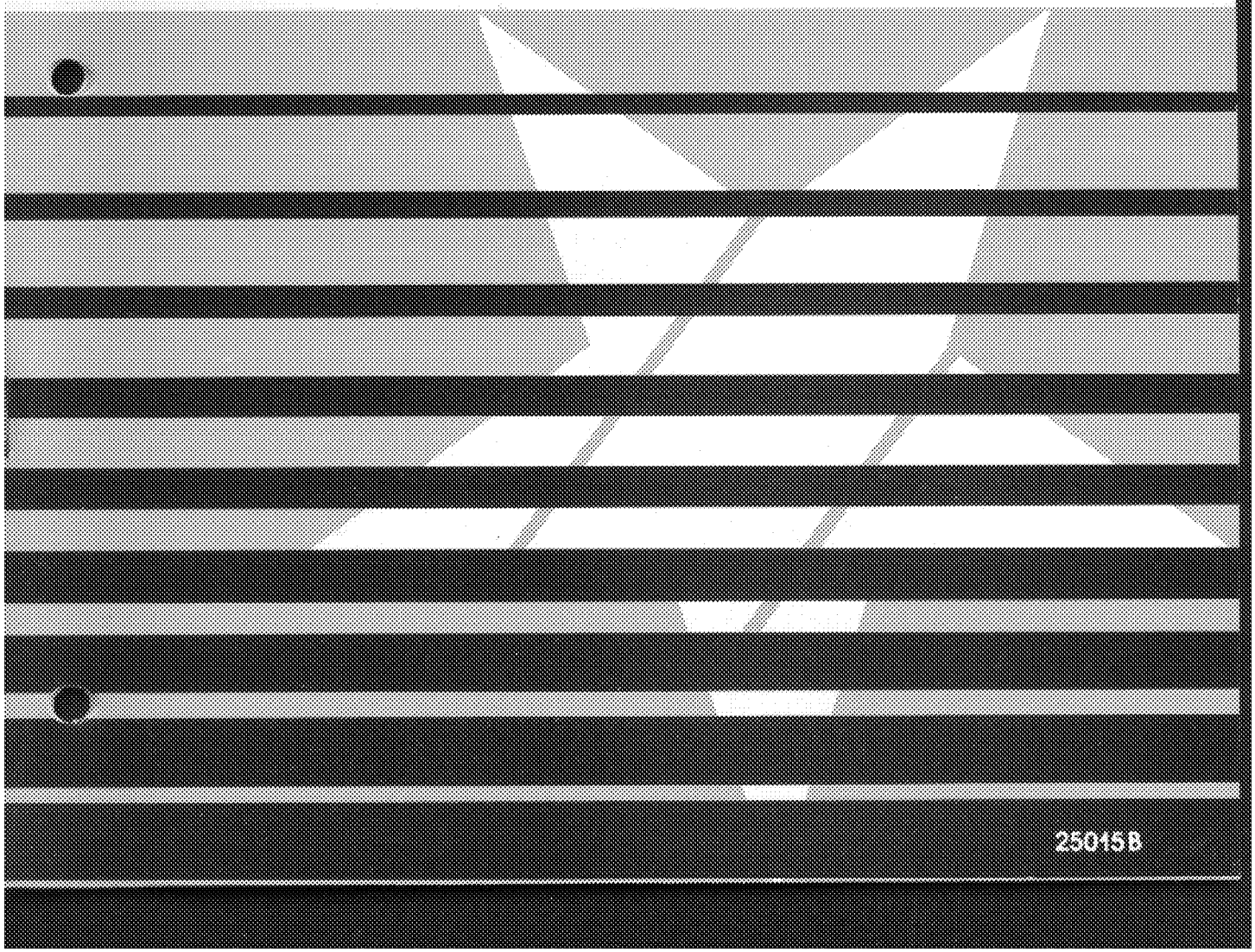
NorthStar™

Hardware Floating Point Board FPB-A Manual



25015B

NorthStar™
North Star Computers, Inc.
14440 Catalina St., San Leandro, CA 94577 USA
(415) 357-8500 TWX/Telex (910) 366-7001

Hardware Floating Point Board
FPB-A Manual

24136B

25015B

Contents

CAUTIONS

1. Correct this document from the addenda, if any, before doing anything else.

2. DO NOT insert or remove the FPB from the computer while the power is turned on.

3. DO NOT insert or remove IC's from the board while the power is turned on.

WARRANTY

North Star Computers, Inc., warrants the electrical and mechanical parts and workmanship of this product to be free of defects for a period of 90 days from date of purchase. If such defects occur, North Star Computers, Inc., will repair the defect at no cost to the purchaser. This warranty does not extend to defects resulting from improper use or assembly by purchaser, nor does it cover transportation to the factory. Also, the warranty is invalid if all instructions included in the accompanying documentation are not carefully followed. Any adjustment, modification, or disassembly of any disk drive unit will void the warranty on that disk drive unit. Should a unit returned for warranty repair be deemed by North Star Computers, Inc., to be defective due to purchaser's action, then a repair charge (not to exceed $50 without purchaser's consent) will be assessed. ANY UNIT(S) OR PART(S) RETURNED FOR WARRANTY REPAIR MUST BE ACCOMPANIED BY A DATED COPY OF THE ORIGINAL SALES RECEIPT. The item should be returned to the dealer from whom the product was purchased, for implementation of the warranty. When sending the item to the factory for repair, the dealer must call the North Star Technical Hotline to receive a Return Material Authorization (RMA) number to accompany the item to the factory. Terminals and printers are covered under separate warranties.

