52Ø
MQ51Ø
10S6 MQ50Ø
10S6 MQ49Ø
10S6 MQ48Ø
10S7

3N5 LMNB0
2N9 MIST0Ø

12J3 MAL56Ø
12J3 MAL57Ø
12J3 MAL58Ø
12J2 MAL59Ø
3N6 ALMB51
3N6 LMNA0

5B AL53Ø
5B AL54Ø
5B AL55Ø
12S7 AL56Ø
3N7 ALMA51

10J2 MAL51Ø
10J2 MAL50Ø
10J3 MAL49Ø
10J3 MAL48Ø

3N8 SHSL10B
3N9 SHSL00A
3N7 ALSRN0
4E6 ALCK0
5D8 DCLR0A

2E8 BAD031
2E7 BAD021
2E7 BAD011
2E8 BAD001

4E3 BMAS1

16H 19-075 16x4 REGSTK — ADRS, DATA IN, DATA OUT, WE, ME

4E5 BWN0
4E1 BMAN0

BST551 15E3
BST541 15E3
BST531 15E2
BST521 15E2

BMX55Ø
BMX54Ø
BMX53Ø
BMX52Ø

2E6 AAD031
2E6 AAD021
2E6 AAD011
2E6 AAD001

4E4 AMXS1

16J 19-075 16x4 REGSTK

4E6 AWN0
10A8 REGN30

4E9 AMN0

AL551
AL541
AL531
AL521

15F 19-057 SG — 13, 11, 09, 05 — 12, 10, 08, 06

14S — 13 D Q, 12 D, 11 D, 05 D, 04 D, 09 D — 15 NC, 14 MQ55Ø A2,K2, 12K3; 10 NC, 11 MQ54Ø A2,K3, 12K3; 07 NC, 06 MQ53Ø A1,K3, 12K4; 02 NC, 03 MQ52Ø A1,10K6 K3,12K4 — 19-131 (STTL) SDF

13F — 05 D, 12 D, 13 D, 09 D — 02 ALS2Ø 10K7; 03 ALS3Ø K6; 06 ALS4Ø K6; 10 ALS5Ø K6; 11, 15, 14 — 19-131 (STTL) SDF

SCALE—

NAME	TITLE	DATE
C. MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE FUNTIONAL SCHEMATICS HPFPP-B

TASK 03905 — DWG 33-716R02 D08 — SHEET OF 11—16

REVISIONS				
ON IC'S 16C, 16D, 16E04, 02 16F04, 02 BUBBLES WERE ON OUTPUTS. TITLE WAS DFU-B				
DS	3951		4-5-79	R01
DELETED "MODEL 3220" FROM TITLE BLOCK				
DV	5065	R	8-1-82	R02 X

SCALE—

TOLERANCE
XXX ± .063
XX ± .02
X ± .03
ANGLES ± 1°
UNLESS OTHERWISE SPECIFIED

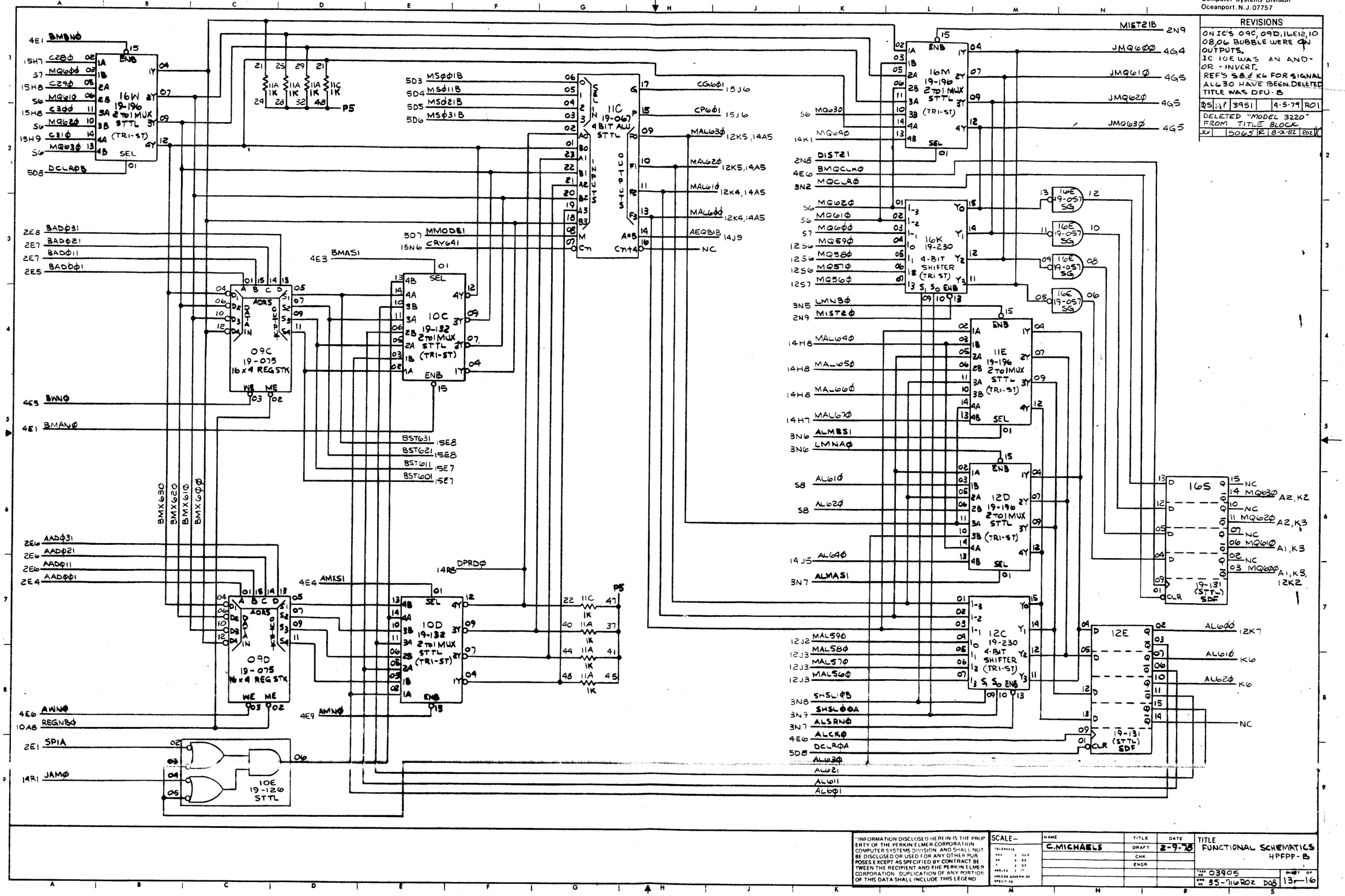
	NAME	TITLE	DATE
	C.MICHAELS	DRAFT	2-9-78
		CHK	
		ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-B

TASK NO. 03905
DWG. NO. 35-716 R02 D08
SHEET OF 12—16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS				
ON IC'S 09C, 09D, 16E12, 10 08, 06 BUBBLE WERE ON OUTPUTS. IC 10E WAS AN AND-OR-INVERT. REF'S 5B & K6 FOR SIGNAL AL630 HAVE BEEN DELETED TITLE WAS DFU-B				
05	JAF	3951	4-5-79	R01
DELETED "MODEL 3220" FROM TITLE BLOCK				
JLV		5065	R 8-2-82	R02



"INFORMATION DISCLOSED HEREIN IS THE PROPERTY OF THE PERKIN ELMER CORPORATION COMPUTER SYSTEMS DIVISION, AND SHALL NOT BE DISCLOSED OR USED FOR ANY OTHER PURPOSES EXCEPT AS SPECIFIED BY CONTRACT BETWEEN THE RECIPIENT AND THE PERKIN ELMER CORPORATION. DUPLICATION OF ANY PORTION OF THIS DATA SHALL INCLUDE THIS LEGEND

