The VLSI-PLM Board: Design, Construction, and Testing

Lau T. Nguyen, Linda G. Bushnell, and Vason P. Srini

Computer Science Division, EECS

University of California, Berkeley, CA 94720

ABSTRACT

We present the details of the design, simulation, construction, and testing of the VLSI-PLM Board. The VLSI-PLM Board is a wire-wrapped processor board for the VLSI-PLM Chip [STN88], a high performance CMOS processor for executing computer programs written in the Prolog language. All work was performed at the University of California at Berkeley. The design and simulations were performed using Mentor Graphics Computer Aided Design (CAD) tools on Apollo workstations. By using these tools, we were able to draw the gate-level design schematics on the computer and simulate the functionality and timing of the design. After the gate-level design passed all simulation tests, a wire-wrapped board was constructed with assistance from the Electronics Research Lab of the Electrical Engineering and Computer Science (EECS) Department. This wire-wrapped board was tested using a custom-made tester panel. The wire-wrapped board tests verified the computer simulations of the gate-level design. The total wire-wrapped part occupies 18 cm by 22 cm in a board 40 cm by 36 cm, with a total of 95 Integrated Circuit (IC) chips including the VLSI-PLM Chip.

Glossary of Terms

CAD - Computer Aided Design

CB - Xenologic Corporation's X-1 cache board

DAS - Digital Analysis System

EECS - Electrical Engineering and Computer Science

FAB - Force Address Bus

IC - Integrated Circuit

LSSD - Level Sensitive Scan Design

MG - Mentor Graphics Corporation

MAR - Memory Address Register in VLSI-PLM

MDB - Memory Data Bus

MDR - Memory Data Register in VLSI-PLM

MIR - Micro Instruction Register in VLSI-PLM (160 bits)

P Register - Program counter's register

PF1 - Prefetch 1 operation

PF2 - Prefetch 2 operation

PLM - Prolog Machine

PU - Prefetch Unit

SAM - Stand Alone Mode

STATUS - Status Register in VLSI-PLM (18 bits)

TTL - Transistor-Transistor Logic

VLSI-PLM - the VLSI-PLM Chip

VPB - the VLSI-PLM wire-wrapped board

Table of Contents

List of Tables

List of Figures

1. Introduction

In this paper we describe the design, simulation, construction, and testing of the VLSI-PLM Board. Both the design and simulations were accomplished using Mentor Graphics (MG) CAD tools on Apollo workstations. By using these tools, we were able to draw the gate-level schematics on the computer and simulate the functionality and timing of the design. After the design passed all simulation tests, a wire-wrapped version of the board was constructed with assistance from the Electronics Research Lab of EECS. A custom-made tester (called Software Interface) was used to test the wire-wrapped board, verifying the computer simulations of the gate-level design.

The main purpose of the wire-wrapped VLSI-PLM Board is to debug the VLSI-PLM Chip [STN88] and to interface the chip to the Xenologic Corporation's X-1 cache board. The chip is a high performance VLSI CMOS 32-bit microprocessor that executes Prolog programs with high regularity, low power dissipation, and a small instruction set. The chip has 120 pins and is 11 mm by 9.5 mm. The wire-wrapped VLSI-PLM Board measures 40 cm by 36 cm with 95 TTL parts (including the VLSI-PLM Chip) occupying an area of 18 cm by 22 cm. The board environment is designed to be used in a Sun system or as a stand-alone device, operating at a frequency range of 4 to 20 MHz.

Throughout this paper, we use VLSI-PLM for the VLSI-PLM Chip, VPB for the VLSI-PLM Board, and CB for the Xenologic's X1 cache board. Note that signals with .C suffixes are connected to CB, those with .B suffixes are connected to VPB, and those with .P suffixes are connected to VLSI-PLM. The signals are represented in upper case letters. The * notation for the signal names indicates an active-low signal (active when set low). There is an index of signals that indicates the first occurrence of each signal in the design. The signals (without their suffixes) are defined when they are first introduced.

In order to fully understand this paper, knowledge of the different signals used by VPB is necessary. Being both an interface board and a test board, VPB has to process signals from VLSI-PLM, CB, internally, and the Software Interface. We suggest that the reader first review the signals and behaviors of both VLSI-PLM and CB in [STN88]. Knowledge of the CB and VLSI-PLM signals and timing patterns are assumed throughout this report, though background information is often made available. The signals used in a design block of VPB are described as they relate to the block. It may be helpful to reference all the blocks that use the signals.

The VPB design is described in section 1. Simulation information for the gate-level VPB design is given in section 3. The testing procedures and results for the wire-wrapped VPB are presented in section 4. A conclusion is stated in section 5.

2. VLSI-PLM Board Design

VPB was designed using the Mentor Graphics IDEA Station schematic capture tools *Neted* and *Symed* [Men88] on Apollo workstations. Using these tools, we were able to create a multi-level hierarchical design. With Neted, we used standard parts, custom-made parts, and interconnecting wires (nets) to create a gate-level design schematic on the workstation. Symed was used to draw the custom parts' symbols and any component symbols that were placed on the Neted schematic sheets. Please see [BuS89] for a complete discussion on using the MG tools.

In the following sections on the VPB design blocks, we describe their inputs, outputs, and purpose. Some of the outputs that appear in the figures are not described because they are either for future testing of VPB with a Digital Analysis System (DAS), or for some other future display purpose. Please consult the index of signals to locate where the signals are first presented and defined. In the functional descriptions of the blocks, we only mention the larger TTL parts used to create the necessary logic. We use a double vertical bar in the tables for the functional descriptions of the VPB blocks to separate the inputs from the outputs. All figures and simulation results are in the appendix.

2.1. VLSI-PLM Board

VPB is a wire-wrapped board designed for debugging VLSI-PLM [STN88] and connecting VLSI-PLM to the cache board of Xenologic Corporation's X-1 system. The debugging of VLSI-PLM is performed by placing it on the wire-wrapped VPB and using the custom-made tester. To interface VPB to the cache board, the X-1 system is connected to the VME bus of a Sun 3/260 computer so that the Sun can be used as a host for VPB. Prolog programs are then compiled and loaded into the memory of the Sun and a start signal is sent to VPB. VPB processes the Prolog program with memory management and input and output (I/O) provided by the Sun. In this way, the Sun views VPB as a Unix device.

In this section, we describe the top-level of VPB's hierarchical design. This top-level is broken down into eleven blocks: Clock Generator, Pushbuttons, Toggle, Atomicb, MIRsignals, MARbuff,

Passmclk, Opcodebus, Forceaddr, Xcver, and Memdatbus. The gate-level design was made by creating a symbol for the entire VPB using Symed. This symbol (shown in Figure 1) shows the inputs and outputs of the entire VPB. On the lower right hand corner of the symbol are the signals (labeled DAS--) used for future testing with a DAS. The top-level schematic sheet (shown in Figure 2), created using Neted, contains twelve symbols that are interconnected by wires. The signals connected to the input and output port indicators correspond to the inputs and outputs of the entire VPB; the signals connected to non-port wires are internal. Eleven of these symbols are the design blocks of VPB, and the remaining symbol is a standard 74F244 octal buffer/line driver. Under each one of the block symbols lies a schematic for that block, thereby creating a hierarchical design. When all of the design schematics were complete, they were checked for errors, corrected, and prepared for simulation.

Inputs

Table 1 VPB Inputs from CB

Input	Description
CCLK.C	internal CB clock
CMCLK.C	CB output clock
CONTINUE.C	Continue signal
FORCEADDR.C(8:0)	Force Address Bus
FORCEBR.C	Force Branch signal
IBUFEMPTY.C	CB ibufempty signal
MEMDATBUS.C(31:0)	Memory Data Bus
OPCODE.C(7:0)	Opcode Bus
RST*.C	cold or hard reset signal
SCY.C	Single Cycle signal
SINST.C	Single Instruction signal

As summarized in Table 1 above, CCLK is the internal CB clock and is always running; CMCLK is the CB output clock. When CB asserts CONTINUE, this tells the MCLK/MCLK* clock to start up again after it was stopped by one of SCY or SINST. FORCEADDR(8:0) is a 9-bit bus to transfer the forced microbranch address to VLSI-PLM when FORCEBR is asserted and to output the contents of the ROM latch when OUTROMADDR is asserted. FORCEBR is a one cycle long control signal from CB that informs VLSI-PLM to do a forced microbranch to the address on FORCEADDR(8:0). IBUFEMPTY is asserted when CB does not have the next opcode in response to INSTREN* (signal issued by VLSI-PLM).

The input MEMDATBUS(31:0) is the primary data path to memory. The reading and writing of data to and from the Memory Data Register (MDR) passes on this bi-directional bus, as well as instruction arguments during instruction prefetch. OPCODE(7:0) is the path for the 8-bit opcode from the prefetch buffer to the instruction register used during instruction prefetch by VLSI-PLM. The prefetch buffer is internal to CB. When RST* is asserted by CB, VLSI-PLM resets itself, i.e., it branches to the boot00 microstate (one of the 512 states) of the ROM. This is a hard reset. SCY and SINST are the VLSI-PLM Single Cycle and Single Instruction signals, respectively. When CB asserts SCY in cycle t (any cycle), MCLK is stopped after cycle t+1. When CB asserts SINST, CB wants VLSI-PLM to stop after the current micro-instruction sequence. One single instruction is equal to a sequence of micro-instructions.

Table 2 VPB Inputs from VLSI-PLM

Input	Description
DSPACE.P	msb for memory address
EXCEPT.P	indicates exception on VLSI-PLM
EXTERNALFU*.P	indicates function ready for execution
FAIL*.P	tells PU that failure occurred
INSTREN*.P	requests transfer of data from prefetch buffer
LASTMI*.P	indicates end of macro instruction execution
MARBUS.P(27:0)	memory address
MEMDATBUS.P(31:0)	Memory Data Bus
MEMREAD*.P	requests a memory read
MEMWRITE*.P	requests a memory write
NEWP1*.P	tells PU that MEMDATBUS(31:0) holds 32 bits
NEWP2*.P	tells PU that MEMDATBUS(31:0) holds 8 bits
SHIFTOUT1.P	data from MIR scan path
SHIFTOUT2.P	data from STATUS scan path
WAIT.P	indicated VLSI-PLM halted

The input DSPACE is the most significant address bit (msb) for memory access. EXCEPT is a one cycle long status signal indicating that an exception has occurred on VLSI-PLM. EXTERNALFU* is a one cycle long control output to the cache board indicating that a built-in function is to be executed by an external functional unit. FAIL* is a one cycle long control signal to tell the Prefetch Unit (PU) that a failure has occurred and that the prefetch buffer is to be flushed. INSTREN* is a one cycle long control signal to request a transfer of data from the prefetch buffer. LASTMI* is a one cycle long control signal indicating that the last micro-instruction of a Prolog Machine (PLM) instruction is in execution (end of

macro-instruction execution). If EXTERNALFU* and LASTMI* are asserted at the same time, then the transfer of all data to the cache board for the execution of the external built-in function has been completed.

The input MARBUS(27:0) is a 28-bit memory address. MEMREAD* and MEMWRITE* are one cycle long control signals that request a memory read, and write, respectively. NEWP1* is a one cycle long control signal to tell the PU that MEMDATBUS(31:0) holds a 32-bit value to reload the program counter's register (called the P register). NEWP2* is a one cycle long control signal to tell the PU that MEMDATBUS(31:0) holds an 8-bit value to be added to the P register. SHIFTOUT1 is the data from the MIR scan path controlled by the signal TEST1. SHIFTOUT2 is the data from the STATUS scan path controlled by the signal TEST2. WAIT is a one cycle long control signal to the cache board indicating that VLSI-PLM is halted (looping on a micro-instruction).