The following warranty limitation applies to units located outside the United States of America: All costs and arrangements for transportation of the product to and from the factory are borne entirely by the customer.

No warranty, expressed or implied, is extended concerning completeness, correctness, or suitability of the North Star equipment for any particular application. There are no warranties which extend beyond those expressly stated herein. This limited warranty is made in lieu of all other warranties, expressed or implied, and is limited to repair or replacement of the product.

USING THE FLOATING POINT BOARD (FPB-A)

To use the current BASIC VERSION 5-FPB with a current board, FPB-A, no action is necessary. The standard addresses for both are RESTART=DFF2 and DATAIN=DFF1. To verify this, skip to the CHECK-OUT section of this manual. To alter the BASIC, refer to the North Star System Software Manual. To alter the board, read the following section on ADDRESS SELECTION.

ADDRESS SELECTION

Before using the floating point board, the memory addresses to which the FPB responds must be selected with jumper wires. The jumpers determine the four most significant bits of the addresses. The remaining 12 bits are fixed on the board. The jumper area is located just above IC XR. There are four pads, labeled 15, 14, 13, and 12, associated with the four most significant address bits. The two pads at the ends provide ground (G) and high logic level (H). All address bits that should be "ones" should be daisy chain jumpered to G and all "zero" bits should be jumpered to H. The following table shows the RESTART and DATAIN addresses in hexadecimal corresponding to the 16 possible combinations.

15	14	13	12	RESTART	DATAIN
H	H	H	H	0FF2	0FF1
H	H	H	G	1FF2	1FF1
H	H	G	H	2FF2	2FF1
H	H	G	G	3FF2	3FF1
H	G	H	H	4FF2	4FF1
H	G	H	G	5FF2	5FF1
H	G	G	H	6FF2	6FF1
H	G	G	G	7FF2	7FF1
G	H	H	H	8FF2	8FF1
G	H	H	G	9FF2	9FF1
G	H	G	H	AFF2	AFF1
G	H	G	G	BFF2	BFF1
G	G	H	H	CFF2	CFF1
G	G	H	G	DFF2	DFF1
G	G	G	H	EFF2	EFF1
G	G	G	G	FFF2	FFF1

NOTE: For proper operation no other memory in the computer can respond to addresses in the same block of sixteen as RESTART and DATAIN. For example, if RESTART and DATAIN are BFF2 and BFF1, then the FPB requires full use of addresses in the range BFF0-BFFF.

PROGRAMMING

The 8080 or Z80 program for causing the FPB to perform an arithmetic operation must perform the following steps in sequence:

1. Restart the FPB with a memory read of the RESTART address.

2. Send the command byte which specifies the precision and operation to be performed. The command byte is the first data byte (i.e. not the first byte of an instruction) read after the RESTART.

3. Send the N bytes of the right operand. These are the N data bytes read after the command byte. No other data bytes may be read during the transmission of arguments.

4. Send the N bytes of the left operand. These are the next N data bytes read. The FPB immediately starts performing the specified operation after the Nth data byte is read.

5. Wait for and receive the result status byte. The status byte is read by performing a read to the DATAIN address. If the sign bit of this value is zero, then the FPB is still computing the result value. If the sign bit is one, then the read byte is the status byte of the result.

6. Read the N bytes of the result value. The next N reads of the DATAIN address after reading the status byte provide the N bytes of the result value. After the last result byte is received, at least 10 microseconds must elapse before another RESTART can be done. The result bytes must be read even after an error, or else the next operation will not perform correctly.

The following sample program demonstrates the efficient use of the FPB.

IMPORTANT NOTE: After power-on, a dummy operation must be performed to initialize the FPB. Subsequent operations will then perform correctly. The following program will initialize the FPB.