SCALE—

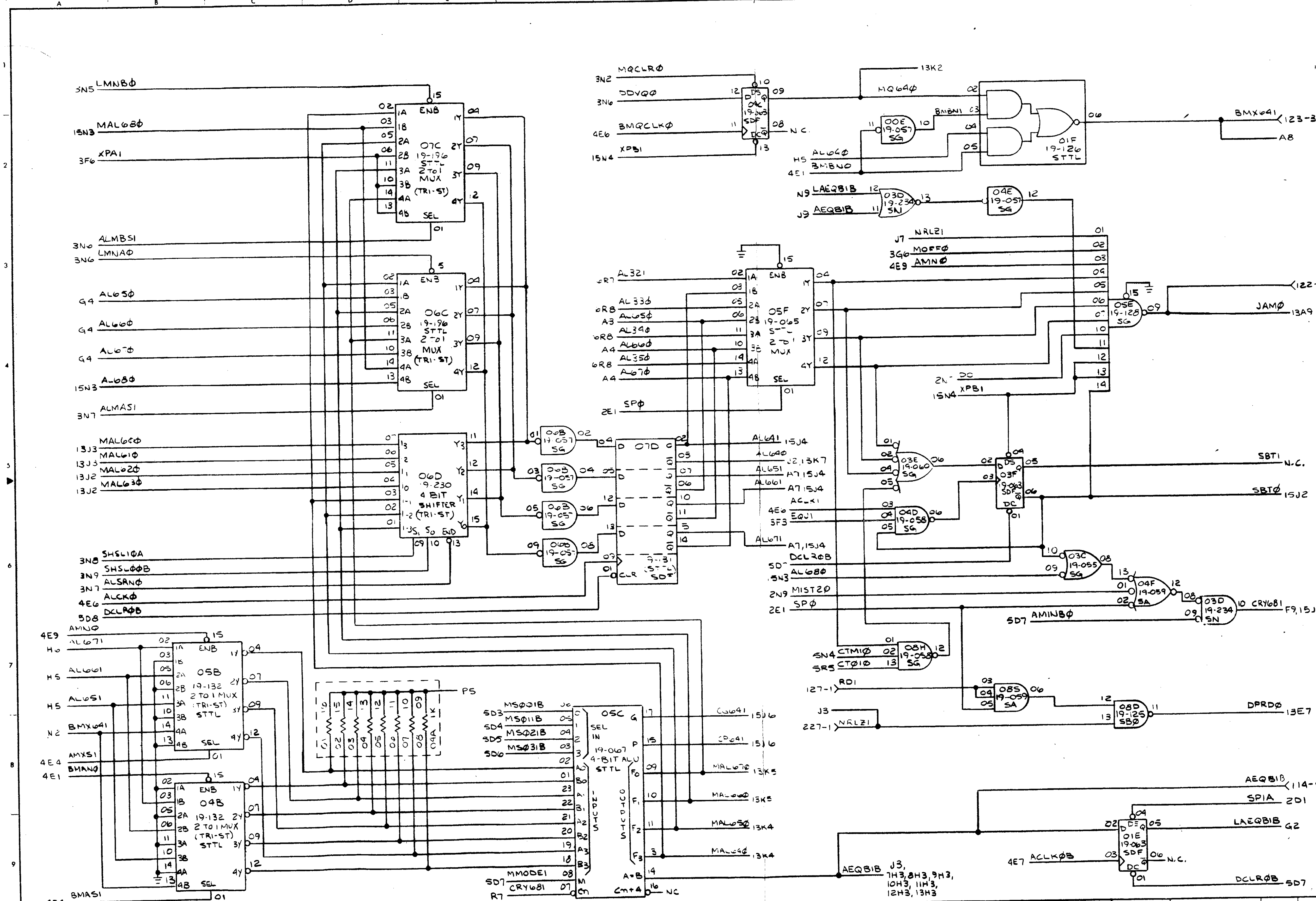
NAME	TITLE	DATE
C.MICHAELS	DRAFT	2-9-78
	CHK	
	ENGR	

TITLE
FUNCTIONAL SCHEMATICS
HPFPP-B

TABS NO 03905
DWG NO 55-716702 D08
SHEET 13 OF 16

PERKIN-ELMER

Computer Systems Division
Oceanport, N.J. 07757



REVISIONS

THE FOLLOWING WERE SHOWN AS AND GATES; 03E06, 03C08, 04F12, ON IC'S 06B02, 04, 06, 08, 00E10, & 04E12 BUBBLES WERE ON OUTPUTS. ON IC'S 04C, 03F, & 01E DS, Q, Q̄ & DC WERE S, 1, 0, & R RESPECTIVELY. TITLE WAS DFU-B. SIGNAL SHSLOOB WAS SHSLOO. IC 03D WAS AN OR GATE

RS	VJP	3951	4-6-79	R01

REMOVED 19-059 AREA K8. AREA R8 ADDED AEQ81B 114-4

KR		4096	M 7-16-79	R02

DELETED "MODEL 3220" FROM TITLE BLOCK

JLV		5065	R 8-1-82	R03

SCALE—

NAME		TITLE	DATE
C MICHAELS	DRAFT		2-13-78
	CHK		
	ENGR		

TITLE: FUNCTIONAL SCHEMATICS HPFPP-B

TASK NO 03905
DWG NO 35-716R03 D08
SHEET 14 OF 16

PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS				
GATE 03D04 WAS SHOWN AS AN OR GATE. ON IC 04C D5, Q, Q̄ & DC WAS S, I, O & R - RESPECTIVLY IN AREA J7 & J8 CAPACITORS 02B WAS 02A, PIN 4 WAS PIN 33, PIN 5 WAS PIN 34. ON IC 02E PIN 5 & 6 WERE A NO CONNECTION. TITLE WAS DFU-B.				
DS		3951		4-9-79 R01
DELETED "MODEL 3220" FROM TITLE BLOCK.				
JLV		5065	R	8-1-82 R02

09T
19-134
A

6E6 BST32I → 14
6E6 BST33I → 12
6E5 BST34I → 02
6E5 BST35I → 04
7E6 BST36I → 06
7E6 BST37I → 10

13 CΦØØ 6A1 (136-0)
11 CΦ1Φ 6A1 (236-0)
03 CΦ2Φ 6A2 (135-0)
05 CΦ3Φ 6A2 (235-0)
07 CΦ4Φ 7A1 (134-0)
09 CΦ5Φ 7A1 (234-0)

10T
19-134
A

7E5 BST38I → 14
7E5 BST39I → 12
8E6 BST4ØI → 02
8E6 BST41I → 04
8E5 BST42I → 06
8E5 BST43I → 10

13 CΦ6Φ 7A2 (133-0)
11 CΦ7Φ 7A2 (233-0)
03 CΦ8Φ 8A1 (132-0)
05 CΦ9Φ 8A1 (232-0)
07 C1ØΦ 8A2 (131-0)
09 C11Φ 8A2 (231-0)

11T
19-134
A

9E6 BST44I → 14
9E6 BST45I → 12
9E5 BST46I → 02
9E5 BST47I → 04
10E6 BST48I → 06
10E6 BST49I → 10

13 C12Φ 9A1 (130-0)
11 C13Φ 9A1 (230-0)
03 C14Φ 9A2 (129-0)
05 C15Φ 9A2 (229-0)
07 C16Φ 10A1 (110-0)
09 C17Φ 10A1 (210-0)