Table 3 VPB Inputs from VPB Software Interface

Input	Description	Function
CMCLK.B	internal source clock	
KPFA(8:0)	keypad input to FAB	used in testing
KPMD(31:0)	keypad input to MDB	used in testing
POR*	power-on-reset signal	used during power-up
S1	RST*	cold or hard reset signal
S2	FORCEBR	Force Branch signal
S3	SAM (or Debug)	allows for SAM testing of VLSI-PLM
S4	ENBREAK	Enable Break signal
S5	OUTMEMDAT	VLSI-PLM pin
S6	ENTER	data on KPMD entered to MEMDATBUS.P(31:0)
S7	TEST1	allows for testing of MIR
S8	TEST2	allows for testing of STATUS
S9	TESTIN	allows for shifting into MIR or STATUS
S10	SINST	Single Instruction signal
S11	CONTINUE	Continue signal
S12	SCY	Single Cycle signal
S13	Continue Shiftout	continues or starts shift out process
S14	FBRCSY	used in TestB
S15	FBRSI	not currently used
S16	SCYCONT	not currently used
S17	SINCONT	not currently used
SHIFTA.B	internal source clock	clock for shifting data to LSSD

The Software Interface is the custom-made tester used to test and debug VPB. It generates the switches S1 through S17 in order to simulate the cache board. In the Cache Mode, all of the switches, except S3, are not generated by the Software Interface. They are only active in the SAM. Note that in Table 3 FAB is the Force Address Bus, MDB is the Memory Data Bus, MIR is the Micro Instruction Register (160 bits wide), and STATUS is the Status Register (18 bits wide).

KPFA(8:0) holds the value of FORCEADDR(8:0) that is entered by the user, and is sent to the Force Address Bus when S2 (FORCEBR) is asserted. KPMD(31:0) holds the value of MEMDATBUS(31:0) that is entered by the user, and is sent to the Memory Data Bus when the S6 (ENTER) is asserted. POR* resets all of the registers during the power-up of VPB.

S1, S2, S10, S11, and S12 are equivalent to RST*, FORCEBR, SINST, CONTINUE, and SCY in CB. S3 is used to control the switching between the SAM and Cache Mode. If S4 is asserted when VLSI-PLM is forcing out a ROM address, the clock MCLK stops, causing the processor to stop. This is

used to debug VLSI-PLM. S5 is a direct connection to the OUTMEMDAT pin of VLSI-PLM. S6 allows for data on KPMD(31:0) buffer to be entered to MEMDATBUS(31:0) when the enter key is pressed on the tester.

SHIFTA is the clock for shifting data into the master register of Level Sensitive Scan Design (LSSD). S7 and S8 are equal to the TEST1 pin and TEST2 pin of VLSI-PLM, respectively. When S7 is asserted, TESTIN = 0, and SHIFTA is given to VLSI-PLM, then the contents of MIR (160 bits) are shifted out to the SHIFTOUT1 pin of VLSI-PLM according to the frequency of SHIFTA (at 1 bit per cycle). If we have the same situation, but TESTIN=1, then data on the SHIFTIN1 pin of VLSI-PLM are shifted into MIR. Similarly, when S8 is asserted, TESTIN = 0, and SHIFTA is given to VLSI-PLM, then the contents of STATUS (18 bits) are shifted out to the SHIFTOUT2 pin of VLSI-PLM with the frequency of SHIFTA. Also if we have the same situation, but TESTIN = 1, then data on the SHIFTIN2 pin of VLSI-PLM are shifted into STATUS. S9 (TESTIN) controls the direction of these shifts. When S9 = 1, the contents of the MIR or STATUS are shifted in, and when S9 = 0 then are shifted out. In the TEST1 out mode (TESTIN = 0), 160 bits (5 words) are shifted out of MIR. These 5 words have to be displayed on the tester. VPB gives VLSI-PLM 32 clock cycles to shift out one word of data. After that time, VLSI-PLM stops shifting out data so that the Software Interface is able to display the word. When the Software Interface has read in the 32 bits to be displayed, it asserts S13 (CONTSHOUT, i.e., Continue Shift Out) to allow for the next 32 bits to come in.

S14 (FBRCSY) is CONTINUE asserted, followed by FORCEBR and SCY asserted on the next clock cycle. It allows for two cycles of MCLK to pass when MCLK has already stopped. The two cycles are one to Force Branch, and the other to move the ROM contents to MIR. It allows VPB, when in SAM, to stop VLSI-PLM at a desired location in the ROM. It is used in the TestB test for the wire-wrapped board. S15 (FBRSI) is CONTINUE asserted, followed by FORCEBR and SINST. It allows for MCLK to pass until LASTMI* is asserted. S16 (SCYCONT) is CONTINUE and SCY asserted in the same cycle. It allows for Passmclk to pass one cycle of MCLK at a time. S17 (SINCONT) is CONTINUE and SINST asserted in the same cycle. It allows one instruction to be executed at a time. S15, S16, and S17 are not currently used by Software Interface.

Outputs

Table 4 VPB Outputs to CB

Output	Description
ATOMIC*.C	indivisible read-write operations
DSPACE.C	msb for memory address
EXCEPT.C	indicates exception on VLSI-PLM
EXTERNALFU*.C	indicates external function ready for execution
FAIL*.C	tells PU that failure occurred
INSTREN*.C	requests transfer of data from prefetch buffer
LASTMI*.C	indicates end of macro instruction execution
MARBUS.C(27:0)	memory address
MEMREAD*.C	requests a memory read
MEMWRITE*.C	requests a memory write
NEWP1*.C	tells PU that MEMDATBUS(31:0) holds 32 bits
NEWP2*.C	tells PU that MEMDATBUS(31:0) holds 8 bits
WAIT*.C	indicated VLSI-PLM halted

The VPB output ATOMIC* to CB is a signal used by CB for indivisible memory read and write operations.

Table 5 VPB Outputs to VLSI-PLM

Output	Description
FORCEADDR.P(8:0)	Force Address Bus
FORCEBR.P	Force Branch signal
MCLK	master clock
MCLK*	master clock
MEMDATBUS.P(31:0)	Memory Data Bus
OPCODE.P(7:0)	Opcode Bus
OUTMEMDAT.P	indicates MEMDATBUS(31:0) in output mode
OUTROMADDR.P	requests ROM latch contents for FORCEADDR(8:0)
RESET*.P	resets VLSI-PLM
RLMDR*.P	reloads MDR
SHIFTA	clock for shifting data to LSSD
SHIFTIN1.P	data for first scan path
SHIFTIN2.P	data for second scan path
TEST1.P	indicates scan of MIR
TEST2.P	indicates scan of STATUS

Table 5 above shows the outputs from VPB to VLSI-PLM. MCLK and MCLK* are the master 100 ns clock for VLSI-PLM. OUTMEMDAT is a control signal from CB indicating that the 32 pads of MEMDATBUS(31:0) should be in the output mode. OUTROMADDR is a one cycle long control input

to VLSI-PLM requesting the contents of the ROM latch to be output on FORCEADDR(8:0). RESET* is an arbitrarily long but synchronized control signal to VLSI-PLM for reseting. RLMDR* is a one cycle long control signal to reload the MDR register of VLSI-PLM once data are available for a memory read after a cache miss. It is RESET* delayed by one cycle in order to be synchronized to the initialized value of MDR. SHIFTIN1 and SHIFTIN2 are data for the first, respectively second, scan path controlled by TEST1, respectively TEST2. The output TEST1 is a one cycle long control signal to initiate the scan of MIR as a part of testing VLSI-PLM. TEST2 is a one cycle long control signal to initiate the scan of status bits in STATUS of VLSI-PLM.

Functional Description

VPB has two modes of operation: the Cache Mode and Stand-Alone Mode (SAM). In the appendix of [STN88], the timing diagrams for VLSI-PLM are given, explaining these modes in detail. The function of VPB in each mode is also discussed in the sections on the VPB blocks. We often refer to VPB in different test modes. These modes (Test Mode, Test1 Mode, Testin Mode, etc.) indicate the state of VPB during the tests.

In the Cache Mode (S3 = 0), VPB is connected to CB and functions as an interface between CB and VLSI-PLM. The system is a complete processor for Prolog. Some of its functions in this mode are to (1) provide drive to the CB inputs, (2) tri-state the CB inputs, (3) stop and resume MCLK and MCLK* at appropriate times, (4) deal with empty instruction buffers (IBUFEMPTY), and (5) latch the VLSI-PLM outputs for inputs to CB.

In SAM (S3 = 1), VPB is not connected to CB and functions as a test board for VLSI-PLM. Some of its functions in this mode are to (1) shift data into or out of MIR or STATUS, (2) execute Single Cycles, (3) execute Single Instructions, (4) Force Branch to desired ROM location, (5) provide data to MDR, and (6) stop MCLK and MCLK* at appropriate times.

2.2. Clock Generator

The Clock Generator is the source of all clock signals for VPB and VLSI-PLM. It produces the MCLK, MCLK*, SHIFTA, CMCLK, and CMCLK* clock pulses. The logic required to stop, start, and resume these clock signals is implemented in Clock Generator. The symbol and schematic sheet for the

Clock Generator are shown in Figures 3 and 4, respectively.

Inputs

Table 6 Inputs to Clock Generator

Input	Description
CMCLK.C	CB clock output
CMCLK.B	VPB clock output
PASSMCLK	from Passmclk to pass or stop MCLK
POR*	power-on-reset pulse to set initial conditions
R3*	from Pushbuttons to indicate SAM
R10	from Pushbuttons to indicate Test Mode (Test1 or Test2)
R20	from Pushbuttons to indicate Test Mode entered
RESUME*	from Memdatbus to resume SHIFTA.P
SHIFTA.B	VPB clock used for SHIFTA.P
STOPSHIFTA*	from Memdatbus to inhibit SHIFTA.P

PASSMCLK is sent from Passmclk to stop MCLK (PASSMCLK = 0) or to allow MCLK to keep going (PASSMCLK = 1). Note that the R signals equal the S switches after the switches pass through a gate. R3* is used to indicate SAM. R10 is asserted when the Test1 (R10 = R8) or Test2 (R10 = R9) test is selected on the tester by the user. It stays on until the Test Mode is finished. R20 equals R10, but only for the first clock pulse. R20 is used by Controller (in Memdatbus) to stop SHIFTA in order to allow the Software Interface enough time to get ready for displaying a word. When the Software Interface is ready for a word, it issues CONTSHOUT*. RESUME* is then asserted to allow SHIFTA to start up again. When shifting out the contents of MIR to MEMDATBUS(31:0), STOPSHIFTA* is asserted every 32 bits to allow the Software Interface time to display a word from MEMDATBUS(31:0).

Outputs

Table 7 Outputs from Clock Generator

Output	Description
CMCLK	used in VPB
CMCLK*	not used
MCLK	to VLSI-PLM
MCLK*	to VLSI-PLM
SHIFTA	to VLSI-PLM

Functional Description

Table 8 Functional Description of Clock Generator

R10	SHIFTA
0	0
1	SHIFTA.B

Table 8 above shows the usage of R10. When the Test Mode is initially entered, R20 is asserted, inhibiting SHIFTA. When the Software Interface is ready to receive the MEMDATBUS(31:0) data, RESUME* is asserted and SHIFTA is allowed to start. After a word has been shifted into the Software Interface, STOPSHIFTA* is asserted, inhibiting SHIFTA. This allows the Software Interface enough time to display data from MEMDATBUS(31:0). When the Software Interface is finished displaying, RESUME* is asserted and SHIFTA is resumed. VLSI-PLM will shift out the next word to MEMDATBUS(31:0). Figure 4 shows the design of Clock Generator. Four 74F74 dual D flip-flops and three 74F241 octal buffer/line drivers with tri-state output were used to implement the necessary logic.

Table 9 Functional Description of Clock Generator (continued)

R3*	R10	PASSMCLK	CMCLK, CMCLK*	MCLK	MCLK*
0	0	0	CMCLK.B	0	1
0	0	1	CMCLK.B	CMCLK.B	CMCLK.B
0	1	X	CMCLK.B	0	1
1	0	0	CMCLK.C	0	1
1	0	1	CMCLK.C	CMCLK.C	CMCLK.C
1	1	X	forbidden input condition		

The signals R3*, R10, and PASSMCLK are decoded together. CMCLK and MCLK are chosen in parallel to reduce the delay between CMCLK and MCLK. R10 and R3* are synchronized correctly to insure that the clocks begin and stop at the proper time, i.e., complete clock pulses, not spikes.

2.3. Pushbuttons

Pushbuttons, along with Toggle, enables VPB to interact with the user in SAM. All data and control signals from the user and CB are arbitrated here. The appropriate source is directed to VLSI-PLM. Figures 5 and 6 show the symbol and design schematic for Pushbuttons, respectively.

Inputs

There is only one non-switch input to Pushbuttons. The input SHIFTA is the clock output from Clock Generator and is used to pulse R20. Table 10 shows the inputs for Pushbuttons. L denotes logic low, H denotes logic high, and X denotes don't care.