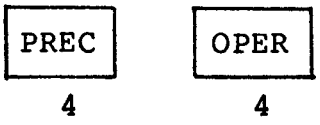
```
        LDA RESTART
        MVI A,2*16
        LDA 0
        LDA 0
        LDA 0
```

```
*SAMPLE USE OF THE NORTH STAR FPB
*FOR A DIVIDE OPERATION WITH 6 DIGIT PRECISION.
*IN THIS EXAMPLE ASSUME ARGUMENTS ARE IN MEMORY IN FORM:
*  MOST SIGNIFICANT BYTE (MSB) DIGIT PAIR
*  SUBSEQUENT DIGIT PAIRS FOLLOW THE MSB
*  EXPONENT+SIGN BYTE FOLLOWS LSB DIGIT PAIR.
*POINTERS ADDRESS THE EXPONENT+SIGN BYTE.
*BC HAS LEFT ARG POINTER.
*DE HAS RIGHT ARG POINTER.
*HL HAS RESULT POINTER.

*THE FPB RECEIVES ITS ARGUMENTS BY "PEEKING" AT THE 8080 BUS
*WHEN THE ARGUMENT VALUES ARE LOADED TO ACCUMULATOR.

  FDIV LDA RSTRT              "WAKE UP" FPB
     MVI A,6*16+DIVOP         SPECIFY PRECISION AND OPERATION CODE
     LDAX D                   EXPONENT+SIGN BYTE OF RIGHT ARG
     DCX D                    ADVANCE POINTER TO NEXT BYTE
     LDAX D                   LEAST SIGNIFICANT DIGITS OF RIGHT ARG
     DCX D                    ADVANCE POINTER TO NEXT BYTE
     LDAX D
     DCX D
     LDAX D                   MOST SIGNIFICANT DIGITS OF RIGHT ARG
     LDAX B                   EXPONENT+SIGN BYTE OF LEFT ARG
     DCX B
     LDAX B                   LEAST SIGNIFICANT DIGITS OF LEFT ARG
     DCX B
     LDAX B
     DCX B
     LDAX B                   MOST SIGNIFICANT DIGITS OF LEFT ARG
  *  NOW THE FLOATING POINT BOARD IS PERFORMING THE OPERATION
     LXI D,DATAIN             RECEIVE DATA ADDRESS FOR FPB
  FDIV1 LDAX D                WAIT LOOP FOR COMPLETION SIGNAL
     ORA A                    SIGN BIT "1" MEANS FPB IS DONE
     JP FDIV1                 LOOP IF SIGN BIT IS STILL "0"
     ANI EBITS                CHECK FOR ERROR, TESTED AT END
     LDAX D                   EXPONENT+SIGN OF RESULT
     MOV M,A                  STORE EXPONENT+SIGN OF RESULT
     DCX H                    ADVANCE POINTER
     LDAX D                   LEAST SIGNIFICANT DIGITS OF RESULT
     MOV M,A
     DCX H
     LDAX D
     MOV M,A
     DCX H
     LDAX D                   MOST SIGNIFICANT DIGITS OF RESULT
     MOV M,A                  STORE IT
     RZ                       RETURN IF NO ERROR WAS DETECTED
     JMP ERROR                GO REPORT ERROR
```

COMMAND BYTE FORMAT



PREC specifies the precision of the operand and result. It must be one of the following values:

Digits of Precision	PREC (hexadecimal)
2	2
4	4
6	6
8	8
10	A
12	C
14	E

OPER specifies the operation to be performed. It must be one of the following values:

Operation	OPER
Add	1
Subtract	2
Multiply	3
Divide	4

STATUS BYTE FORMAT

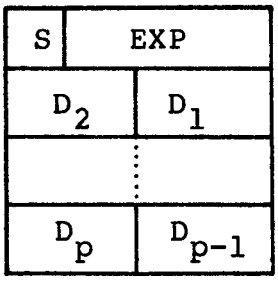
RD, if 1, indicates the operation is done and that the next N bytes read from the FPB will be the result. The other bits are valid only when RD is 1.

OF, if, 1 indicates an overflow error. That is, the magnitude of the result value would be too large to be represented.

UF, if 1, indicates an underflow error. That is, the magnitude of the result value would be too small to be represented.

DZ, if 1, indicates a divide by zero error.

FLOATING POINT VALUE FORMAT



P/2 bytes

The first byte of a floating point value passed to the FPB contains the sign bit(S) and exponent(EXP). S=0 indicates a positive value and S=1 indicates a negative value. The exponent is represented in excess 64 binary. The exponent represented is to the base ten. Some examples of exponent values follow.

Exponent	EXP(hexadecimal)
-63	01
-1	3F
0	40
1	41
10	4A
63	7F

The value zero is represented by a first byte with all zero bits (i.e. S=0 and EXP=0). If the precision is P digits, then the fraction part is represented by P/2 bytes following the first byte. Two binary encoded decimal digits are packed per byte. The decimal point is assumed to be to the left of the most significant digit. D_1 is the least significant digit and D_p is the most significant digit. All argument values must be normalized (i.e. D_p non-zero). The FPB will always return a normalized value.

Examples in six digit precision:

Value	Representation(hexadecimal)
0	00 00 00 00
1	41 00 00 10
$.123456 \times 10^{8}_{-9}$	48 56 34 12
$-.987654 \times 10^{-9}$	B7 54 76 98

FPB-A BUS INTERFACE DESCRIPTION

The FPB-A is compatible with the S-100 bus, i.e. the bus used by ALTAIR and IMSAI computers. The following signals on the bus are used by the FPB-A. All signals are positive, high true, TTL logic levels.

DI_7-DI_0	The 8 input data lines. The FPB reads operand bytes from these lines and tri-states result bytes onto these lines.
A_{15}-A_0	The 16 address lines. The FPB decodes these lines to recognize the RESTART and DATAIN addresses.
SMEMR	Status line that indicates to the FPB that a bus cycle is memory read.
SM1	Status line that indicates to the FPB whether a memory read is a first byte of an instruction fetch or a date byte fetch.
PDBIN	Timing signal used by FPB to strobe result bytes onto DI bus and to indicate that operand bytes have been strobed onto the DI bus by a memory module.
PSYNC	Used by the FPB to recognize the beginning of a memory cycle.
PRDY XRDY	The logical AND of these two signals indicates that read data is stable on the DI bus. The FPB uses these signals when reading operand bytes.
PHLDA	Indicates that the memory reference is a DMA cycle. The FPB ignores DMA cycles while reading operand bytes.