2N2 LSHENΦ

13T
19-134
A

10E5 BST5ØI → 14
10E5 BST51I → 12
11E6 BST52I → 02
11E6 BST53I → 04
11E5 BST54I → 06
11E5 BST55I → 10

13 C18Φ 10A2 (109-0)
11 C19Φ 10A2 (209-0)
03 C2ØØ 11A1 (108-0)
05 C21Φ 11A1 (208-0)
07 C22Φ 11A2 (107-0)
09 C23Φ 11A2 (207-0)

15T
19-134
A

NC
NC
NC

12E6 BST56I → 02
12E6 BST57I → 04
12E5 BST58I → 06
12E5 BST59I → 10

13 NC
11 NC
03 C24Φ 12A1 (106-0)
05 C25Φ 12A1 (206-0)
07 C26Φ 12A2 (105-0)
09 C27Φ 12A2 (205-0)

16T
19-134
A

202-0 SCLRØA

13E6 BST6ØI → 02
13E6 BST61I → 04
13E5 BST62I → 06
13E5 BST63I → 10

13 BSCLRØA 4G6
03 C28Φ 13A1 (104-0)
05 C29Φ 13A1 (204-0)
07 C3ØΦ 13A2 (103-0)
09 C31Φ 13A2 (203-0)

AL681

14N6 SBTØ → 01 of 01D 19-127 XORS (02, 03)
4E8 STØ3Ø → 05 of 03D 19-234 SN (06, 04)
3F3 EQU1
5D7 AMINB1

02C
19-116 STTL 4 TO 1 MUX
INPUTS 0: 06, 05, 04, 03 — 1: 01, 02, 14, 15 — 2: NC, NC, NC, NC — 3: NC, NC, NC, NC
STB1, SELB, SELA, STB2
1Y → 07
2Y → 09 NC

5D9 SBCLRØ
03D 19-234 SN: 02, 03 → 01
04E7 ALCKØB → 03 of 04C 19-063 SDF DC: 02, 04, 05, 06, 01
MAL68Ø → 14A2
AL68Ø → 14A4, 14H6
06 OSA 11 — 1K — PS
XPB1 → 14G2, 14K4

03B
19-116 STTL 4 TO 1 MUX
14H6 AL67I → 00
14H5 AL66I → 05
14H5 AL65I → 04
14H5 AL64I → 03
INPUTS 0: 00, 05, 04, 03 — 1: 01, 02, 14, 15 — 2: NC, NC, NC, NC — 3: NC, NC, NC, NC
3F4 EQUØ
4R6 MSCØ1
4R5 MSC11
STB1, SELB, SELA, STB2
1Y → 07
2Y → 09 NC

14H7 CRY6BI → 13 of 02D 19-068 STTL LOOK AHEAD CAR. GEN. (07, 12, 11, 09, 10)
14H8 CG64I → 03
14H8 CP64I → 04
13J2 CG6Ø1 → 01
13J2 CP6Ø1 → 02
12J2 CG56I → 14
12J2 CP56I → 15
11J2 CG52I → 05
11J2 CP52I → 06
CRY64I → 13E3
CRY6Ø1 → 12E3
CRY56I → 11E3

02E 19-068 STTL LOOK AHEAD CAR. GEN.
13, 03, 04, 01, 02, 14, 15, 05, 06
07 N.C.
12 CRY52I → 10E3
11 CRY36I → 6E3 → 123-4
09 CRY2Ø1 → 220-4
10 N.C.

221-4 CGX2Ø1
121-4 CPX2Ø1
04 02B 36 220pF 37 02A 32 100Ω
05 02B 35 220pF 38 02A 31 100Ω

10J2 CG48I → 03
10J2 CP48I → 04
9J2 CG44I → 01
9J2 CP44I → 02
8J2 CG4Ø1 → 14
8J2 CP4Ø1 → 15
7H2 CG36I → 05
7H2 CP36I → 06