Table 10 Inputs to Pushbuttons

Input	Description	Cache Mode	SAM	Initially
S3	SAM	L	H	H
S4	ENBREAK	L	X	L
S5	OUTMEMDAT.P	X	X	H
S7	TEST1.P	L	X	L
S8	TEST2.P	L	X	L
S9	TESTIN	L	X	L
SHIFTA	pulses R20			

Outputs

Table 11 Outputs from Pushbuttons

Output	Description
ENBREAK	to allow break point feature in SAM
OUTMEMDAT.P	to VLSI-PLM
R3	indicates SAM
R3*	R3 inverted
R5	same as OUTMEMDAT.P
R8	indicates Test1 Mode
R9	indicates Test2 Mode
R10	indicates Test Mode
R20	indicates Test Mode
S9.O	indicates Testin Mode
TEST1.P	indicates Test1 Mode
TEST2.P	indicates Test2 Mode

R5 is connected directly to the OUTMEMDAT pin of VLSI-PLM, and is not currently used. R8 indicates the Test1 Mode, and is connected to the TEST1 pin of VLSI-PLM. Similarly, R9 indicates the Test2 Mode, and is connected to the TEST2 pin of VLSI-PLM. S9.O is equivalent to S9.

Functional Description

Pushbuttons takes the six input switches S3, S4, S5, S7, S8, and S9 along with the clock SHIFTA and produces the necessary signals for the Software Interface. The logic is created by using a 74F244 and

a 74F74 as shown in Figure 6.

2.4. Toggle

Toggle works with Pushbuttons to enable VPB to interact with the user in SAM. The symbol and design schematic are shown in Figures 7 and 8, respectively. The inputs to Toggle are shown in Tables 12 and 13 below.

Inputs

Table 12 Inputs to Toggle

Input	Description
CMCLK	from Clock Generator
CONTINUE	from CB
FORCEBR.C	from CB
POR*	power-on-reset signal from VPB
R3, R3*	SAM indicators from Pushbuttons
RST*.C	reset signal from CB
SCY	from CB
SINST	from CB
SHIFTA	from Clock Generator

Table 13 Switch Inputs for Toggle

Switch	Description	Cache Mode	SAM	Initially
S1	RST*	H	X	H
S2	FORCEBR	L	X	L
S6	ENTER	H	X	H
S10	SINST	L	X	L
S11	CONTINUE	L	X	L
S12	SCY	L	X	L
S13	CONTSHOUT*	H	X	H
S14	FBRCSY	L	X	L
S15	FBRSI	L	X	L
S16	SCYCONT	L	X	L
S17	SINCONT	L	X	L

Outputs

Table 14 below shows the outputs of Toggle. R6* is a pulse of width equal to the period of CMCLK. It is used in the Testin Mode and when MEMREAD* is issued in SAM to tell Memdatbus that data has been entered on MEMDATBUS(31:0). R21* is a pulse of width equal to the period of SHIFTA. It is used in the Testout Mode to tell Memdatbus to continue or start receiving data.

Table 14 Outputs from Toggle

Output	Where to	Description
FADDROE*	Forceaddr	used to tristate FORCEADDR.C(8:0) in SAM
FADDRTR*	Forceaddr	direction of data flow in FORCEADDR.C(8:0)
FORCEBR.P	VLSI-PLM	Force Branch signal
KPOE*	Forceaddr	allows data from KPFA(8:0) to be entered
OUTROMADDR.P	VLSI-PLM	complement of FORCEBR.P
R6*	Memdatbus	data entered onto MEMDATBUS(31:0)
R11	Passmclk	indicates Single Cycle
R12	Passmclk	indicates Single Instruction
R13	Passmclk	indicates Continue
R21*	Memdatbus	ready to continue or start receiving data
RESET*.P	VLSI-PLM	resets VLSI-PLM
RLMDR*	VLSI-PLM, Memdatbus	RESET*.P delayed by one cycle

Functional Description

R11, R12, R13, RESET*, RLMDR*, FORCEBR, and OUTROMADDR are all pulses of one CMCLK long. We used two 74F273 octal D flip-flops, two 74F244s, and four 74F74s to create the necessary logic for Toggle (see Figure 8). Recall that R3 = 0 indicates Cache Mode, and R3 = 1 indicates SAM.

Table 15 Functional Description of Toggle

R3	α	Comments
0	from CB	in Cache Mode
1	from VPB	in SAM
R3	FADDROE*	
0	0	let FADDROE* go to CB
1	1	Tri-state FORCEADDR.C(8:0)
FORCEBR.C	FADDRTR*	
0	1	Transmit data from VLSI-PLM to CB when OUTROMADDR.P asserted.
1	0	Supply data to VLSI-PLM from CB when FORCEBR.C asserted.

α= R11, R12, R13, RESET*.P, RLMDR*, FORCEBR.P, OUTROMADDR.P

Table 16 Functional Description of Toggle (continued)

R3	FORCEBR.P	KPOE*	Comments
0	X	1	Deactivate keypad input to FORCEADDR.P(8:0) for Cache Mode.
1	0	1	Only Allow data from keypad to FORCEADDR.P(8:0) on Force Branch in SAM.
1	1	0	Only Allow data from keypad to FORCEADDR.P(8:0) on Force Branch in SAM.
β	R6*	R13	
0	0	1	if none of β asserted, R13 = R6
0	1	0	if none of β asserted, R13 = R6
1	X	1	if any of β asserted, R13 = 1

β= S11,S14,S15,S16,S17

2.5. Atomicb

To support multiprocessing, VLSI-PLM has the instructions lock and unlock. The lock instruction performs a memory read as an indivisible operation. To support this operation, the control signals WAIT and EXTERNALFU* are provided. If both these signals and MEMREAD* are asserted, then an atomic read operation is performed. The unlock instruction performs a memory write as an indivisible operation. If WAIT, EXTERNALFU*, and MEMWRITE* are asserted, then an atomic write operation is performed. Atomicb (a.k.a. Atomic*) allows for these functions to be done. Figures 9 and 10 show the symbol and schematic, respectively. Notice the simplicity of the logic in the schematic required to perform this function.

Inputs

Table 17 Inputs to Atomicb

Input	From where
EXTERNALFU*.P	VLSI-PLM
WAIT.P	VLSI-PLM

Outputs

ATOMIC* is the only output of Atomicb; it is sent to CB.

2.6. MIRsignals

MIRsignals contains the control signals from VLSI-PLM to CB. Recall that MIR denotes Micro Instruction Registers. The symbol and schematic are shown in Figures 11 and 12, respectively.

Inputs

Table 18 Inputs to MIRsignals

Input	From where
EXTERNALFU*.P	VLSI-PLM
FAIL*.P	VLSI-PLM
LASTMI*.P	VLSI-PLM
MEMREAD*.P	VLSI-PLM
MEMWRITE*.P	VLSI-PLM
MCLK*	Clock Generator
NEWP1*.P	VLSI-PLM
NEWP2*.P	VLSI-PLM
PASSMCLK	Passmclk
R3*	Pushbuttons
WAIT.P	VLSI-PLM

Outputs

Table 19 Outputs from MIRsignals

Output	To where
EXTERNALFU*.C	CB
FAIL*.C	CB
LASTMI*.C	CB
MEMREAD*.C	CB
MEMWRITE*.C	CB
NEWP1*.C	CB
NEWP2*.C	CB
WAIT*.C	CB

Functional Description

For the functionality of MIRsignals in the Cache Mode, the signals MEMREAD*, MEMWRITE*, FAIL*, NEWP1*, and NEWP2* must be available to CB no later than 20 ns after the rising edge of CMCLK. They are available to VPB at the falling edge of the MCLK*. The total delay from CMCLK to MCLK* is 7.5 ns. Thus the total delay from the inputs to the output must be less than 12.5 ns. If WAIT is asserted, PASSMCLK is asserted, or in SAM (R3* asserted), then the outputs to CB are tri-stated. The inverter in the path of MCLK* serves two purposes: to assure that 74F374 is positively edge triggered, and to give time for VLSI-PLM signals to stabilize after the falling edge of MCLK*. The signals LASTMI*, EXTERNALFU*, and WAIT* are included for future use. We used a 74F374 octal D flip-flop with tri-state outputs, as seen in Figure 12, to create the necessary logic for MIRsignals. The reason for choosing a F374 was that we needed all of the input signals to be asserted for the whole clock pulse. We also had to make sure certain clock delay times were observed (see timing diagrams in [STN88]).

2.7. MARbuff

MARbuff provides the necessary drive for sending the contents of the Memory Address Register to CB. The symbol and schematic are shown in Figures 13 and 14, respectively.

Inputs

Table 20 Inputs to MARbuff

Input	From where
MARBUS.P(27:0)	VLSI-PLM
R3	Pushbuttons

Outputs

MARBUS(27:0) is the only output from MARbuff; it is sent to CB.

Functional Description

CB requires signals with high drives. Therefore it is critical that the total delay of sending MARBUS(27:0) from VLSI-PLM to CB be less than 5 ns. To guarantee this, we implement MARbuff (see Figure 14) using a 74F244 which has a propagation delay of less than 5 ns.

Table 21 Functional Description of MARbuff

R3	MARBUS.C(27:0)
0	MARBUS.P(27:0)
1	Tri-state

2.8. Passmclk

The master clock to VLSI-PLM (MCLK and MCLK*) is stopped and resumed at certain times, for example, during Single Cycles and Single Instruction operations. Passmclk implements the logic necessary for this to occur. Figures 15 and 16 show the symbol and schematic for Passmclk, respectively.

Inputs

Table 22 Inputs to Passmclk

Input	From where
BREAK	Forceaddr
CMCLK	Clock Generator
ENBREAK	Pushbuttons
MEMREAD*.P	VLSI-PLM
LASTMI*	VLSI-PLM
POR*	VPB
R3	Pushbuttons
R11	Toggle
R12	Toggle
R13	Toggle

Outputs

There are two outputs from Passmclk. The output PASSMCLK is used to stop MCLK and MCLK*. The output CACHEMISS may be ignored; it is only for future use.

Functional Description

Table 23 State Table for Passmclk

Current State	R11	R12	R13	α	Next State	β	Results
0	0	0	X	X	0	1	normal operation
0	1	0	0	X	1	1	Single Cycle
0	1	0	1	X	0	1	no change
0	0	1	1	X	0	1	no change
0	1	1	0	X	3	1	Single Instruction
0	0	1	0	X	3	1	Single Instruction
0	1	1	1	X	0	1	no change
1	0	0	0	X	2	0	Single Cycle
1	0	0	1	X	0	1	return to state 0
2	0	0	0	X	2	0	wait for Continue
2	0	0	1	X	0	1	return to state 0
2	1	0	1	X	2	1	allow one cycle of clock
3	0	0	1	1	0	1	return to state 0
3	0	0	0	1	3	1	no change
3	0	0	0	0	4	0	wait for Continue
4	0	0	0	1	4	0	wait for Continue
4	0	0	1	1	0	1	return to state 0
4	0	1	1	1	3	1	allow for 1 instruction

α = LASTMI*, β = PASSMCLK

Recall that R11 is SCY, R12 is SINST, and R13 is CONTINUE after these signals have been processed by CB or the Software Interface. Table 23 shows the states of Passmclk for Single Cycle, Single Instruction, and Continue. State 0 is normal operation (nothing stopped), state 1 is Single Cycle, state 2 is for waiting for Continue, state 3 is Single Instruction, and state 4 is for waiting for Continue. One instruction consists of passing through state 0, state 3, state 4, and back to state 0. One cycle consists of passing through state 0, state 1, state 2, and back to state 0. The finite state machine is of Mealy type. The signal PASSMCLK changes only when CMCLK is low. We used eight 74F74s (see Figure 16) to create the necessary logic for Passmclk.

Table 24 Passmclk in Break Mode

Present State	BREAK	R13	Next State	PASSMCLK	Comments
0	0	X	0	1	normal
0	1	0	1	0	break
1	X	0	1	0	wait for Continue
1	X	1	0	1	resume normal operation

As seen in Table 24, the signal ENBREAK is asserted in the Break Mode. These states are different from the previous states shown in Table 23. The signal PASSMCLK changes only when CMCLK is low.

Table 25 Passmclk in SAM

Present State	MEMREAD*.P	R13	Next State	PASSMCLK	Comments
0	1	X	0	1	normal
0	0	0	1	0	wait for data
1	X	0	1	0	wait for Continue
1	X	1	0	1	resume normal operation

As seen in Table 25, when the signal MEMREAD is asserted in SAM, PASSMCLK goes low stopping MCLK. When data have been entered onto MEMDATBUS(31:0) and ENTER* is asserted, MCLK continues. The signal R3 is asserted in SAM. These states are different from the previous states shown in Tables 23 and 24. The signal PASSMCLK changes only when CMCLK is low.