THEORY OF OPERATION

HARDWARE

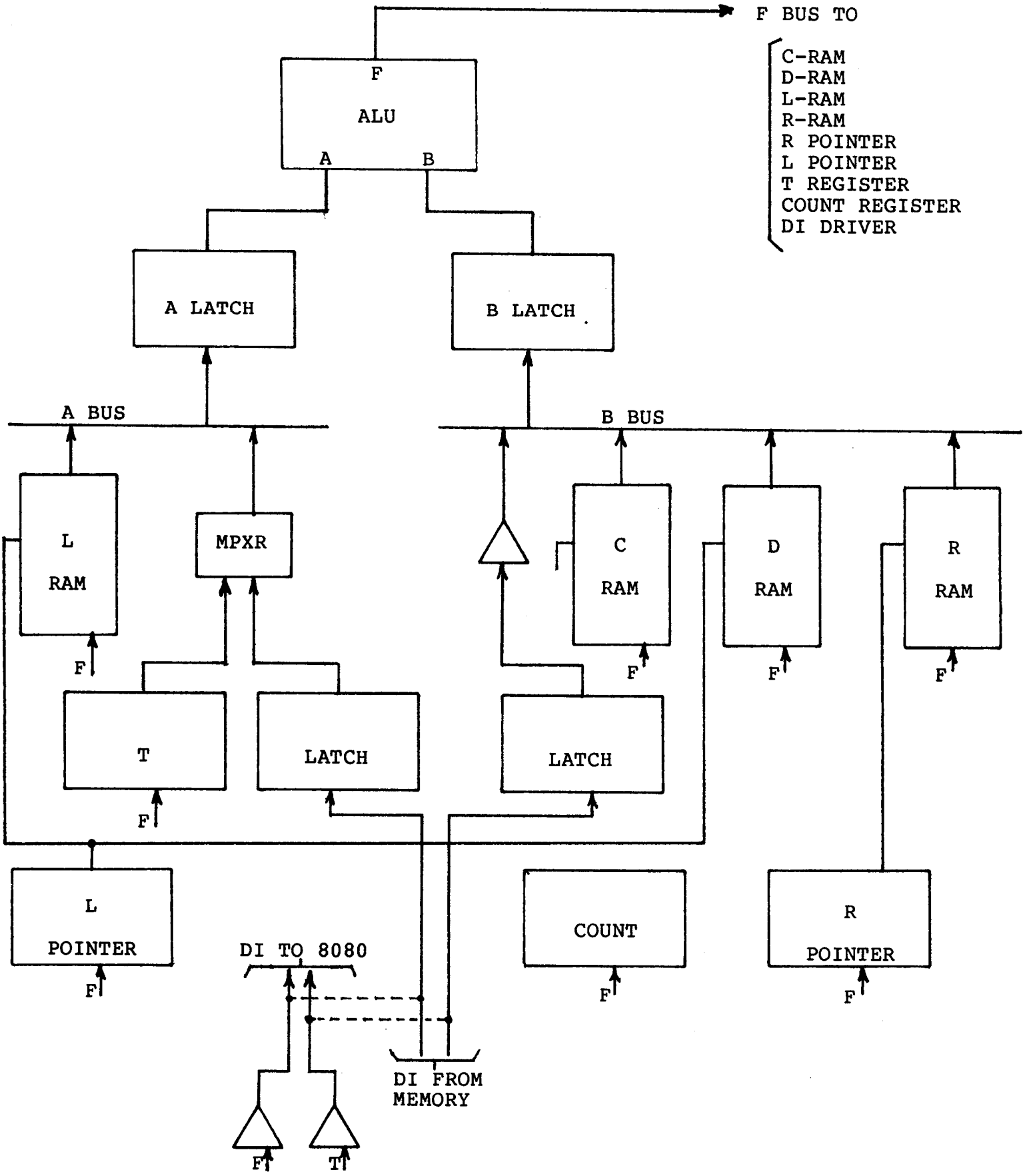
The FPB is a microprogram controlled processor, designed specifically to perform high speed decimal floating point arithmetic operations. The unit is implemented entirely from medium and small scale TTL integrated circuits and PROM memory. All data paths in the microprocessor are 4 bits wide, thus enabling processing one decimal digit at a time. The FPB is divided into four functional sections:

1. Data Paths

The following diagram shows the microprocessor data paths. There are three main buses in the data paths: A, B, and F. During each instruction cycle of the microprocessor, values are gated onto the A and B buses, which are latched at the end of the first half of the microcycle by the A and B latches. The latched A and B values serve as inputs to the ALU. The ALU output goes on the F-bus, which is clocked into one of the RAM's or counters at the end of the microcycle. The T-LATCH is always loaded from the F-bus. L-RAM and R-RAM generally hold the fraction part of the left and right floating point operands and the D-RAM holds the result. C-RAM is used to hold the exponents, various constants and temporary results required during the calculation. There are three counters. L-PTR and R-PTR are used to address the L-, R-, and D-RAM's. CNT is used to count the number of iterations through a program loop. The F-bus and T-LATCH together are used to gate an 8 bit result byte onto the DI bus. The two data input latches are used to catch operand bytes off the DI bus.