02F 19-068 STTL LOOK AHEAD CAR. GEN. (07, 12, 11, 09, 10)
CRY48I → 9E3
CRY44I → 8E3
CRY4Ø1 → 7E3

SCALE —
TOLERANCE: XXX ± .005; XX ± .02; X ± .05; ANGLES ± 1°; UNLESS OTHERWISE SPECIFIED

NAME	TITLE	DATE
C. MICHAELS	DRAFT	2-13-78
	CHK	
	ENGR	

TITLE: FUNCTIONAL SCHEMATICS HPFPD-B
TASK NO 03905
DWG NO 35-7167R02 DOB
SHEET OF 15—16

PERKIN ELMER

Computer Systems
Oceanport N.J. 07757

CROSS REFERENCE NET#, MNEMONIC, SHEET#

NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.	NETS	MNEMONIC	SH.
0001	WCLK1	04	0063	1ØA12	05	0125	XCTØØØ	05	0187	13CØ7	12	0249	16DØ7	12	0311	ALS1Ø	10	0373	BMX32Ø	06	0435	BST591	12	0497	Ø1JØ7	08	0559	JVGT1	03	0621	Ø8HØ8	03	0683	PSØ2H	08	0745	R1	02	0807	Ø3WØ4	07
0002	EACLKØ	04	0064	11AØ7	05	0126	XCTØ1Ø	05	0188	13CØ9	12	0250	16DØ9	12	0312	ALS11	10	0374	BMX321	06	0436	BST6Ø1	13	0498	AEQBIA	14	0560	Ø7MØ8	04	0622	Ø8H12	14	0684	PSØ2S	07	0746	OBC11	04	0808	Ø3WØ7	07
0003	ACLK1	04	0065	11AØ9	05	0127	XCTØ2Ø	05	0189	13C12	12	0251	16D11	12	0313	ALS2Ø	11	0375	BMX33Ø	06	0437	BST611	13	0499	Ø4F12	14	0561	LØ	02	0623	Ø8KØ3	03	0685	PSØ3C		0747	RD1	14	0809	PSØ3K	06
0004	ALCKØ	04	0066	12AØ7	05	0128	XCTØ3Ø	05	0190	13MØ4	10	0252	16EØ2	12	0314	ALS21	11	0376	BMX34Ø	06	0438	BST621	13	0500	Ø4SØ4	06	0562	L1	02	0624	MQ33Ø	06	0686	PSØ3D		0748	Ø3DØ4	15	0810	PSØ2K	07
0005	BMQCLKØ	04	0067	12AØ9	05	0129	XCTØ4Ø	05	0191	13MØ7	10	0253	16EØ4	12	0315	ALS3Ø	11	0377	BMX35Ø	06	0439	BST631	13	0501	Ø4SØ7	06	0563	LENBØ	03	0625	MQ34Ø	06	0687	PSØ3E		0749	Ø3E06	14	0811	PSØ2E	08
0006	CGX2Ø1	15	0068	12BØ1	05	0130	XCTØ5Ø	05	0192	13MØ9	10	0254	16EØ6	13	0316	ALS31	11	0378	BMX36Ø	07	0440	BSTKNN1	04	0502	Ø4SØ9	06	0564	Ø7TØ5	06	0626	MQ35Ø	06	0688	PSØ3F		0750	Ø3F08	03	0812	PSØ5E	09
0007	CPX2Ø1	15	0069	12BØ4	05	0131	XCTA01Ø	04	0193	13M12	10	0255	16E08	13	0317	AL54Ø	11	0379	BMX37Ø	07	0441	03C03	03	0503	04S12	06	0565	Ø7TØ7	06	0627	MQ36Ø	07	0689	PSØ4C		0751	REGNAØ	06	0813	PS11E	10
0008	CRY2Ø1	15	0070	12BØ9	05	0132	XCTA02Ø	03	0194	ØGBØ2	14	0256	16E1Ø	13	0318	AL541	11	0380	BMX38Ø	07	0442	CØØØ	15	0504	01K06	05	0566	LOAD1	03	0628	MQ37Ø	07	0690	PSØ4D		0752	REGNBØ	10	0814	PS14E	11
0009	CRY361	15	0071	12B1Ø	05	0133	ØSBØ7	14	0195	Ø6BØ4	14	0257	16E12	13	0319	AL55Ø	11	0381	BMX39Ø	07	0443	C01Ø	15	0505	01C06	05	0567	LSHEN0	02	0629	MQ38Ø	07	0691	PSØ4E		0753	RR1	02	0815	PS11A	13
0010	CG321	06	0072	12B13	05	0134	ØSBØ9	14	0196	14BØ4	05	0258	16FØ2	12	0320	AL551	11	0382	BMX4ØØ	08	0444	CØ2Ø	15	0506	Ø7MØ5	09	0568	MØ71	03	0630	MQ39Ø	07	0692	PSØ4F		0754	SBMBNØ	04	0816	PSØ4A	14
0011	CØ3Ø	15	0073	13AØ7	05	0135	XMQ3Ø0	04	0197	14B1Ø	05	0259	16FØ4	12	0321	AL56Ø	12	0383	BMX41Ø	08	0445	CP3Ø	15	0507	Ø7MØ7	09	0569	M1	02	0631	MQ4ØØ	08	0693	PSØ5A		0755	SBTØ	14	0817	Ø3D13	14
0012	CØ9Ø	15	0074	13AØ9	05	0136	XMQ301	04	0198	14B13	04	0260	16JØ5	11	0322	AL561	12	0384	BMX42Ø	08	0446	CP4Ø	15	0508	CTCLKØ	03	0570	MIST2Ø	02	0632	MQ41Ø	08	0694	PSØ5C		0756	BSCLRØA	15	0818	Ø4E12	14
0013	C1ØØ	13	0075	13BØ1	05	0137	XMQ31Ø	04	0199	14FØ4	11	0261	16JØ7	11	0323	AL57Ø	12	0385	BMX43Ø	08	0447	CE5Ø	15	0509	DØ	02	0571	MIST21A	02	0633	MQ42Ø	08	0695	PSØ5H	09	0757	SEL1	02	0819	P508C	05
0014	C11Ø	15	0076	13BØ4	05	0138	XMQ311	04	0200	14FØ7	11	0262	16JØ9	11	0324	AL571	12	0386	BMX44Ø	09	0448	CGGØ	15	0510	D1	02	0572	MIST21B	02	0634	MQ43Ø	08	0696	PSØ55	06	0758	Ø3C08	14	0820	1ØBØ3	05
0015	C12Ø	15	0077	14AØ8	04	0139	1ØE06	13	0201	14FØ9	11	0263	16J11	11	0325	ALS8Ø	12	0387	BMX45Ø	09	0449	CØ7Ø	15	0511	DIST21	02	0573	Ø7TØ9	06	0635	MQ44Ø	09	0697	PSØ6A		0759	Ø9DØ5	13	0821	1ØBØ6	05
0016	C13Ø	15	0078	14BØ1	04	0140	1ØF02	06	0202	14F12	11	0264	16T13	15	0326	AL581	12	0388	BMX46Ø	09	0450	CG361	07	0512	DIST4Ø	04	0574	Ø8A1Ø	04	0636	MQ45Ø	09	0698	PSØ6B		0760	Ø9DØ7	13	0822	1ØB11	05
0017	C14Ø	15	0079	15AØ2	04	0141	1ØF04	06	0203	14MØ4	11	0265	Ø3M12	07	0327	AL59Ø	12	0389	BMX47Ø	09	0451	CG4Ø1	08	0513	DB8Ø	04	0575	Ø8A12	03	0637	MQ46Ø	09	0699	PSØ6E		0761	Ø9DØ9	13	0823	11BØ3	05
0018	C15Ø	15	0080	15AØ8	04	0142	1ØF06	06	0204	14MØ7	11	0266	Ø4BØ4	14	0328	AL591	12	0390	BMX48Ø	10	0452	CG441	09	0514	DCLKØ	03	0576	Ø8BØ8	05	0638	MQ47Ø	09	0700	PSØ6F		0762	Ø9D11	13	0824	CTM1Ø	05
0019	SCLRØA	15	0081	15BØ8	04	0143	1ØF08	06	0205	14M09	11	0267	Ø4BØ7	14	0329	AL6ØØ	13	0391	BMX49Ø	10	0453	CG481	10	0515	DCLRØA	05	0577	MAL32Ø	06	0639	MQ48Ø	10	0701	PSØ6K		0763	SKSIGØ	04	0825	SBCLRØ	05