2.9. Opcodebus

The opcode is provided to VLSI-PLM via Opcodebus (a.k.a. Opcodebus_instren*). The opcode usually comes from CB. When CB is not able to provide the opcode in sufficient time, VPB will provide the opcode Noop during the time that it waits for CB to fetch the next opcode. Opcodebus provides the necessary logic for this to occur. The symbol and schematic for Opcodebus are shown in Figures 17 and 18, respectively. Opcodebus is composed of two blocks: Noop_dip and Halt_dip. The symbols and schematic for these blocks are shown in Figures 19 through 22. Noop_dip holds the value of the Noop opcode (04). Halt_dip holds the value of the Halt opcode (06).

Inputs

Table 26 Inputs to Opcodebus

Input	From where
CCLK.C	CB
IBUFEMPTY.C	CB
INSTREN*.P	VLSI-PLM
OPCODE.C(7:0)	CB
PASSMCLK	Passmclk
POR*	VPB
R3*	Pushbuttons

Outputs

Table 27 Outputs from Opcodebus

Output	To where
INSTREN*.C	CB
OPCODE.P(7:0)	VLSI-PLM

Functional Description

Table 28 shows the functional description for Opcodebus in SAM. Recall that R3* = 1 for the Cache Mode. When PASSMCLK is asserted in the Cache Mode, INSTREN* is tri-stated.

Table 28 Opcodebus in SAM

R3*	INSTREN*.P	OPCODE.P	INSTREN*.C
0	1	XX	Tri-stated
0	0	0X06 (Halt)	Tri-stated

Table 29 shows the functional description for Opcodebus in the Cache Mode. Q1 and Q2 are pins on Opcodebus that indicate the current state. State 0 is the normal mode. When INSTREN* = 0, a data transfer is requested from the prefetch buffer. When INSTREN* = 1, nothing is requested, so nothing is done. Please reference the timing diagrams for Preftech 1 (PF1), Preftech 2 (PF2), and IBUFEMPTY in [STN88]. As shown in Figure 18, we used four 74F244s, two 74F74s, along with the blocks Halt_dip and Noop_dip to create the necessary logic for Opcodebus.

Table 29 State Table for Opcodebus in Cache Mode

Current State			Next State			Comments
Q1Q2	α	β	Q1Q2	OPCODE.P	γ	
00	0	0	00	OPCODE.C	0	Normal state
00	0	1	01	Noop	0	ibuff is empty
00	1	0	00	OPCODE.C	1	no change
00	1	1	00	OPCODE.C	1	no change
01	0	0	11	OPCODE.C	1	ibuff not empty
01	0	1	01	XX	0	wait for instr.
01	1	0	10	XX	X	forbidden input
01	1	1	10	XX	X	forbidden input
10	0	0	10	XX	X	forbidden state
10	0	1	10	XX	X	forbidden state
10	1	0	10	XX	X	forbidden state
10	1	1	10	XX	X	forbidden state
11	0	0	00	OPCODE.C	1	execute opcode
11	0	1	00	OPCODE.C	1	in this state
11	1	0	10	XX	X	forbidden state
11	1	1	10	XX	X	forbidden state

α = INSTREN*.P, β = IBUFEMPTY.C, γ = INSTREN*.C

2.10. Forceaddr

The force branch (FORCEBR) signal from CB (or the user in SAM) directs VLSI-PLM to jump to the ROM location specified by FORCEADDR(8:0). In addition, FORCEBR also allows for the current ROM location to be displayed. Forceaddr implements these functions and also makes it possible for the user to stop VLSI-PLM at a specific ROM location. The symbol and schematic for Forceaddr are shown in Figures 23 and 24, respectively. There is one block inside of Forceaddr, called Fabreak. The symbol and schematic for Fabreak are shown in Figures 25 and 26, respectively.

Inputs

Table 30 Inputs to Forceaddr

Input	From where
FADDROE*	Toggle
FADDRTR*	Toggle
FORCEADDR.C(8:0)	CB
FORCEBR.P	Toggle
KPFA(8:0)	VPB
KPOE*	Toggle
MCLK	Clock Generator
POR*	VPB

Outputs

There are two outputs from Forceaddr. The output BREAK is sent to Passmclk for the break point feature. It is asserted when the value on FORCEADDR(8:0) equals the break point (register) value. The output FORCEADDR(8:0) is sent to VLSI-PLM.

Functional Description

Table 31 Forceaddr in Cache Mode

FADDROE*	FORCEBR.C	FADDRTR*	Direction of FORCEADDR(8:0)
0	0	1	from VLSI-PLM to CB
0	1	0	from CB to VLSI-PLM

Table 32 Forceaddr in SAM

FADDROE*	KPOE*	FORCEADDR.C(8:0)	Direction of FORCEADDR(8:0)
1	0	tri-state	from KPFA(8:0) to VLSI-PLM
1	1	tri-state	from VLSI-PLM to VPB

To implement the necessary logic for Forceaddr, we used two 74F245 octal bi-directional transceivers with tri-state I/O, two 74F244s, one 74F273, one 74F74, one 74F521 8-bit identity comparator, and the block Fabreak (see Figure 24). Fabreak is a dip switch that holds the value of the break point register. Note that OUTROMADDR is the complement of FORCEBR.

When in SAM, the Force Branch is done by first entering data onto KPFA(8:0) then asserting any of the Force Branch switches in Toggle. Toggle sets the necessary value of KPOE*. For the break point feature in SAM, if the value in BREAKDIP(8:0) is equal to the value of FORCEADDR(8:0) then BREAK is asserted. The signal BREAK goes to Passmclk depending on the value of ENBREAK and

BREAK. Passmclk stops MCLK and MCLK*. For the display, if OUTROMADDR is asserted (FORCEBR is unasserted) then the the ROM address from VLSI-PLM is sent out to FORCEADDR(8:0) and is latched by VPB for display purposes at the falling edge of MCLK (see the Force Branch and Force Branch Low timing diagrams in [STN88]).

2.11. Xcver

Xcver (transceiver) provides the necessary drive to VLSI-PLM and CB for the bi-directional data bus MEMDATBUS(31:0). The symbol and schematic for Xcver are shown in Figures 27 and 28, respectively.

Inputs

Table 33 Inputs to Xcver

Input	Description
INSTREN*.P	from VLSI-PLM
MEMDATBUS.C(31:0)	from CB
MEMDATBUS(31:0)	from VPB's Memdatbus
MEMREAD*.C	from CB
R3*	from Pushbuttons

Outputs

The only outputs is MEMDATBUS(31:0) to both the Memdatbus block and CB.

Functional Description

Table 34 Functional Description of Xcver

R3	MEMREAD*.C	INSTREN*.P	Direction
1	X	X	MEMDATBUS.C(31:0) is tri-stated
0	0	X	from CB to VPB
0	X	0	from CB to VPB
else	X	X	from VPB to CB

We used four 74F245s to implement the necessary logic for Xcver, as shown in Figure 28. When sending MEMDATBUS(31:0) between CB and VPB the delay is about 4.6 ns. This may be critical during the memory read operation (see MEMREAD* timing diagram [STN88]) and during the PF1 and PF2 operations.

2.12. Memdatbus

Data to and from VLSI-PLM are communicated through MEMDATBUS(31:0). The destination and source may be CB or the Software Interface. In addition, the Micro Instruction Register (MIR) and the Status Register (STATUS) contents can be shifted into (or out of) the Software Interface from (or to) MEMDATBUS(31:0) for testing. Memdatbus contains the necessary logic to implement these functions.

Memdatbus is composed of the four blocks: Controller, Iointerface, Bootreg, and Tri-State Buffer. The symbol and schematic sheet for Memdatbus are shown in Figures 29 and 30, respectively. The schematic contains the four symbols for the blocks of Memdatbus. Each of these four symbols itself contains a schematic sheet.

Inputs

Table 35 Inputs to Memdatbus

Input	Description
KPMD(31:0)	keypad connector
MEMDATBUS.C(31:0)	from Xcver
MEMDATBUS.P(31:0)	from VLSI-PLM
R6*	from Toggle
R8	from Pushbuttons
R10	from Pushbuttons
R20	from Pushbuttons
R21*	from Toggle
RLMDR*	from Toggle
SHIFTA	from Clock Generator
SHIFTOUT1	from VLSI-PLM
SHIFTOUT2	from VLSI-PLM
S9	from Pushbuttons

Outputs

The output of Memdatbus is shown in Table 36 below. The output MIRCOM is sent to the Software Interface to indicate that the contents of MIR have been shifted. STATCOM is sent to the Software Interface to indicate that the contents of STATUS have been shifted.

Table 36 Outputs from Memdatbus

Output	To where	Function
MEMDATBUS.C(31:0)	to Xcver	Memory Data Bus
MEMDATBUS.P(31:0)	to VLSI-PLM	Memory Data Bus
MIRCOM	Software Interface	indicates shifting of MIR completed
SHIFTIN1	VLSI-PLM	data for first scan path
SHIFTIN2	VLSI-PLM	data for second scan path
STATCOM	Software Interface	indicates shifting of STATUS completed
STOPSHIFTA*	Clock Generator	inhibits SHIFTA
RESUME*	Clock Generator	allows SHIFTA to resume

Functional Description

Table 37 Functional Description of Memdatbus

Signal	Please refer to
MIRCOM	Controller
SHIFTIN1	Iointerface
SHIFTIN2	Iointerface
STOPSHIFTA*	Controller
STATCOM	Controller
RESUME*	Controller

In the Cache Mode, Memdatbus drives MEMDATBUS(31:0) from VLSI-PLM to CB, and vice versa, as shown in Figure 30. Please refer to the functional description of the blocks Toggle and Pushbuttons. During the reset of VLSI-PLM on the cycle after receiving RESET*, VLSI-PLM expects the boot value (0X1FFFFC00) to be on MEMDATBUS(31:0).

In SAM, VPB performs Test1 (Testin or Testout) and Test2. If the Test1 and Testout functions are selected on the tester by the user, the contents of the MIR are shifted out to MEMDATBUS(31:0) one word (32 bits) at a time and displayed. In Test2 and Testout, the contents of the STATUS Register are shifted out to the high-order-bits of MEMDATBUS(31:0). Zeroes are added to the left of these 18 bits for display purposes by the Software Interface.

2.12.1. Controller

All internal control signals for the Memdatbus block and external control signals (to and from Clock Generator, to and from Software Interface, etc.) are generated and arbitrated in Controller. The symbol and schematic for Controller are shown in Figures 31 and 32, respectively.

Inputs

Table 38 Inputs to Controller

Input	From where
S9	Pushbuttons
SHIFTA	Clock Generator
R6*	Toggle
R8	Pushbuttons
R10	Pushbuttons
R20	Pushbuttons
R21*	Pushbuttons

Outputs

Table 39 Outputs from Controller

Output	To where	Description
L_S*	Iointerface	shifts or loads the shift registers
MIRCOM	Software Interface	indicates shifting of MIR completed
OEDATA*	Iointerface	output enables the shift registers
RESUME*	Clock Generator	allows SHIFTA to resume
STATCOM	Software Interface	indicates shifting of STATUS completed
STOPSHIFTA*	Clock Generator	inhibits SHIFTA

Functional Description

The Controller is responsible for collecting the shiftout data from MIR and STATUS into a word and sending it on MEMDATBUS(31:0) to the Software Interface. We used two 74F169 4-stage synchronous bi-directional counters and one 74F74 to implement Controller correctly (see Figure 32). The counters were used to count the number of words being shifted to and from Memdatbus. To shift STATUS data in Test2 Mode: when 17 bits have been counted, C17 is enabled to stop SHIFTA. To shift MIR data in Test1 Mode: when 31 bits have been counted, C31 is enabled to stop SHIFTA. Controller issues R20 to clear the counters when the Test button is pressed on the tester. The signal MIRCOM is asserted during the 159th cycle in Test1 Mode. The signal STATCOM is asserted during the 17th cycle in Test2 Mode.