2. Clock Circuit

The clock circuit is a crystal controlled oscillator (IC's FN and FA). The generated MHz signal is used to generate three clock signals. LCLK/ is a free running 8 MHz signal. DST-CLK and WRT-T are timing signals that operate at 4 MHz. These signals do not occur while the microprocessor is waiting for a bus memory cycle.


F BUS TO
C-RAM
D-RAM
L-RAM
R-RAM
R POINTER
L POINTER
T REGISTER
COUNT REGISTER
DI DRIVER
F
ALU
A
B
A LATCH
B LATCH
A BUS
B BUS
L
RAM
MPXR
C
RAM
D
RAM
R
RAM
F
T
LATCH
LATCH
L
POINTER
COUNT
R
POINTER
DI TO 8080
DI FROM
MEMORY
T

3. Control Logic

The control logic is responsible for:

a. sequencing the micro program counter. At the end of every microcycle the p-counter is either loaded causing a branch, incremented to address the next instruction, or left unchanged during loops. The branch multiplexor (BMX) determines which of seven cnditions will cause a branch. The P-CON signal out of the condition PROM (CCOP) determines if the p-counter will increment.

b. interfacing and decoding bus control lines. The ARGS-AV signal indicates when an operand byte is on the DI bus and is used to gate the DI latches. The GO signal indicates that a new floating point computation is to begin. After being synchronized to the microprocessor clock, it is used to force the microprogram counter to zero. The BUSY RESPONSE and DONE RESPONSE signals indicate that a status or result byte should be gated onto the DI bus.

c. synchronizing the microprocessor with memory references from the CPU. The bus synchronizing logic (IC's GR, CH, AOI, SY, and DS) cause the microprocessor to hang if an expected memory read reference on the bus has not yet occurred. The SELA8 and RES8 microprogram conditions indicate that an argument byte is required by the microprogram or that the CPU should take a result byte, respectively. The SY flip-flops catch the memory conditions with the CPU clock. HSY is the or of these conditions synchronized to the microprocessor clock. The signal is then used to generate GATE. GATE is used to enable DST-CLK and WRT-T and to enable A-LATCH and B-LATCH. GATE is kept high by the synchronizing logic to hang the microprocesor during the first half of a microcycle.

4. Microprogram Storage

The microprogram is stored in ten bipolar PROM's organized 4 bits by 256 words. A hexadecimal listing of the PROM contents follows. The meaning of each PROM is now given.

a. Branch Address PROM (BAMP). Most significant 4 bits of the branch address.

b. Branch Address PROM (BALP). Least significant 4 bits of the branch address.

c. Branch Condition PROM (BCSP).
bit 3 indicates arithmetic should be done in decimal mode.
bits 2-0 indicate one of eight branch conditions.

d. Misc. PROM (MSCP).
   bit 3 enables clocking of the carry flip-flop at the end of a microcycle.
   bit 2 indicates that a result byte in the T-LATCH and on the F-bus is ready for the CPU.
   bit 1 indicates whether L=PTR and R-PTR should count up or down.
   bit 0 enables clocking of the three counters at the end of the microcycle.

e. Const PROM (CONP). Contains the 4 bit constant to gate onto the A-bus.

f. A Source PROM (ASRP). Specifies what to gate onto the A-bus.

g. ALU Control PROM (ACSP). Specifies the ALU function bits.

i. Store PROM (STOP).
   bit 3 is the ALU mode bit
   bits 2-0 specify which of seven destinations will get loaded from the F-bus at the end of the microcycle.

j. B Source PROM (BSRP). Specifies what to gate onto the B-bus.

k. C Address PROM (CADP). Specifies which word in the C-RAM to address.

MICROPROGRAM

The FPB microprogram contains 256 instructions. The function of the program is now described. The common start up sequence starts at address zero and is exectuted when a RESTART memory read is performed by the CPU.

Common Start Up Sequence

1. Receive the command byte and save the precision and operation code.

2. Receive the right operand and save in the R-RAM.

3. Receive the left operand and save in the L-RAM.

4. Dispatch to one of the four arithmetic routines based on the operation code.

ADD Routine

1. Test for zero args. If the left argument in zero, then return the right argument as the result. If the right argument is zero, then return the left argument as the result.

2. Scaling. If the exponent difference is greater than the precision, then the argument with larger exponent is returned as the result. Otherwise, the argument with the smaller exponent is scaled right by the exponent difference. For example:

$$.1234 \times 10^{3} + .5678 \times 10^{6}$$

after scaling becomes:

$$.00012 \times 10^{6} + .56780 \times 10^{6}$$

Notice that an extra digit of precision is maintained internally.