0020	CP321	06	0082	16AØ2	04	0144	1ØF1Ø	07	0206	14M12	11	0268	Ø4BØ9	14	0330	AL6Ø1	13	0392	BMX5ØØ	10	0454	CGS21	11	0516	DCLR0B	05	0578	MAL33Ø	06	0640	MQ49Ø	10	0702	PSØ6M		0764	Ø9F1Ø	03	0826	Ø2A37	15
0021	C16Ø	15	0083	16AØ3	04	0145	1ØF12	07	0207	14TØ2	07	0269	ØØT1Ø	02	0331	AL61Ø	13	0393	BMX51Ø	10	0455	CG561	12	0517	DCLR1	05	0579	MAL34Ø	06	0641	MQ5ØØ	10	0703	PSØ6N		0765	SPØ	02	0827	Ø2A3Ø	15
0022	C17Ø	15	0084	16AØ6	04	0146	1ØJØ4	10	0208	14TØ4	07	0270	ACRYØ	03	0332	AL611	13	0394	BMX52Ø	11	0456	CG6Ø1	13	0518	DDVQØ	03	0580	MAL35Ø	06	0642	MQ51Ø	10	0704	PSØ7A		0766	SP1	02	0828	GNDØ2A	15
0023	C18Ø	15	0085	16AØ7	04	0147	1ØJØ7	10	0209	14TØ6	08	0271	OSB12	14	0333	AL62Ø	13	0395	BMX53Ø	11	0457	CG641	14	0519	LIST3Ø	04	0581	MAL36Ø	07	0643	MQ52Ø	11	0705	PSØ7B		0767	SP1A	02	0829	PSØ1W	02
0024	C19Ø	15	0086	16BØ4	04	0148	1ØJØ9	10	0210	14TØ8	08	0272	XILØ11		0334	AL621	13	0396	BMX54Ø	11	0458	CP361	07	0520	DPRDØ	14	0582	MAL37Ø	07	0644	MQ53Ø	11	0706	PSØ7E		0768	STØØ1	04	0830	GNDØ2E	15
0025	C2ØØ	15	0087	16BØ7	04	0149	1ØJ12	10	0211	14T1Ø	08	0273	AL32Ø	06	0335	AL63Ø	13	0397	BMX55Ø	11	0459	CP4Ø1	08	0521	DVQØ	03	0583	MAL38Ø	07	0645	MQ54Ø	11	0707	PSØ7F		0769	STØ1Ø	04	0831	REGDI5Ø	06
0026	C21Ø	15	0088	16BØ9	04	0150	1ØM04	07	0212	14T12	08	0274	AL321	06	0336	AL64Ø	14	0398	BMX56Ø	12	0460	CP441	09	0522	ØØTØ9	02	0584	MAL39Ø	07	0646	MQ55Ø	11	0708	PSØ7K		0770	STO11	04	0832	FNCEØ	06
0027	C22Ø	15	0089	16B12	04	0151	1ØM07	07	0213	Ø6BØ6	14	0275	AL33Ø	06	0337	AL641	14	0399	BMX57Ø	12	0461			0523	EQUØ	03	0585	MAL4ØØ	08	0647	MQ56Ø	12	0709	PSØ7M		0771	STØ2Ø	04	0833	ALGNØ	06
0028	C23Ø	15	0090	ALMAS1	03	0152	1ØM09	07	0214	Ø6BØ8	14	0276	AL331	06	0338	AL65Ø	14	0400	BMX58Ø	12	0462	CPS21	11	0524	EQU1	03	0586	MAL41Ø	08	0648	MQ57Ø	12	0710	PSØ7W		0772	STØ21	04	0834	MACØ	06
0029	AWNØ	04	0091	ALMBS1	03	0153	1ØM12	07	0215	Ø6H04	09	0277	AL34Ø	06	0339	AL651	14	0401	BMX59Ø	12	0463	CP561	12	0525	FØØ1	02	0587	MAL42Ø	08	0649	MQ58Ø	12	0711	PSØ8A		0773	STØ3Ø	04	0835		
0030	BWNØ	04	0092	ALSRNØ	03	0154	ALCKØB	04	0216	15DØ4	12	0278	AL341	06	0340	AL66Ø	14	0402	BMX6ØØ	13	0464	CP6Ø1	13	0526	FØ11	02	0588	MAL43Ø	08	0650	MQ59Ø	12	0712	PSØ8B		0774	STØ31	04	0836		
0031	DFTENØ	02	0093	AMNØ	04	0155	ØSFØ4	14	0217	15DØ7	12	0279	AL35Ø	06	0341	AL661	14	0403	BMX61Ø	13	0465	CP641	14	0527	FØ21	02	0589	MAL44Ø	09	0651	MQ6ØØ	13	0713	PSØ8D		0775	STØ4Ø	03	0837		
0032	SFTENØ	02	0094	AMXS1	04	0156	11EØ4	13	0218	15DØ9	12	0280	AL351	06	0342	AL67Ø	14	0404	BMX62Ø	13	0466	CRY4Ø1	15	0528	FCLKØ	03	0590	MAL45Ø	09	0652	MQ61Ø	13	0714	PSØ8E		0776	STØ5Ø	03	0838		
0033	RDØ9Ø	02	0095	BMANØ	04	0157	11EØ7	13	0219	15D12	12	0281	AL36Ø	07	0343	AL671	14	0405	BMX63Ø	13	0467	CRY441	15	0529	GNDØØT	02	0591	MAL46Ø	09	0653	MQ62Ø	13	0715	PSØ8F		0777	Ø3WØ9	07	0839		
0034	RD1Ø	02	0096	BMAS1	04	0158	11EØ9	13	0220	15E02	09	0282	AL361	07	0344	AL68Ø	15	0406	BMX641	14	0468	CRY481	15	0530	GNDØØW	02	0592	MAL47Ø	09	0654	MQ63Ø	13	0716	PSØ8H		0778	VGTØ	03	0840		
0035	RD11Ø	02	0097	BMBNØ	04	0159	11E12	13	0221	15E04	09	0283	AL37Ø	07	0345	AL681	15	0407	BONLYØ	05	0469	CRY521	15	0531	GNDØ4B	14	0593	MAL48Ø	10	0655	MQ64Ø	14	0717	PSØ8J		0779	ØØSØ8	02	0841		
0036	RD251	02	0098	CTØ1Ø	05	0160	11MØ4	08	0222	15E06	09	0284	AL371	07	0346	ØØT11	02	0408	BST321	06	0470	CRY561	15	0532	GNDØ5B	14	0594	MAL49Ø	10	0656	MQCEØ	03	0718	PSØ8K		0780	XALØ2Ø	05	0842		
0037	YDØ81	02	0099	CTØ11	05	0161	11MØ7	08	0223	15E08	09	0285	AL38Ø	07	0347	Ø6TØ9	06	0409	BST331	06	0471	CRY6Ø1	15	0533	GNDØ5E	14	0595	MAL5ØØ	10	0657	MQCLRØ	03	0719	PSØ8M		0781	XALØ3Ø	05	0843		
0038	YDØ91	02	0100	JMQ28Ø	04	0162	11MØ9	08	0224	15E1Ø	10	0286	AL381	07	0348	Ø6T12	06	0410	BST341	06	0472	CRY641	15	0534	GNDØ5F	14	0596	MAL51Ø	10	0658	MSØØ1A	05	0720	PSØ8S		0782	XALØ4Ø	05	0844		
0039	YD1Ø1	02	0101	JMQ29Ø	04	0163	11M12	08	0225	15E12	10	0287	AL39Ø	07	0349	ØØW1Ø	02	0411	BST361	06	0473	CRY681	15	0535			0597	MAL52Ø	11	0659	MSØØ1B	05	0721	PSØ8T		0783	XALØ5Ø	05	0845		
0040	YD111	02	0102	JMQ3ØØ	04	0164	ØSFØ7	14	0226	15FØ2	10	0288	AL391	07	0350	AMINBØ	05	0412	BST361	07	0474	CBENØ	02	0536	GNDØ9E	03	0598	MAL53Ø	11	0660	MSØ11A	05	0722	PSØ8W		0784	XALØ6Ø	05	0846		