Table 40 Functional Description of Controller

S9	R6*	R21*	α	Comments
0	X	0	0	Resume SHIFTA (in Shiftout Mode) when CONTSHOUT* (R21*) is asserted.
0	X	1	1	Do not resume SHIFTA.
1	0	X	0	Resume SHIFTA (in Shiftin Mode) when ENTER* (R6*) is asserted.
1	1	X	1	Do not resume SHIFTA.
S9	R6*		L_S*	
0	X		0	If in Shiftout Mode, then only shift.
1	0		1	If in Shiftin Mode, then load when data have been entered on bus.
1	1		0	If in Shiftin Mode, then shift when data have been entered on bus.
R8	C17	C31	β	
0	1	X	0	In Test2 Mode, inhibit SHIFTA when 18 bits have been shifted.
0	0	X	1	In Test2 Mode, do not stop SHIFTA; has not been 18 bits yet.
1	X	1	0	In Test1 Mode, inhibit SHIFTA when 32 bits have been shifted.
1	X	0	1	In Test1 Mode, do not stop SHIFTA; has not been 32 bits yet.
S9	β	α	γ	
0	0	0	X	In Testout Mode, do not output to bus when SHIFTA is resumed.
0	0	1	0	In Testout Mode, put STATUS or MIR contents out MEMDATBUS(31:0).
0	1	X	1	In Testout Mode, do not output to bus when shifting.
1	X	X	1	In Testin Mode, never output data from shift registers to MEMDATBUS(31:0).

α = RESUME*, β = STOPSHIFTA*, γ = OEDATA*

2.12.2. Iointerface

Iointerface allows for data to be shifted serially into the SHIFTIN pin of VLSI-PLM. It also allows for serial data from SHIFTOUT1 and SHIFTOUT2 to be collected and displayed on the tester. The symbol and schematic are shown in Figures 33 and 34, respectively.

Inputs

Table 41 Inputs to Iointerface

Input	From where
L_S*	Controller
MEMDATBUS(31:0)	VLSI-PLM
OEDATA*	Controller
R8	Pushbuttons
SHIFTA	Clock Generator
SHIFTOUT1	VLSI-PLM
SHIFTOUT2	VLSI-PLM

Outputs

The outputs from Iointerface are MEMDATBUS(31:0) and SHIFTIN. Both are sent to VLSI-PLM.

Functional Description

The signal SHIFTIN is always yielding data. The data are valid only when the Testin Mode is selected on the tester during testing of VPB. When the Software Interface is ready to display data, Controller issues OEDATA* to make the MEMDATBUS(31:0) value be shifted, via Iointerface, to the display. If SHIFTOUT1 is asserted, the MIR data will be shifted out of VLSI-PLM. If SHIFTOUT2 is asserted, the STATUS data will be shifted out. One 74F241 and four 74F299 8-input universal shift registers with common parallel I/O pins were used to create the necessary logic for Iointerface.

Table 42 Functional Description of Iointerface

L_S*	Operation of Shift Registers
0	Shift
1	Parallel load
OEDATA*	
0	data from Shift Registers to MEMDATBUS(31:0)
1	outputs of Shift Registers are tri-stated

2.12.3. Bootreg

Bootreg holds the initial value needed by the Memory Data Register (MDR) for a reset. Bootreg is composed of two blocks called Bootdip and Tri-State Buffer. The Tri-State Buffer is the same block that is on the Memdatbus sheet. The symbol and schematic for Bootreg are shown in Figures 35 and 36.

Toggle sends the input RLMDR* to Bootreg. The output from Bootreg is BOOTREG(31:0); it is sent to MEMDATBUS.P(31:0). A functional description of Bootreg is shown in Table 43 below. Figures 37 and 38 show the symbol and schematic for Bootdip, respectively. Bootdip holds the value (0X1FFFFC00) to boot up VLSI-PLM in BOOTVAL(31:0). This is the first value that MDR needs on power-up.

Table 43 Functional Description of Bootreg

RLMDR*	BOOTREG(31:0)
0	desired value in Bootdip
1	Tri-state

2.12.4. Tri-State Buffer

Tri-State Buffer is used frequently between a source and bus to provide control. The input IN(31:0) is a 32-bit bus. The input EN* is the active low output enable. The output is OUT(31:0), a 32-bit output. When RLMDR* is enabled, Tri-State Buffer allows the boot value in BOOTVAL(31:0) to be sent to MEMDATBUS(31:0). We used a 74F244 to provide the necessary drive for the bus. The symbol and schematic for Tri-State Buffer are shown in Figures 39 and 40, respectively.

Table 44 Functional Description of Tri-State Buffer

EN*	OUT(31:0)
0	IN(31:0)
1	Tri-state

3. VLSI-PLM Board Simulation

In this section we present the simulation results and comments for the gate-level VPB design. After the design schematics were completed and correct, we verified the timing and functionality of the gate-level design by performing simulation tests on all of the VPB blocks. The MG tool *Expand* was used to transform the multi-level hierarchical schematics into a flat array of primitive elements consisting only of basic cells and wires. A basic cell is a digital component that has associated with it a computer program that models its functional behavior.

The simulations were performed by using the MG *QuickSim* tool. QuickSim interactively allows for the verification of the functionality and timing of the design. Please see [BuS89] for a detailed description of the MG simulation tools. Simulations at the top-level of VPB were done with Version 6.1 MG software; the blocks were simulated using Version 5.3 MG software. In this discussion we often refer to the .do files. These are files that contain the exact line-by-line instructions for the entire simulation of the block. These .do files are called from within QuickSim permitting one to easily run the simulations. We include one sample of the .do files, Reset1.do, in the Appendix. Every VPB block has been extensively simulated using test patterns similar to Reset1.do. These test patterns are not included in this report, but will be supplied if requested.

We also describe differences between the gate-level VPB design and the wire-wrapped board. The differences were few, allowing for total timing and functional compatibility. The tables referred to in this discussion appear in section 2 with the corresponding description of the blocks.

3.1. VLSI-PLM Board

Simulation at the top level of VPB consisted of the tests Reset0, Reset1, TestA0, TestA, and TestB0. Please refer to sections 4.2 and 4.3 for a step by step description of TestA and TestB. The test TestA0 is a shortened version of TestA. The results were as expected.

The tests Reset0 and Reset1 are the same except for the period of CMCLK. The signal CMCLK is set at 5 MHz for Reset0 and 10 MHz for Reset1. VPB behaved as expected, stopping MCLK and MCLK* after each Memread and resuming after data were entered to Memdatbus. In addition, the timing of RESET* and RLMDR* is as desired. The Reset1.do file is in the Appendix along with the corresponding trace picture and Reset1.list file. The Reset1.do file contains its own comments and shows the step by step testing procedure used to simulate VPB. The picture of the trace results shows the values of certain signals from 0 to 946 clock cycles. The Reset1.list file shows the numerical version of the trace signals.

The test TestB0 is a shortened version of TestB. This test was performed with SHIFTA at 20 MHz. VPB behaved as expected, interacting with the user by waiting for signals to start and resume.

3.2. Clock Generator

The simulation results of Clock Generator followed the specifications in Tables 8 and 9. The signals MCLK and MCLK* behaved correctly in the forbidden state. The clock skew used in the simulations is shown in Figure 41. All clocks have a period of 100 ns. CCLK is always running. The CB output clock, CMCLK, has its starting and falling edges 1 ns after the corresponding edges of CCLK. The clocks MCLK and MCLK* are for VLSI-PLM. MCLK rises and falls less than 5 ns after CCLK; MCLK* rises less than 3.5 ns after MCLK and falls less than 4 ns after MCLK. The clocks CMCLK and CMCLK* are used internally by VPB. MCLK rises and falls with CMCLK; MCLK* rises and falls with CMCLK*.

There were a few differences between Clock Generator and the wire-wrapped board. The TTL parts 74F125 and 74F126 were used on the wire-wrapped board. These parts were not available in the MG parts library. The parts 74F244 and 74F241 were used in the computer design of the board, preserving timing and functional compatibility. On the wire-wrapped board, SHIFTA is not driven by a buffer. This is acceptable since the drive of the 74F11 is adequate. The delay associated with the buffer is also eliminated, becoming an advantage at fast frequencies of SHIFTA.

3.3. Pushbuttons

Table 45 Pushbuttons Test Modes

TESTIN	TEST1	TEST2	Test Mode
0	0	0	None
0	0	1	Testout, Test2
0	1	0	Testout, Test1
0	1	1	Testout, Test1
1	0	0	None
1	0	1	Testin, Test2
1	1	0	Testin, Test1
1	1	1	Testin, Test1

The above Table 45 shows the test modes for Pushbuttons. The three tests Testin, Test1, and Test2 were performed on Pushbuttons, with simulation results as expected. Some of the signals on the wire-wrapped board are driven directly and do not go through 74F244. There is total timing and functional compatibility, but the drive capabilities are different.

3.4. Toggle

The simulation results of Toggle followed the specifications in Tables 15 and 16. One should pay special attention to the results of the switches S13, S14, S15, S16, and S17. There were no differences with the wire-wrapped board.

3.5. Atomicb

The simulation results of Atomicb were as expected. There were no differences with the wire-wrapped board.

3.6. MIRsignals

The simulation results of MIRsignals were as expected and the delay times were adequate. There were no differences with the wire-wrapped board.

3.7. MARbuff

The simulation results of MARbuff followed the specifications in Table 21 and show that there is no delay through 74F244. There were no differences with the wire-wrapped board. With the earlier version of MG CAD tools, there was a bug in 74F244, resulting in no delay times. This bug has been fixed in the current 6.1 version of MG software.

3.8. Passmclk

The simulation results of Passmclk followed the specifications in Tables 23, 24, and 25. When more than one input is asserted, one should pay special attention to its behavior. There were no differences with the wire-wrapped board.

3.9. Opcodebus

The simulation results of Opcodebus followed the specifications in Tables 28 and 29. There were a few differences with the wire-wrapped board. DIP switches were not used on the wire-wrapped board. The Halt and Noop values are clocked into 74F374s upon power-up. Once these values have been latched in, there is functional and timing compatibility between 74F374 and 74F244. The wire-wrapped board

needs to be updated to use 74AS20. Use of the 74AS20 reduces the the delay from the IBUFEMPTY to the Noop Enable by one logic level. This is important since Noop needs to be latched during phase 0. The next-state logic of Q1 on the wire-wrapped board also needs to be changed. There is no functional change, but there is a race condition in the next-state logic of Q1.

3.10. Forceaddr

The simulation results of Forceaddr followed the specifications in Tables 31 and 32, except for display D. The source of the problem is not known at this time. It might lie within the part 74F273 or in the use of the bus ripper. Since it is only a display, this problem does not affect other parts in VPB.

The differences with the wire-wrapped board were few. DIP switches were not used on the wire-wrapped board. Instead, the desired values are clocked into tri-state registers (74F374). Once data have been latched in, there is functional and timing compatibility between 74F374 and 74F244.

On the wire-wrapped board, 74F374 was used in place of 74F273. We used 74F273 because of its Master Reset feature. The 74F374 is a better choice because there are simulation problems with the 74F273. Since this register is only used for display purposes, either one will do. On the wire-wrapped board the 9-bit 74F863 was used in place of the two 74F245 to save hardware. The 74F863 was not available in the MG parts library. The gate-level design uses two 74F245s, preserving timing and functional compatibility.

There is a discrepancy in the way BREAK is determined. From the simulation results, BREAK is asserted (after a propagation delay) whenever the value on FORCEADDR(8:0) is equal to the break point regardless of the value of FORCEBR. On the wire-wrapped board, BREAK is only allowed to be asserted during Force Branch.

3.11. Xcver

The simulation results of Xcver followed the specifications in Table 34. There were no differences with the wire-wrapped board.

3.12. Memdatbus and Clock

A simulation on Memdatbus alone was not done; it was performed on Memdatbus connected to Clock Generator. We simulated in this way because Memdatbus needs to communicate with Clock Generator to stop and start SHIFTA. Memdatbus behaved as expected in all simulations.

Simulations were performed separately on all blocks of Memdatbus, except for Tri-State Buffer. Tri-State Buffer was simulated with Bootreg. In addition to simulating the blocks separately, those that communicated with each other were connected and simulated to verify their expected behavior. The blocks were simulated at 25 ns cycles (40 MHz), except for Iointerface. Iointerface was simulated at 50 ns cycles. This was necessary because the cell 74F299 was not available in the MG parts library for the simulation. Thus 74LS299 was substituted for the tests. From hand calculation, Iointerface should be able to operate at 40 MHz.

On the wire-wrapped board, the Software Interface latches KPMD(31:0) data into 74F374s. Once data have been clocked in, the behaviors of 74F374 and 74F244 have timing and functional compatibility.