3. Perform arithmetic. If the sign of the arguments are the same, then the argument fraction parts are added. If the precision is P, then P+2 digits are added (one lower order rounding digit and one higher order digit in case of overflow). For example, if P=4, then the result of the previous example is 0.56792. If the sign of the arguments are different, then the fraction part of the right argument is

subtracted from the fraction part of the left argument. An extra low order digit is included in the subtraction. If the right argument fraction part is larger than the left argument fraction part, then the subtraction result must be complemented. For example:

.1234 - .5678 = .55560
and, after complementing, becomes:
.44440

4. Normalize result. If an ADD was performed, then the result is normalized right by one if the ADD caused an overflow. If a SUBTRACT was performed, then the result must be normalized left for each high order zero digit in the result. For example, the following result must be normalized by two digits:

$.5567 \times 10^6 - .5511 \times 10^6 = 0.00560 \times 10^6$
and, after normalizing, becomes
$.5600 \times 10^4$

If the result of the subtraction is all zero digits, then a zero result value is returned.

5. Rounding. If the extra low order digit is greater than or equal to five, then one is added to the fraction part of the result. If rounding causes an overflow, then the result must be normalized right by one digit. For example:

$.99999 \times 10^6$
must be rounded to
1.0000×10^6
and now must be normalized to
$.1000 \times 10^7$

6. Test and return result. If the operations generated a result with too large or small an exponent, then a zero result value is returned and the overflow or underflow error flag is returned in the status byte. Otherwise, the result is returned without an error flag in the status byte.

SUBTRACT Routine

1. The sign of the right argument is complemented and the the ADD routine is performed.

MULTIPLY Routine

1. Zero test. If either argument is zero, then a zero result is returned.

2. Perform multiply. The right argument is the multiplicand and the left argument is the multiplier. First, the result is set to zero. Then, the MULTIPLY is performed by doing a repeated ADD of the multiplicand for each digit of the multiplier. The ADD of the multiplicand is done to P+2 digit accuracy, thus maintaining a rounding and overflow digit. The multiplicand is added D times, where D is the value of the next digit of the multiplier. After each repeated ADD of the multiplicand, the result is shifted right. The multiplier is processed from least significant to most significant digit.

3. Normalize. If the result has a zero overflow digit, then the result must be normalized left by one digit.

4. Rounding. If the rounding digit is greater than or equal to five, then one is added to get the result exponent.

5. Compute exponent. The exponents of the two arguments are added to get the result exponent.

6. Test and return result. The result is tested and returned as in the ADD routine.

DIVIDE Routine

1. Test for zero. If the right arg is zero, then a zero value is returned and the divide by zero error flag is set in the returned status byte. Otherwise, if the left arg is zero, then a zero value is returned.

2. Perform divide. Division is done by performing repeated subtracts of the divisor from the dividend in a loop. The repeat loop is done once for each digit of the dividend. The dividend is processed from most significant to least significant digit. After each digit is processed, the dividend is shifted left. The repeated subtracts of the divisor continue until the dividend becomes negative, then the divisor is added back to make the dividend positive again. The successive digits of the result are the number of times the subtraction must be performed.

3. Normalize. If the first digit of the result is zero, then the result must be normalized left by one digit.

4. Round. If the rounding digit of the result is greater than or equal to five, then one is added to the result value fraction part.

5. Compute exponent. The result exponent is computed by subtracting the right arg exponent from the left arg exponent.

6. Test result and return. The result is tested and returned as in the ADD routine.

CHECK-OUT

For North Star systems, locate the factory master diskette, shipped with the system, or a closely related copy. Boot the DOS from this diskette, and type:

```
     GO FPBASIC
and  PRINT 2/3
```

If the FPB is not installed, or it is not addressed to match the BASIC, the result will be "NUMERIC OV ERROR". If the FPB performs the calculation correctly, the result should be ".66666667". This result also verifies that the standard FPBASIC calculates with eight digit precision.

If there is some doubt about whether a renamed copy of BASIC is an FPB version, one test is the above exercise with the FPB removed. Another way to test this, is by the speed of the calculations. The following program

```
10 FOR I=1 TO 1234
20 LET R=1/I
30 NEXT I
```

runs about 12-1/2 seconds in 8-digit BASIC, and about 4 seconds in 8-digit FPBASIC (on a North Star HORIZON).

To verify the address jumpering with an oscilloscope, use a "DS 4000" command of the M2D00 monitor program, to enter a two-instruction endless loop, and a "JP 4000" command to jump to it.

```
3A F2 DF   LDA RESTART
C3 00 40   JUMP BACK
```

Observe low-true GO pulses at MN, pin 8, or analyze the lack of them. For non-standard addressing, the "DF" in the program will be some other value.

Similarly, to check the DATAIN address, execute:

```
3A F1 DF   LDA DATAIN
C3 00 40   JUMP BACK
```

and observe low-true DATAIN pulses at MN, pin 6.

For serious debugging of the FPB, an example program is given on the last page of the schematics. Enter the program (and consider saving it on a file). Compare the example waveforms with those observed, and compare the answer stored in the last four bytes with the correct answer.

Simple diagnostic programs are easy to write in FPBASIC. Some of the following may be useful:

a. examples with pre-calculated resusts
b. division by one
c. multiplication vs repeated addition
d. trig identities of random numbers

Intermittant or complex failures will probably necessitate return of the FPB to the dealer or factory for repair. Be sure to include a description of the problem, preferably one which narrows it down to a simple example, and notes any correlation with external factors (e.g., heat, disk activity, etc.). In case of problems needing factory repair of the board, please consult the WARRANTY section of this manual.