0041	YSØ1	02	0103	JMQ31Ø	04	0165	ØSFØ9	14	0227	15FØ4	10	0289	AL4Ø1	08	0351	AMINB1	05	0413	BST371	07	0475	Ø1JØ9	08	0537	GND11A	05	0599	MAL54Ø	11	0661	MSØ11B	05	0723	PSØ9A		0785	XALØ7Ø	05	0847		
0042	YS11	02	0104	JXCRYØ	05	0166	ØSF12	14	0228	15FØ6	11	0290	AL4Ø1	08	0352	Ø7A12	03	0414	BST381	07	0476	Ø1J12	08	0538	GND12A	05	0600	MAL55Ø	11	0662	MSØ21A	05	0724	PSØ9B		0786	XCRYØ	03	0848		
0043	YS21	02	0105	LMNAØ	03	0167	ØSKØ4	09	0229	15FØ8	11	0291	AL41Ø	08	0353	AMXSØ	04	0415	BST391	07	0477	Ø1KØ5	08	0539	GND13A	05	0601	MAL56Ø	12	0663	MSØ21B	05	0725	PSØ9E		0787	XCRY1	03	0849		
0044	UNNLDØ	04	0106	LMNBØ	03	0168	ØSKØ7	09	0230	15F1Ø	11	0292	AL411	08	0354	Ø7C04	14	0416	BST4Ø1	08	0478	Ø1K07	08	0540	GND13B	05	0602	MAL57Ø	12	0664	MSØ31A	05	0726	PSØ9F		0788	Ø9JØ5	10	0850		
0045	AADØØ1	02	0107	MAL28Ø	06	0169	ØSKØ9	09	0231	15F12	11	0293	AL42Ø	08	0355	AONLY1	05	0417	BST411	08	0479	Ø1K09	08	0541	GND15B	04	0603	MAL58Ø	12	0665	MSØ31B	05	0727	PS1ØA		0789	Ø9J07	10	0851		
0046	AADØ11	02	0108	MAL29Ø	06	0170	ØSK12	09	0232	15J04	11	0294	AL421	08	0356	APLSBØ	05	0418	BST421	08	0480	Ø1K11	08	0542	GND16B	04	0604	MAL59Ø	12	0666	Ø8SØ6	14	0728	PS1ØE		0790	Ø9J09	10	0852		
0047	AADØ21	02	0109	MAL3ØØ	06	0171	12HØ4	10	0233	15J07	11	0295	AL43Ø	08	0357	AS1	02	0419	BST431	08	0481	Ø1S08	02	0543	GND16T	15	0605	MAL6ØØ	13	0667	ØSBØ4	14	0729	PS1ØF		0791	Ø9J11	10	0853		
0048	AADØ31	02	0110	MAL31Ø	06	0172	12HØ7	10	0234	15J09	11	0296	AL431	08	0358	AST321	06	0420	BST441	09	0482	Ø1W0Ø	02	0544	JACRYØ	03	0606	MAL61Ø	13	0668	MSHØ	02	0730	PS11C	13	0792	Ø9MØ4	06	0854		
0049	BADØØ1	02	0111	MCOUT1	03	0173	12HØ9	10	0235	15J12	11	0297	AL44Ø	09	0359	ASTKWN1	04	0421	BST451	09	0483	Ø1W1Ø	02	0545	JAMØ	13	0607	MAL62Ø	13	0669	MSUM1	04	0731	PS11H	10	0793	Ø9MØ7	06	0855		
0050	BADØ11	02	0112	MQ28Ø	06	0174	12H12	10	0236	15M04	12	0298	AL441	09	0360	Ø3BØ7	15	0422	BST461	09	0484	ØØW12	02	0546	AEQB1B	14	0608	MAL63Ø	13	0670	Ø9B1Ø	05	0732	PS12T		0794	Ø9MØ9	06	0856		
0051	BADØ21	02	0113	MQ29Ø	06	0175	12MØ4	09	0237	15MØ7	12	0299	AL45Ø	09	0361	Ø4B12	14	0423	BST471	09	0485	Ø1W12	02	0547	JBMXØ	04	0609	MAL64Ø	14	0671	PSØØE		0733	PS14A		0795	Ø9M12	06	0857		
0052	BADØ31	02	0114	MQ3ØØ	06	0176	12MØ7	09	0238	15M09	12	0300	AL451	09	0362	Ø4DØ6	14	0424	BST481	10	0486	Ø2C67	15	0548	JBMXAS1	04	0610	MAL65Ø	14	0672	PSØØF		0734	PS14B		0796	1ØDØ4	13	0858		
0053	C24Ø	15	0115	MQ31Ø	06	0177	12M09	09	0239	15M12	12	0301	AL46Ø	09	0363	Ø4DØ8	03	0425	BST491	10	0487	Ø2D07	15	0549	Ø7H09	09	0611	MAL66Ø	14	0673	PSØØH		0735	PS14C	12	0797	1ØDØ7	13	0859		
0054	C25Ø	15	0116	MQ32Ø	06	0178	12M12	09	0240	Ø6H07	09	0302	AL461	09	0364	Ø4D12	03	0426	BST5Ø1	10	0488	Ø2D1Ø	15	0550	Ø7H11	09	0612	MAL67Ø	14	0674	PSØØJ		0736	PS14H	11	0798	1ØDØ9	13	0860		
0055	C26Ø	13	0117	MSC01	04	0179	12T02	03	0241	Ø6H09	09	0303	AL47Ø	09	0365	BADMS1	03	0427	BST511	10	0489	Ø2F07	15	0551	Ø7K08	04	0613	MAL68Ø	15	0675	PSØØM		0737	PS14T		0799	LAEQB1B	14	0861		
0056	C27Ø	15	0118	MSC11	04	0180	12T08	03	0242	Ø6H12	09	0304	AL471	09	0366	BGTØ	03	0428	BST521	11	0490	Ø2F1Ø	15	0552	07K11	04	0614	Ø8B11	03	0676	PSØØS		0738	PS15A		0800	XPA1	03	0862		
0057	C28Ø	15	0119	NRL21	14	0181	12T1Ø	04	0243	Ø6K12	03	0305	AL48Ø	10	0367	BGT1	03	0429	BST531	11	0491	Ø2M04	07	0553	JMQ6ØØ	13	0615	MFNØ	03	0677	PSØ1D		0739	PS15B		0801	XPB1	15	0863		
0058	C29Ø	15	0120	SHSL0ØA	03	0182	Ø5W05	07	0244	Ø6M06	04	0306	AL481	10	0368	Ø7C07	14	0430	BST541	11	0492	Ø2M07	07	0554	JMQ61Ø	13	0616	MMODE1	05	0678	PSØ1E		0740	PS15E		0802	Ø3F09	03	0864		
0059	C3ØØ	15	0121	SHSL0ØB	03	0183	Ø5W07	07	0245	Ø6M08	04	0307	AL49Ø	10	0369	Ø7C09	14	0431	BST551	11	0493	Ø2M09	07	0555	JMQ62Ø	13	0617	MOFFØ	03	0679	PSØ1F		0741	PS15F		0803	Ø3H04	08	0865		
0060	C31Ø	15	0122	SHSL1ØA	03	0184	Ø5W09	07	0246	Ø6T04	06	0308	AL491	10	0370	Ø7C12	14	0432	BST561	12	0494	Ø2M12	07	0556	JMQ63Ø	13	0618	MOFF1	03	0680	PSØ1M		0742	PS16E		0804	Ø3H07	08	0866		
0061	Ø9A06	04	0123	SHSL1ØB	03	0185	05W11	07	0247	Ø6T07	06	0309	AL5ØØ	10	0371	BMBN1	14	0433	BST571	12	0495	Ø1D03	15	0557	JVFLGØ	04	0619	Ø8E1Ø	04	0681	PSØ1S		0743	PS16F		0805	Ø3HØ9	08	0867		
0062	1ØA08	05	0124	SNRLZØ	03	0186	13CØ4	12	0248	16DØ5	12	0310	AL5Ø1	10	0372	ØØT15	02	0434	BST581	12	0496	Ø1J04	08	0558	JVGTØ	03	0620	Ø8E12	03	0682	PSØ1T		0744	RO	02	0806	Ø3H12	08	0868		