3.12.1. Controller

The simulation results of Controller followed the specifications in Table 40. One should pay special attention to when the signals STATCOM and MIRCOM are asserted. The two 74F169 4-bit counters on the wire-wrapped board were replaced by the 8-bit counters 74F269, preserving functional and timing compatibility.

3.12.2. Iointerface

An exhaustive simulation was done on Iointerface. The results followed the specifications in Table 42. Four tests were done, corresponding to the conditions for Test2 and Testout, Test1 and Testout, Test2 and Testin, and Test1 and Testin.

3.12.3. Bootreg and Tri-State Buffer

The simulation results of Bootreg and Tri-State Buffer followed the specifications in Tables 43 and 44, respectively. The Software Interface latches the boot value in 74F374 at power-up to facilitate changes in the boot value. After the boot value is set, 74F374 behaves similar to 74F244 with total

timing compatible. The Software Interface takes the place of Bootdip for providing the boot value (0X1FFFFC00).

4. Testing of the Wire-wrapped VLSI-PLM Board

After determining that VPB was correct with respect to functional and timing specifications, we wire-wrapped it using a 9U VME board. The wire-wrapped board achieves stand-alone operation of VLSI-PLM for debugging and provides interface to the Xenologic Corporation's X-1 cache board. The board has 95 SSI/MSI chips including VLSI-PLM. Many of the SSI chips are drivers and shift registers for loading and capturing data from the scan path of VLSI-PLM. The board has a microprocessor-controlled Software Interface connected to it for automating the testing, providing inputs to the board's switches, displaying desired results, and providing the board with desired constants (such as Noop and Halt opcode values and the boot value of MDR).

In this section the testing and the test results of the wire-wrapped VPB are discussed. The various clock speeds range from 4 to 20 MHz. This flexibility and the ability to test at low frequencies is of great importance. The testing of VPB consisted of checking for connectivity and functionality. The next section describes how to use the tester to run a test. We also describe two of the tests for VPB: TestA and TestB.

4.1. Software Interface

In this section, we describe how to use the Software Interface to test the wire-wrapped VPB. We briefly describe the functionality of each of the sixteen keys on the tester box. The tester provides VPB with the switches S1 through S17, inputs to FORCEADDR(8:0) and MEMDATBUS(31:0), POR*, SHIFTA, and CMCLK. Eventually we will modify the Software Interface to use the switches as buttons to further automate the testing.

When the tester is first turned on it is in local mode, and S3 (SAM) is 1. The Clock is set to run at 20 MHz. The tester goes through a power-up sequence as follows. The registers Memdat Boot, Forceaddr Breakpoint, and Opcode Noop are initialized to 0, and Opcode Halt is initialized to 6. A switch-reset subroutine is performed, pulsing S1 (RST*) and turning off the switches S2, S4, S7, S8, S9, S10, S11, S12, S13*, S14, S15, S16 and S17. Recall that these switches correspond to the signals FORCEBR,

ENBREAK, TEST1, TEST2, TESTIN, SINST, CONTINUE, SCY, CONTSHOUT*, FBRCSY, FBRSI, SCYCONT, and SINCONT, respectively. A boot subroutine is performed next, yielding a constant value to Memdat Boot and pulsing S1. When cache miss did not happen within a certain time period, this message appears:

```
Error on booting. Mcy-[n]
Cache miss did not happen
```

Where n = 1 to 16 indicates the clock cycle with the error. Pressing any key returns the user to the normal display mode. Otherwise, different constant values are sent out to MEMDATBUS(31:0) sixteen times, with each transfer pulsing S6* (ENTER*) and checking for a cache miss. When all goes well, the current values of MEMDATBUS(31:0), MARBUS(27:0), and OUTROMADDR are continuously displayed. Note that MDT, MAR, and RAD denote MEMDATBUS(31:0), MARBUS(27:0), and ROMBUS (the bus to the ROM), respectively.

```
MDT= xxxx xxxx
MAR= xxxx xxxx  RAD= x xx
```

We now describe each of the sixteen keys on the tester. Table 46 shows where each key is placed.

Table 46 Tester Keys

	Local-Cache		FBR	SI
	Reset		SC	CONT
Back Space	Test	Regs	Forceadr	Memdat
Enter	Display	Break-Point	Outromadr	Outmemdat

Enter and Backspace

The *Enter* key is used to send all the commands to VPB. The *Backspace* key is used when typing in the commands on the tester.

Local/Cache

If the unit is currently in the local mode, pressing *Local/Cache* switches it to the Cache Mode, and vice versa. This affects S3. When in the local mode, the switch-reset and boot subroutines are

performed. Otherwise, the word Cache will be displayed in the upper right hand corner of the display and only the switch-reset subroutine is performed.

Reset

This key is active only in the local mode. It performs the power-up sequence as described above.

Display

This key shows the current value of MEMDATBUS(31:0), MARBUS(27:0), and ROMBUS. Pressing any key exits to the normal mode.

```
MDT= xxxx xxxx
MAR= xxxx xxxx  RAD= x xx
```

Continue (CONT)

The key *CONT* is active only in the local mode, pulsing S11.

Breakpoint

```
Breakpoint is disabled
at x xx  1-enable  2-new
```

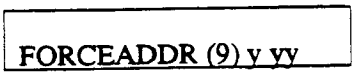
The key *Breakpoint* is active only in the local mode. The signal BREAKPOINT is disabled when S4 = 0. Pressing the number key **1** toggles S4; pressing **2** allows the user to enter a new value into the Forceaddr Breakpoint register.

```
Old Breakpoint = x xx
new value = y yy
```

The value is output to FORCEADDR(8:0) when *Enter* is pressed. Then S4 changes its state from disable to enable, or vice versa.

Forceadr

The key *Forceadr* is active only in the local mode. This key allows the force branch address, FORCEADDR(8:0), to be entered on the keypad. The value is transferred to FORCEADDR(8:0) and S5 (OUTMEMDAT) is pulsed when *Enter* is pressed.

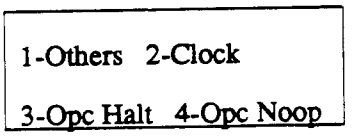


Force Branch (FBR)

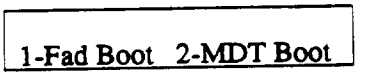
The key *FBR* is active only in the local mode, pulsing S2. This has to be used in conjunction with *Forceadr*.

Registers (Regs)

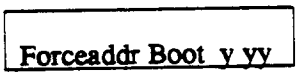
There are four registers to choose from. All are active only in the local mode.



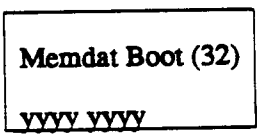
Others



When 1 is selected, the user must enter a value for Forceaddr Boot. This value is sent to FORCEADDR(8:0) when *Enter* is pressed.



When 2 is pressed, the user must enter a new value for Memdat Boot. This value is sent to MEMDATBUS(31:0) when *Enter* is pressed.



Clock

```
1-CMCLK  2-SHIFTA
```

Pressing **1** allows the value of CMCLK to be changed.

```
1-Increase  2-Decrease
CMCLK = 4.0 MHz
```

Pressing **2** allows SHIFTA to be changed.

```
1-Increase  2-Decrease
SHIFTA = 4.0 MHz
```

The values available now are 4.0 MHz and 20.0 MHz only, and can be changed with the number keys **1** and **2**. When the desired value has been reached, pressing *Enter* enters the value. If no change is desired, pressing *Regs* returns the user to the *Regs* display.

Opc Halt

The clock Opc Halt waits for the new value of the Opcode Halt register to be entered on the keypad by the user. It outputs this value to OPCODE(7:0) when *Enter* is pressed.

```
Opcode Halt (8) yy
```

Opc Noop

The clock Opc Noop looks for the new value of the Opcode Noop register on the keypad. This value is output to OPCODE(7:0) when *Enter* is pressed.

```
Opcode Noop (8) yy
```

Single Instruction (SI)

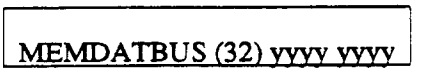
The key *SI* is active only in the local mode. It pulses S10 if PASSMCLK* = 1 and pulses S17 otherwise.

Single Cycle (SC)

The key *SC* is active only in the local mode. It pulses S12 if PASSMCLK* = 1 and pulses S16 otherwise.

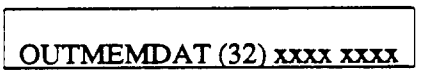
Memdat

The key *Memdat* is only active in the local mode. It allows the new value of MEMDATBUS(31:0) to be entered on the keyboard. This value is transferred to MEMDATBUS(31:0) and S6 is pulsed when *Enter* is pressed.



Outmemdat

The key *Outmemdat* is active only in the local mode. It displays the present value of OUTMEMDAT.

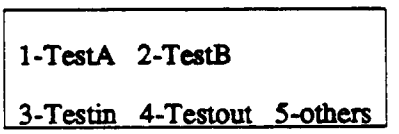


Outromadr

The key *Outromadr* is active only in the local mode. It displays the present value of ROM address in VLSI-PLM.

Test

The *Test* key allows the user to test VPB with the Software Interface. There are four automatic tests to choose from. Each test has an algorithm programmed in the ROM of VLSI-PLM.



TestA

The test TestA outputs 1FB to Forceaddr and pulses S2. If the time is out, it displays this message with a description of the instructions on the bottom of the tester's display:

cache miss timed out

The user may press any key to continue the testing, or *Test* to return to the testing display menu. Otherwise, TestA outputs the constant 1 to MEMDATBUS(31:0), outputs 2 to FORCEADDR(8:0), pulses S2, outputs 3 to FORCEADDR(8:0), pulses S2, outputs 17C to FORCEADDR(8:0), and pulses S2. MEMDATBUS(31:0) is then read to compare with 1. If this fails, this message is displayed:

Fail on transferring

The user may press any key to continue, or *Test* to return to the testing display menu. A check is made to see if cache miss happens. If it does, this message is displayed:

Unexpected cache miss

The user may press any key to continue, or *Test* to return to the testing display menu. Otherwise, this sequence is performed eight times. When all eight runs pass, this message is displayed:

Test A completed

TestB

TestB outputs the integer i to FORCEADDR(8:0), where $i = 0$ initially. The switch S14 is pulsed and the value of SHIFTA is output to the clock. The signal RST* is pulsed, S7 is set, and S13 is pulsed. Next, the time out loop is set to wait for OEDATA*. If OEDATA* did not happen, the tester displays the message:

Interface board error. Didn't receive OEDATA*

Otherwise, 160 bits are read from MEMDATBUS(31:0) and compared with the value in the ROM. If these values do not match, the following is displayed:

```
Bad ROM at (location i)
1-Continue  2-Show
```

The user may press **1** to continue testing on the next address, or **2** to examine the ascii contents of the ROM. If **2** is pressed, the following is displayed:

```
Instruction is
(content of ROM)
```

The user may press any key to examine the contents in the ROM, or *Test* to return to the testing menu. If the user wants to examine the ROM contents, the byte in error, the data value read from MEMDATBUS(31:0), and the expected value in the ROM are displayed as follows.

```
ROM at (i)  byte (x)
ROM =   not =
```

The user may press any key to resume the testing, or *Test* to return to the testing menu. If the testing continues, *i* increments by 1 for the next address, and the whole process repeats until *i* = 512. When no error is encountered, this message is displayed:

```
ROM contents o.k.
```

Otherwise, the test returns to the testing display mode when done.

Testin

```
1-MIR (32)  [in]
2-Status reg. (18)
```

If the user selects **1**, S9 and S7 are set. The new value of MEMDATBUS(31:0) is looked for on the keyboard for five scans. This value is transferred to MEMDATBUS(31:0), and S6 is pulsed each time *Enter* is pressed.

```
MIR (32)  n of 5  [in]
enter new word yyyy yyyy
```

If the user selects **2**, S9 and S7 are set. The new value of MEMDATBUS(31:0) is looked for on the keyboard. This value is transferred to MEMDATBUS(31:0), and S6 is pulsed each time *Enter* is pressed. An example for STATUS is 00031432.

```
Status reg. (18)  [in]
Enter value yyyy yyyy
```

Testout

```
1-MIR (32)  [in]
2-Status reg. (18)
```

If the user presses **1**, S7 is set. This test pauses to wait for any key entry to proceed. The following is displayed:

```
MIR (32)  [out]
Hit any key to start
```

The contents of MIR are displayed five times. With each display, S6 is pulsed and the timeout loop is set to wait for OEDATA*. An error message is displayed if no OEDATA* is received. Otherwise, data are read from MEMDATBUS(31:0) and S7 is reset until there is user input from the keyboard. This process repeats five times. The user may stop the process at any time by pressing *Test*.

```
MIR (32)  [out]
word n of 5 xxxx xxxx
```

If the user selects **2**, S8 is set. The test waits for the user to press another key.

```
Status reg. (18)  [out]
Hit any key to start
```

Testout shifts 18 bits in from MIR if OEDATA* is set accordingly, and displays the value. S8 is reset when completed.