APPENDIX A

FPB-A
PC BOARD ASSEMBLY
08008C BILL OF MATERIALS

ITEM	PART #	QTY	DESCRIPTION	DESIG
1	01001	15	Capacitor, .047uf, Ceramic Disk	
2	01006	1	Capacitor, 100uf, Electrolytic	C1
3	01012	2	Capacitor, 33pf, Dipped Mica	C5,C7
4	01013	1	Capacitor, 100pf, Dipped Mica	C6
5	01015	1	Capacitor, 330pf, Dipped Mica	C4
6	01022	2	Capacitor, 6.8uf, Dipped Tantalum	C2 C3
7	05001	1	PC Board, FPB-A, Unassembled	
8	13026	13	IC Socket, 14 pin AMP	
9	13028	33	IC Socket, 16 pin AMP	
10	13032	1	IC Socket, 24 pin AMP	
11	15002	1	Crystal, 8MHz	
12	38002	2	Lockwashers, #6 Internal Tooth	
13	38010	2	Hex Nuts, 6x32 AF MS plated	
14	38019	2	Screws, 6-32x1/2" BHMS	
15	38044	2	Heat Sink, 680-5220	VR1 VR2
16	43001	1	IC, 74LS00	AN
17	43002	1	IC, 74LS02	NR
18	43010	1	IC, 74LS27	3A
19	43011	1	IC, 74LS30	8N
20	43013	1	IC, 74LS42	SD
21	43014	1	IC, 74LS51	A01
22	43015	2	IC, 74LS74	CH,SY

APPENDIX A

ITEM	PART #		QTY	DESCRIPTION	DESIG
23	43016		5	IC, 74LS75	AL,BL, TL,DIR, DIL
24	43017		1	IC, 74LS109	DS
25	43019		2	IC, 74LS132	MN,ST
26	43020		1	IC, 74LS136	XR
27	43024		1	IC, 74LS151	BMX
28	43025		1	IC, 74LS157 - USE NATIONAL, TI, OR ITT, ONLY	EN
29	43027		2	IC, 74LS161	PCL,PCM
30	43029		3	IC, 74LS169	LP,RP, CNT
31	43032		1	IC, 74LS181	ALU
32	43036		1	IC, 74LS258A	AMX
33	43045		1	IC, 74LS00	FN
34	43054		1	IC, 74S08	FA
35	43054	*	4	IC, 74S189	LRAM, RRAM, CRAM, DRAM
36	43057		1	IC, 7405	OC
37	43061		1	IC, 74109	GR
38	43065		2	IC, 74367	RD,LD

* If TI74S189 is used, use IC's with date code of 7920 and newer.

APPENDIX A

ITEM	PART #	QTY	DESCRIPTION	DESIG
39	43083	1	FPB PROM, ACSP-3	ACSP
40	43084	1	FPB PROM, ASRP-3	ASRP
41	43085	1	FPB PROM, BALP-3	BALP
42	43086	1	FPB PROM, BAMP-3	BAMP
43	43087	1	FPB PROM, BCSP-3	BCSP
44	43088	1	FPB PROM, BSRP-3	BSRP
45	43089	1	FPB PROM, CADP-3A	CADP
46	43090	1	FPB PROM, CCOP-3	CCOP
47	43091	1	FPB PROM, CONP-3	CONP
48	43092	1	FPB PROM, MSCP-3	MSCP
49	43093	1	FPB PROM, STOP-3	STOP
50	61015	3	Resistor, 470 ohm, 1/4w, 5%	R1,R2, R5
51	61019	1	Resistor, 1.2k ohm, 1/4w, 5%	R4
52	61020	2	Resistor, 2.2k ohm, 1/4w, 5%	R3,R6
53	65002	2	Regulator, 7805	VR1, VR2
54	77081	1 in.	26 AWG Solid Wire with Green PVC Insulation	