REVISIONS

DELETED NETS 0504Ø 0505.
NETS 0829 THRU 0834 WERE NOT SPECIFIED.
TITLE WAS DFU-B

RS	JJ	3951		4-9-79	R01

NET 272 WAS AEQBIC-06
ADDED NETS 504,505 & 819

KR	JJ	4096	M	7-13-79	R02

NET 535 WAS GND08F-03

BP	JJ	4146	M	9-20-79	R03

NET 461 WAS CP481 SHT.10

JJ	JJ	4162	M	11-2-79	R04

DELETED "MODEL 3220" FROM TITLE BLOCK

JLV		5065	R	8-1-82	R05

SCALE —

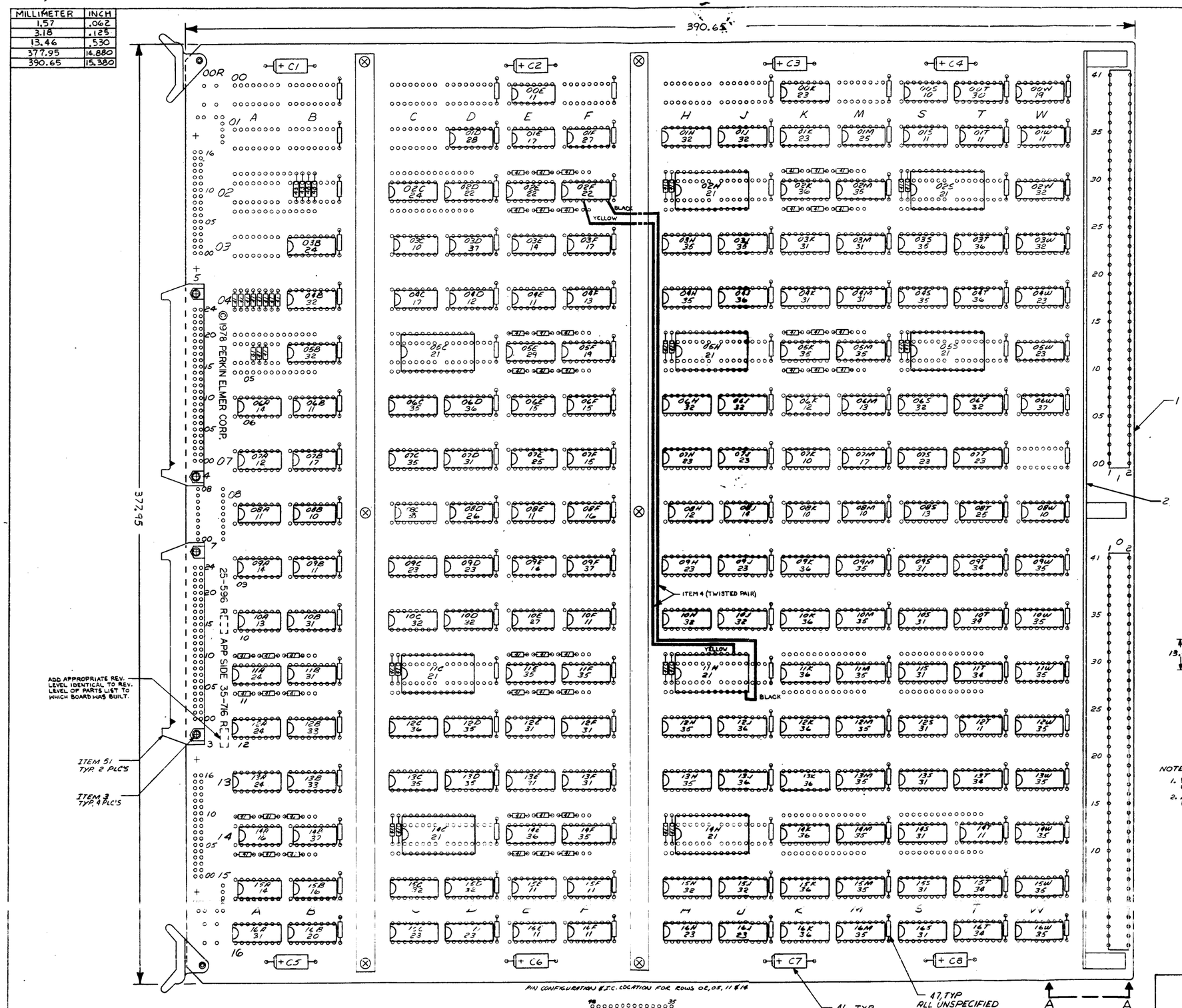
NAME	TITLE	DATE
C MICHAELS	DRAFT	12-6-78
	MFR	
	ENGR	

TITLE MULTIWIRE FUNCTIONAL SCHEMATICS HPFPD-B

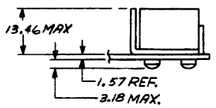
DWG NO 03905 SHEET OF
REV 35-716R05 D08 16 — 16

MILLIMETER	INCH
1.57	.062
3.18	.125
13.46	.530
377.95	14.880
390.65	15.380

REVISIONS			
PRE PRODUCTION APPROVAL	DEV PROD	MFG DATE	
DELETED 2-1K RES-STORS AT GSA POS. 04 & U7 TITLE WAS DFU-B			
03	18	3951	3-30-79 R01
RELEASED FOR PRODUCTION MFG ENG ____ DATE ____			
LOC. GS REMOVED RES. BETWEEN PINS 2 & 4 & 10 ADDED R: BETWEEN PINS 4 & 10.			
KR	11	4096	M 7-17-79 R02
ITEM 4 (TWISTED PAIR) WAS NOT SPEC'D.			
KR	11	4162	M 10-30-79 R03
AT LOCATION 08C, I.C. WAS SHOWN AS A 16 PIN I.C. IN ERROR.			
AT	RS	4190-30	11 5-11-81 R04



METRIC



PARTIAL VIEW A-A
TYP 3 PLACES

NOTES:
1. FOR MOUNTING OF STANDARD HARDWARE SEE 16-642 D12.
2. ALL DIMENSIONS ARE IN MILLIMETERS.

COMPONENT	REF. DESIGNATION

PERKIN ELMER
Computer Systems Division
Oceanport, N.J. 07757

NAME	TITLE	DATE
C. MANOUSELIS / J. STINE	DRAFT	9-25-78
E. CERO	CHK	2-11-79
M. KELLEHER	ENGR	2-11-79
R. BARONE	QC	2-11-79
M. LEMONDE	MGR	6-11-79

TITLE ASSEMBLY MODEL 8/32 MOD-B MULTIBUS

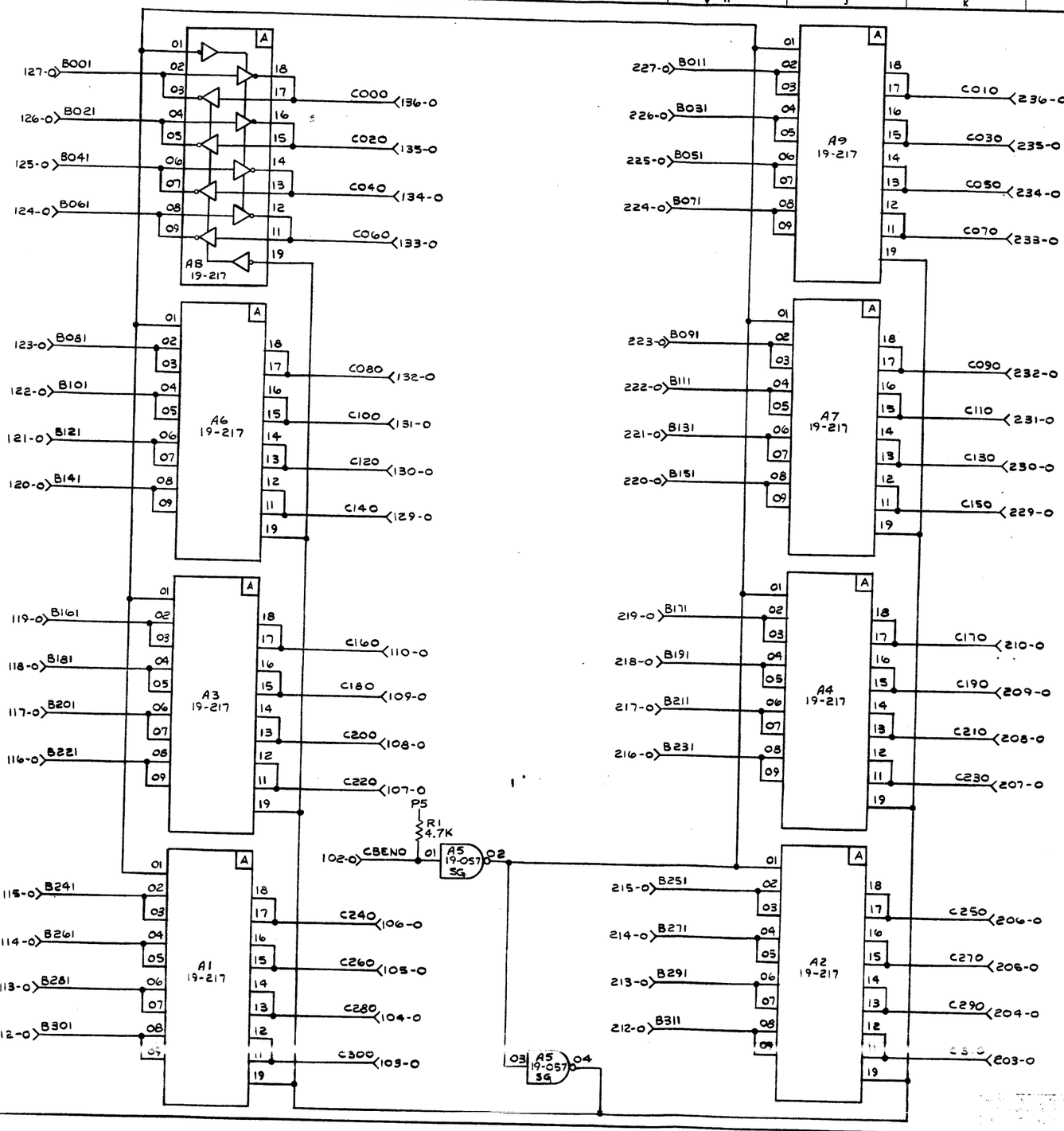
USED IN MANUAL 29-705

SIZE C | 29-716 R04 | SHEET 1 OF 1

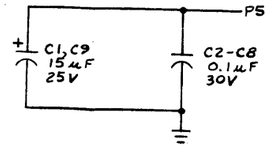
PERKIN-ELMER
Computer Systems Division
Oceanport, N.J. 07757