Others

The user may also select the automatic tests Test1 or Test2. Test1 toggles S7 and waits for any key to be pressed, except for *Test*, to pulse S13. When *Test* is pressed, S7 returns to its normal state.

Test2 toggles S2 and waits for any key to be pressed by the user, except for *Test*, to pulse S13. When *Test* is pressed, S8 returns to its normal state.

4.2. TestA

This section describes the step-by-step procedure for TestA for the wire-wrapped VPB. The test algorithm is programmed into the ROM of VLSI-PLM so that it is performed automatically when the user selects the *TestA* option on the Software Interface. The algorithm selects the tester buttons that in turn assert the necessary signals, as explained in section 4.1. To start the test, the tester and DC power supply must be turned on. The voltage adjustment switch should be turned up so the voltage is 5 volts. This description is given in terms of the buttons (in italics) on the tester. Memreadd, memwrited, mdrtomar (MDR to MAR), and martomdr (MAR to MDR) are some of the 512 microstates in the ROM of VLSI-PLM.

1 VPB is initialized by issuing *Reset*. This puts VPB in SAM and performs the cold reset sequence.

2 The ROM address of memreadd is put on KPFA(8:0) for several cycles (actual number not important, since *FBR* is clocked in and pulsed before use by VPB). The memread state is executed and MEMREAD* is asserted. MEMREAD* is detected by VPB, making VPB stop MCLK and allowing MEMDATBUS(31:0) to be entered from the tester.

3 The code 0X00000000 is put on KPMD(31:0). *Enter* is issued for several cycles. This allows MDR to be loaded with the value on KPMD(31:0).

4 The ROM address of the mdrtomar state is set on KPFA(8:0). *FBR* is issued for several cycles. The ROM address of the martomdr state is put on KPFA(8:0). *FBR* is issued for several cycles. The ROM address of memwrited is put on KPFA(8:0). *FBR* is issued again for several cycles. The user may observe data from MEMDATBUS(31:0) on the display. If they are the same as data supplied in previous step, the user may proceed with testing by pressing any tester button. If they are not the same, the user may stop by pressing *Test*. This means the chip or the Software Interface logic has error.

5 The test repeats steps 2,3, and 4 for each of the VLSI-PLM internal registers T, T1, R, H, H2, S, and N, using the corresponding microstates.

6 Steps 2 through 5 are repeated with the data on KPMD(31:0) as: 0X33333333, 0X55555555, 0X77777777, 0X99999999, 0XAAAAAAAA, 0XCCCCCCCC, 0XFFFFFFFF.

4.3. TestB

This section describes the step-by-step procedure for TestB for the wire-wrapped VPB. The test algorithm is programmed into the ROM of VLSI-PLM so that it is performed automatically when the user selects the *TestB* option on the Software Interface. The algorithm selects the tester buttons that in turn assert the necessary signals, as explained in section 4.1. To start the test, the tester and DC power supply must be turned on. The voltage adjustment switch should be turned up so the voltage is 5 volts.

1 VPB is initialized by issuing *Reset*. This puts VPB in SAM and performs the cold reset sequence.

2 0X000 is put on KPFA(8:0).

3 *CONT* is issued, followed by *FBR* and *SC* on the next cycle for several cycles of CMCLK.B. The Toggle block of VPB then asserts CONTINUE, and FORCEBR and SCY on the next cycle. This allows VLSI-PLM to Force Branch to the desired ROM location, output ROM Address on the next cycle, and stop in phase 1 (MCLK is low). This in turn allows the MIR contents to be shifted.

4 A frequency is selected for SHIFTA.

5 *Testout* is issued.

6 *Test1* is issued.

7 CONTSHOUT* is asserted for several cycles. This starts the shift out process. After 32 cycles of SHIFTA, the first (least significant bit) word of MIR is available on MEMDATBUS(31:0) and is displayed by the Software Interface. SHIFTA stops to allow data to be analyzed. This is repeated until all 5 words of MIR (160 bits) have been shifted out.

8 *Test1* is released, thereby stopping the shifting process.

9 The value on the KPFA(8:0) is incremented by one until it reaches 511. Then the test branches to step 3.

10 If the values scanned from the ROM are as expected, TestB proceeds to the next test. If the values differ, the chip is bad because of the ROM contents, and the test should be stopped by the user.

5. Conclusion

In this paper, we have described the design and construction of the VLSI-PLM Board. The main purpose of the VLSI-PLM Board is to test the performance of the VLSI-PLM Chip on a stand-alone base and to interface the chip to Xenologic's X-1 cache board. The gate-level design was created and simulated on an Apollo workstation using Mentor Graphics software tools. The resulting design was then wire-wrapped. The VLSI-PLM Chip was placed on the VLSI-PLM Board and tested at 40 MHz with TestA, TestB, and other test programs. Benchmarks were performed, with the VLSI-PLM Chip stopping before the end of the test. We are currently trying to analyze the reason why it stopped.

Future considerations include modifying the Software Interface to have the switch signals as the buttons to further automate the testing of the wire-wrapped VLSI-PLM Board. We also plan to interface the board to processors such as those in NEXT Computers. A PC board version of the wire-wrapped board has been designed using MG Board Station tools [BuS89]. We are currently simulating the PC board to verify the design of the wire-wrapped board. The remaining space on the wire-wrapped board will be used for caches, prefetcher, and bus interfaces. When completed, the board will be used as a complete processor node.

Acknowledgement

The VLSI-PLM Board was wire-wrapped with assistance from Joao Wentcovitch and Katherina Law in the Electronics Research Lab. We are thankful to Katherina for the Software Interface instructions. We are thankful to Tam Nguyen, Pang Swee-Chee, Tara Weber, and Alvin Despain for their comments. We are also thankful to the other members of the Aquarius project for their help, comments, and suggestions.

This research was partially sponsored by Defense Advanced Research Projects Agency (DoD) and monitored by Space & Naval Warfare Systems Command under Contract No. N00039-84-C-0089, NCR Corporation in Dayton, Ohio, and National Science Foundation. Equipment and other support for the project has been provided by DEC, NCR, Apollo, ESL, and Xenologic.

References

[BuS89] L. G. Bushnell and V. P. Srini, *A Computer Aided Design Methodology for Printed Circuit Boards*, UCB/ CSD 89/492, Computer Science Division, EECS, University of California, Berkeley, California, January 1989.

[Men88] Mentor Graphics Corporation, Version 6.1 Software Documentation, Beverton, Oregon, 1988.

[STN88] V. P. Srini, J. V. Tam, T. M. Nguyen, B. K. Holmer, Y. N. Patt and A. M. Despain, *Design and Implementation of a CMOS Chip for Prolog*, UCB/ CSD 88/412, Computer Science Division, EECS, University of California, Berkeley, California, March 1988.

Index of Signals

APPENDIX

The appendix includes all of the figures referenced in the paper. We also include the QuickSim file Reset1.do, trace results, and numerical results from the simulation test Reset1. The Reset1.do file contains its own comments. This file was called from within QuickSim to run the entire Reset1 test on the gate-level VPB. The picture of the trace results shows the values of certain signals from 0 to 946 clock cycles. The numerical results are just the numerical version of the trace signals.