APPENDIX B
FPB-A
PROM LISTING

```
CONST:  CONP-3 PROM
A010870F 0F08770F 0C0FEDB0 FFF5BEF4 F7FFFF00 000007F0 0FDFEFE0 AFF0EFFF
4FA00FFF 7B77B70F 0F0F0E00 EDFEF005 F7FF0F5F FE00F7FF 0FFF0FFF FFF0EF0F
F00FF000 0E001EFD 000DF00F F00000FF 7F0F7F0F F0000FFF D7FFFD0F 0FEFF000
FFFF00F0 EEFF0DF4 FA0000F0 00F7F0FF 00FFFF7F 00FFF0FF FFFFFFFF FFFFFFFF

CADDR:  CADP-3A PROM
E746514F FF320F1F 47FF777F FFFF8CB9 015623E6 3B52AAA4 BBFFF14F FFFC32FC
FFF4CF23 22FFFFA4 B6352BCB BFFF4EFF F04C4FF4 FC32FFFF 4FFFFF23 5647CFCF
4FECFF74 CFF77CC4 9BECF7C6 579E3223 210FF023 2C4FFF56 FF23C47C FCF4FEFF
FC4FFEF7 C7FFC7CF FF657810 32FF0232 C4FFFFF1 0565F4FF F0000000 00000000

STORE:  STOP-3 PROM
FFFFFF1D EEFFF8FE 18AA8889 EACFFFFF 0F888808 0F80F888 09DEB814 408777CB
04483888 88888888 08F8F0FF 1BAD9044 0F9B0449 87778888 1880CC88 888F3A7D
122B8478 438FF7B1 02230F7A A2207788 88F888F8 83188088 8888F0FB CB49D022
0B9D2287 7FB87FB0 44AA228F 77888F88 31880888 8F88D188 00000000 00000000

BSOURCE:  BSRP-3 PROM
66F6FF66 6F6FFF66 666F6666 66666666 66666666 66666666 6666666A C6C66666
CCC66C66 66666666 66666666 666666AC 66666CC6 C6666666 66666F66 66666666
6A66CC66 6C666666 66666666 66666666 66666666 66666666 66666666 66C666A6
6666A666 666C666C CC666666 66666666 66CC6666 666666AA 6EEEEEEE EEEEEEEE

ASOURCE:  ASRP-3 PROM
CBCBCCCC BCBCCCAC CCBCCCCC CCCCCCCC CCCCCCCC AACAACCC ACCCCCC0 CCCCCCCC
CCCCACCC CCCCCCCC ACACACAC CCCCCC0C CCCCCCCC CCCCCCCC CCCCBCCC CCCACCCC
C00CCCCC AC0CCCCC 000CCCCC C00C00CC CCACCCAC CCC00CCC CCCCCCAC CCCCCC00
CCCC00CC CCCCCCCC CCCC00CA 00CCCACC CCCCCCCC AACCCCCC CA888888 88888888

ALU: ACSP-3 PROM
FFBFBB9F FAFBBB6F 96FA666F FFFFFFFF F6AAAA96 9F69FBA6 9AFFFA99 99A999FA
99969AAA BBFFFF6A 9AFAF9F6 9FFFA969 96AA969A A999FFFF 9FF9FAAA AAAF9F9F
999AA996 99FFF9A9 99999F9A A99999AA BAFFFAEA A99FF9AA FFAAF9FA FA9AF969
9AAF99F9 9FFA9FA9 996699AF 99FFAEAA 99AA9FFA 6EAAF9AA 90000000 00000000

BRADDR MSBITS:  BAMP-3 PROM
00000000 00000001 171122A1 11111112 222A2E22 222224A3 A3333533 33333334
44444344 D44646E5 E5555555 55535565 66666666 36666667 77717776 76777788
88888778 88888889 99999999 999999A4 4AAAAAAA AAAAA1BB B6B6BBBB BBBBBBCB
BCCCCCCC BCCCCCCD DDDDDDDD D4DDDDDE EEEE1EEE EEEEEEEE 10000000 00000000

BRADDR LSBITS:  BALP-3 PROM
12345678 98BCDEF0 143221E8 9A8CDEF0 02436578 9ABCDE30 32335C79 7AFCDEF0
63145B8C ACBEDE50 52345678 9AA5DE0E 72345758 FAB8DEF0 1217548C ACBCED30
13146FC8 9ACEDEF0 33147638 9ABCDE0C 92345678 9ABCB722 1E4C5678 9ABCDE0E
92347479 7ABEDEF2 20345678 96BCDEF0 12327678 9ABCDEFE 70000000 00000000

MISC:  MSCP-3 PROM
BBBBBBBB 88BBBBBB BB88BBBB BB8BBBBB BBBBBBB3 B33BBBBB BBB8BBB8 0A83BBBB
8808B8BB BBBFBFBB BBBBBBBB BB8BBB80 BBBBB80A B3BABFBF B8CB88BB BBBBBBBB
B80BB32B 3BBBBBB3 880B8B38 A8AB8BBB BBBBFBBB F3B8CBBB BFBBBBBB BBBBB80
BBBB80B3 BBBBBBB8 808A8ABB 8BBFBBBF 3B8CBBFB BBBFBB8C BBBBBBBB BBBBBBBB

BRCOND:  BCSP-33 PROM
77777777 73777777 70730007 77377777 77101077 77777117 57737179 37077777
68777110 01777717 67777777 77377793 27777937 17737777 77377310 10777777
79370717 77077777 68773777 77777710 17777777 77773711 77107777 77777793
27779370 77707776 87777777 77777777 77737777 77777773 70000000 00000000

CONDITION:  CCOP-3 PROM
B7777777 77777777 95555555 55555555 95555555 55555555 95555555 55555555
B7777777 77777777 95555555 55555555 95555555 55555555 95555555 55555555
3333FFFF FFFFFFFF CCCCCCCC CC333333 66666666 66FFFFFF FFFFFF11 11111111
3333FFFF FFFFFFFF 88888888 88333333 AAAAAAAA AAFFFFFF FFFFFF11 11111111
```

North Star Computers, Inc.
14440 Catalina St., San Leandro, CA 94577 USA
(415) 357-8500 TWX/Telex (910) 366-7001

230708