REVISIONS

PRE PRODUCTION APPROVAL	DFV	INIT	DATE
AREA E7, "CBENO" WAS 202-0 AREA N6, PIN NO. 102-0 WAS 202-0. AREA N9, 35-735 WAS R00.			
JPB	RC	3924	2-9-79 R01
RELEASED FOR PRODUCTION ENG. DATE			



CONN	TERM. NO.	BACK PANEL MAP ROW 1	ROW 2
O	41	P5	GND
	40	GND	GND
	39		
	38		
	37		
	36	C000	C010
	35	C020	C030
	34	C040	C050
	33	C060	C070
	32	C080	C090
	31	C100	C110
	30	C120	C130
	29	C140	C150
	28		
	27	B001	B011
	26	B021	B031
	25	B041	B051
	24	B061	B071
	23	B081	B091
	22	B101	B111
	21	B121	B131
	20	B141	B151
	19	B161	B171
	18	B181	B191
	17	B201	B211
	16	B221	B231
	15	B241	B251
	14	B261	B271
	13	B281	B291
	12	B301	B311
	11		
	10	C160	C170
	9	C180	C190
	8	C200	C210
	7	C220	C230
	6	C240	C250
	5	C260	C270
	4	C280	C290
	3	C300	C310
	2	CBENO	
	1	GND	GND
	00	P5	GND



USED IN MANUAL 29-705

35-735	R01
BOARD	REV LEVEL

	NAME	DRAFT	
SCALE-	C. MICHAELS	DRAFT	3-3-78
TOLERANCE	E. GREENSTEIN	SYS TEST	5-11-79
.XX ± .005	D CERO	CHK	5-11-79
.X ± .02	G JOYCE	ENGR	5-11-79
ANGLES ± 1°	P.A. BARKER	Q.C.	5-11-79
UNLESS OTHERWISE SPECIFIED	N. LEMONDE	MGR.	5-11-79

TITLE FUNCTIONAL SCHEMATIC C-BUS INTERFACE
TAPE NO. 03905
DWG NO. 35-735 R01 D08
SHEET OF 1 — 1

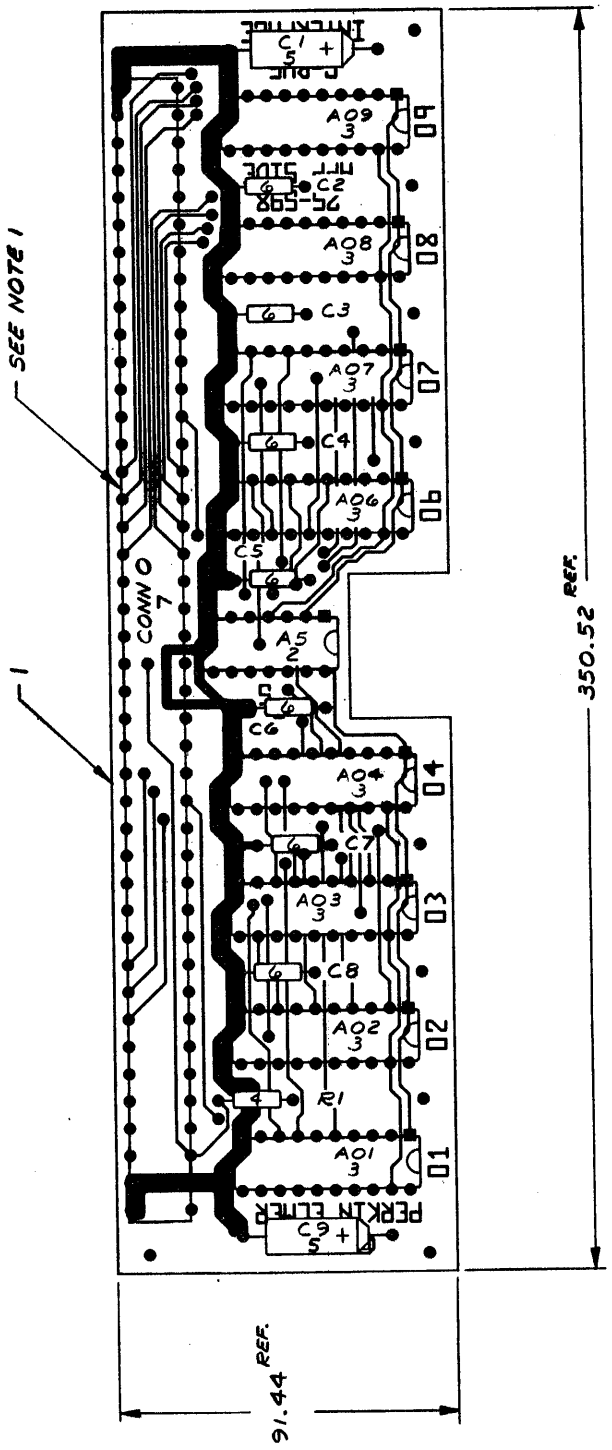
PERKIN-ELMER

REVISIONS			
PRE PRODUCTION APPROVAL	INIT	DATE	
	DEV		
	PROD		
REVISED CIRCUITRY TO AGREE WITH R01 COPPER.			
JRB	RC	3924	2-9-79 R01
RELEASED FOR PRODUCTION MFG. ENG.		DATE	
REVISED TO REFLECT R02 COPPER CHANGE.			
KR	911	4786	R 10-2-81 R02

NOTES
1. CONNECTOR PINS CLOSEST TO EDGE OF BOARD ARE TO BE BENT INWARD PRIOR TO SOLDERING.
2. DIMENSIONS ARE IN MILLIMETERS.

MILLIMETERS	INCHES
91.44	3.6
350.52	13.8

COMPONENT	REF. DESIGNATION
I.C.	A01 - A09
RESISTOR	R1
CAPACITOR	C1 - C9



USED IN MANUAL 29-705

METRIC

TOLERANCE IN MILLIMETERS
X ± .8
.X ± .5
.XX ± .13
UNLESS OTHERWISE SPECIFIED

SCALE - 2:1

NAME	TITLE	DATE
P.MARCUS/J.BIELSKIS	DRAFT	12-4-78
R. CERO	CHK	5-11-79
G. JOYCE	ENGR	5-11-79
R.A. BARKER	Q.C.	5-11-79
N. LEMONDE	MGR	5-11-79

TITLE ASSEMBLY C-BUS INTERFACE
TAPE NO. 03305
DWG. NO. 35-735 R02 C03
SHEET 1 OF 1

INDEX

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From _____ Date _____

Title _____ Publication Title _____

Company _____ Publication Number _____

Address _____

_____

_____

FOLD FOLD

Check the appropriate item.

☐ Error Page No. _____ Drawing No. _____

☐ Addition Page No. _____ Drawing No. _____

☐ Other Page No. _____ Drawing No. _____

Explanation:

FOLD FOLD

CUT ALONG LINE

Fold and Staple

No postage necessary if mailed in U.S.A.

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

PERKIN-ELMER

Computer Systems Division
2 Crescent Place
Oceanport, NJ 07757

TECH PUBLICATIONS DEPT. MS 322A

FOLD

FOLD

STAPLE

STAPLE