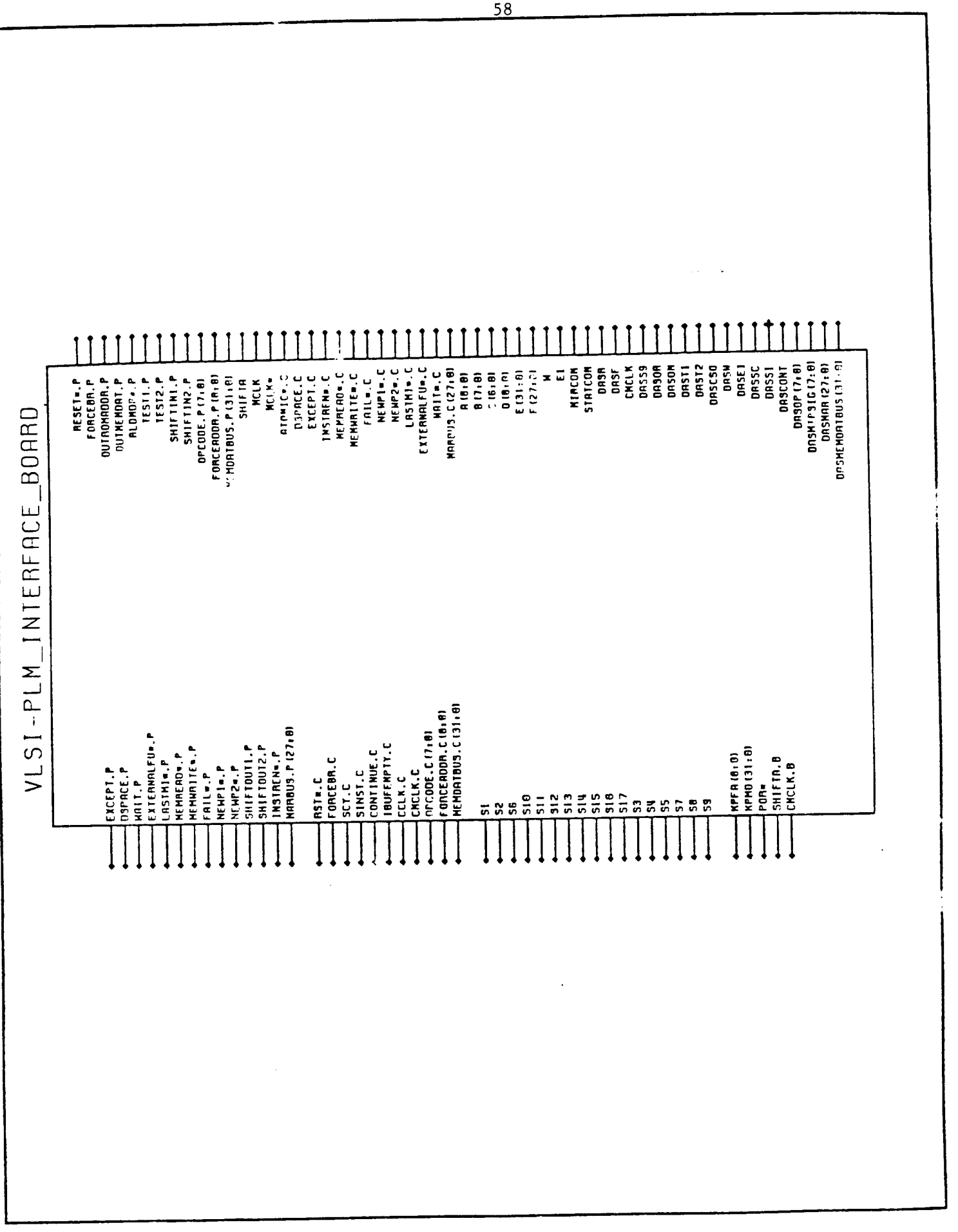


Figure 1 VLSI-PLM Board Symbol

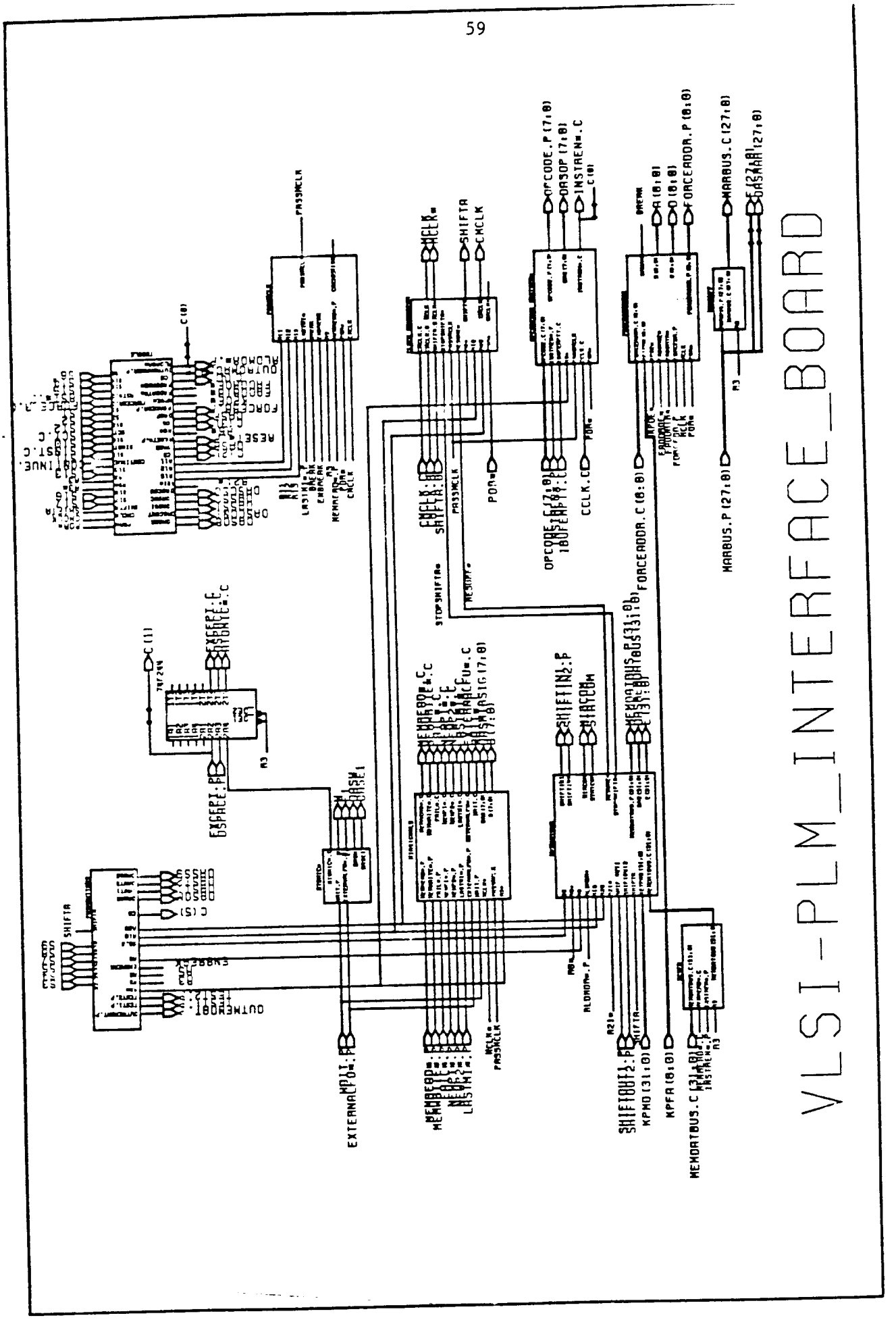


Figure 2 VLSI-PLM Board Schematic

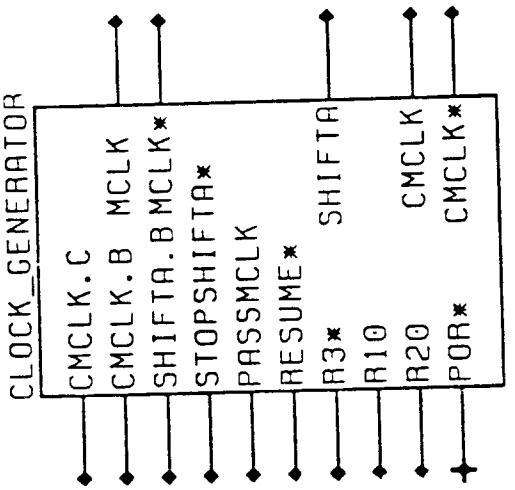


Figure 3 Clock Generator Symbol

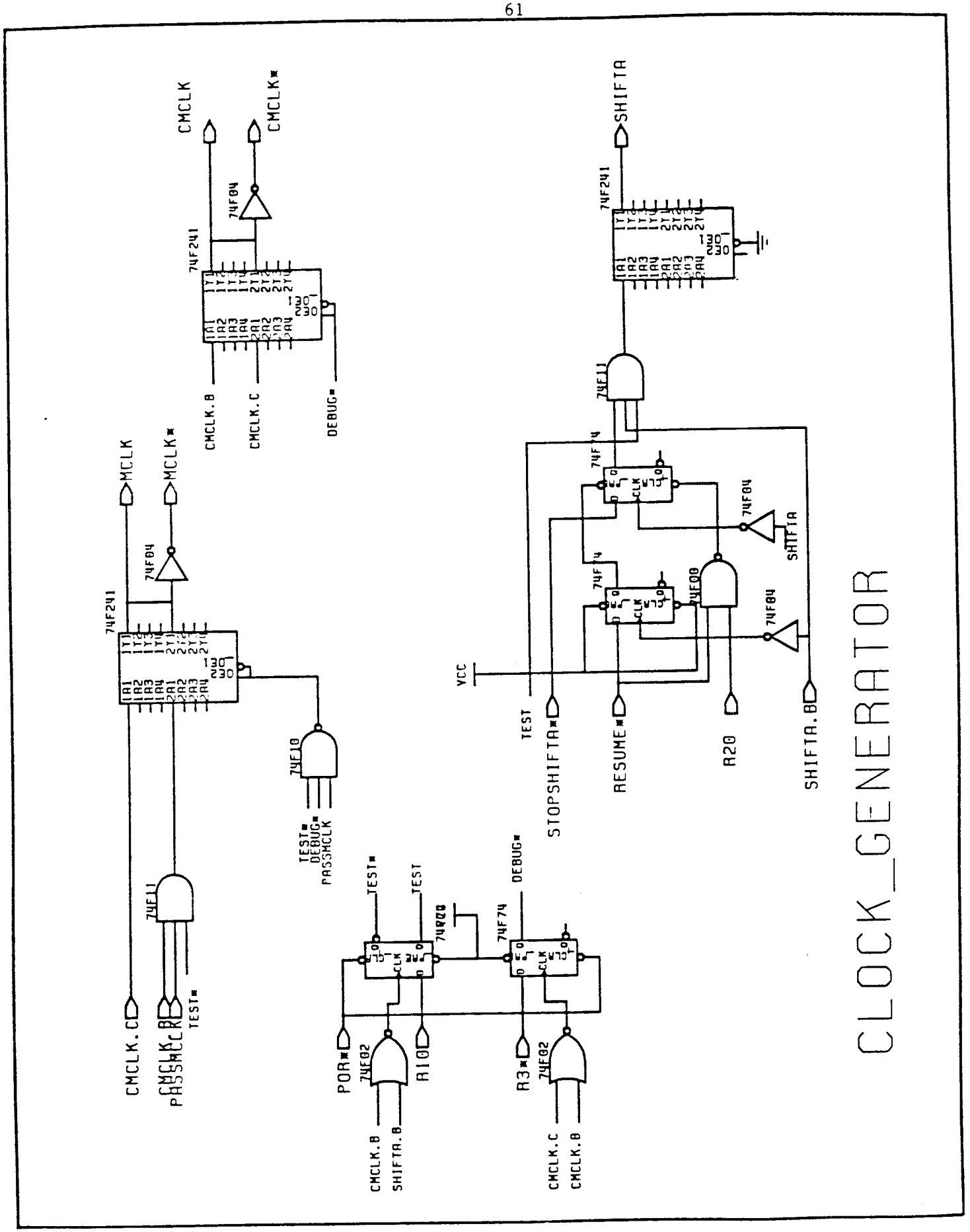


Figure 4 Clock Generator Schematic

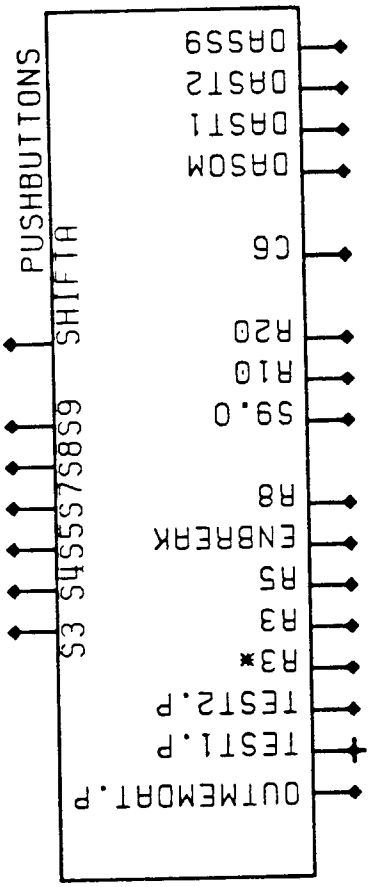


Figure 5 Pushbuttons Symbol

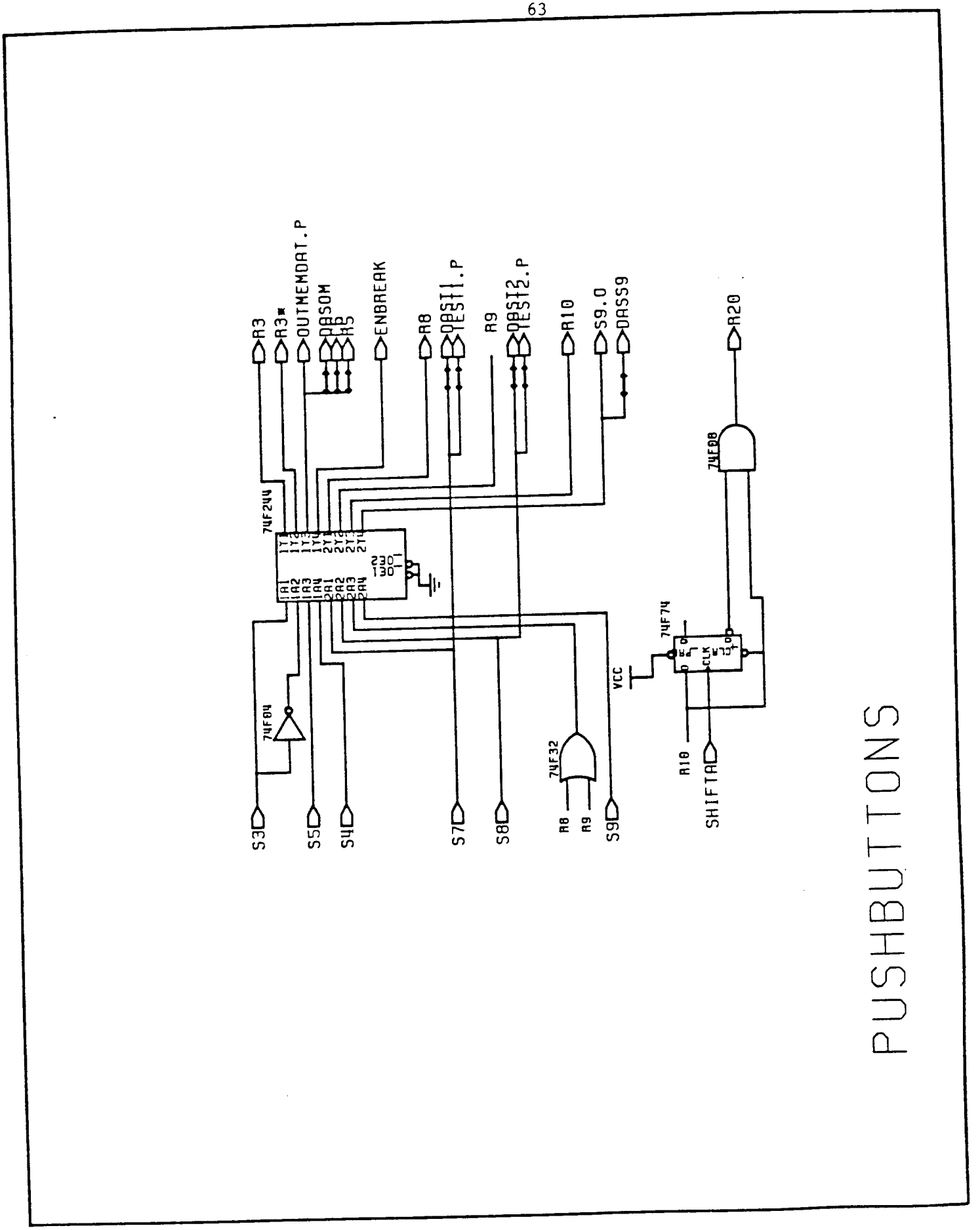


Figure 6 Pushbuttons Schematic

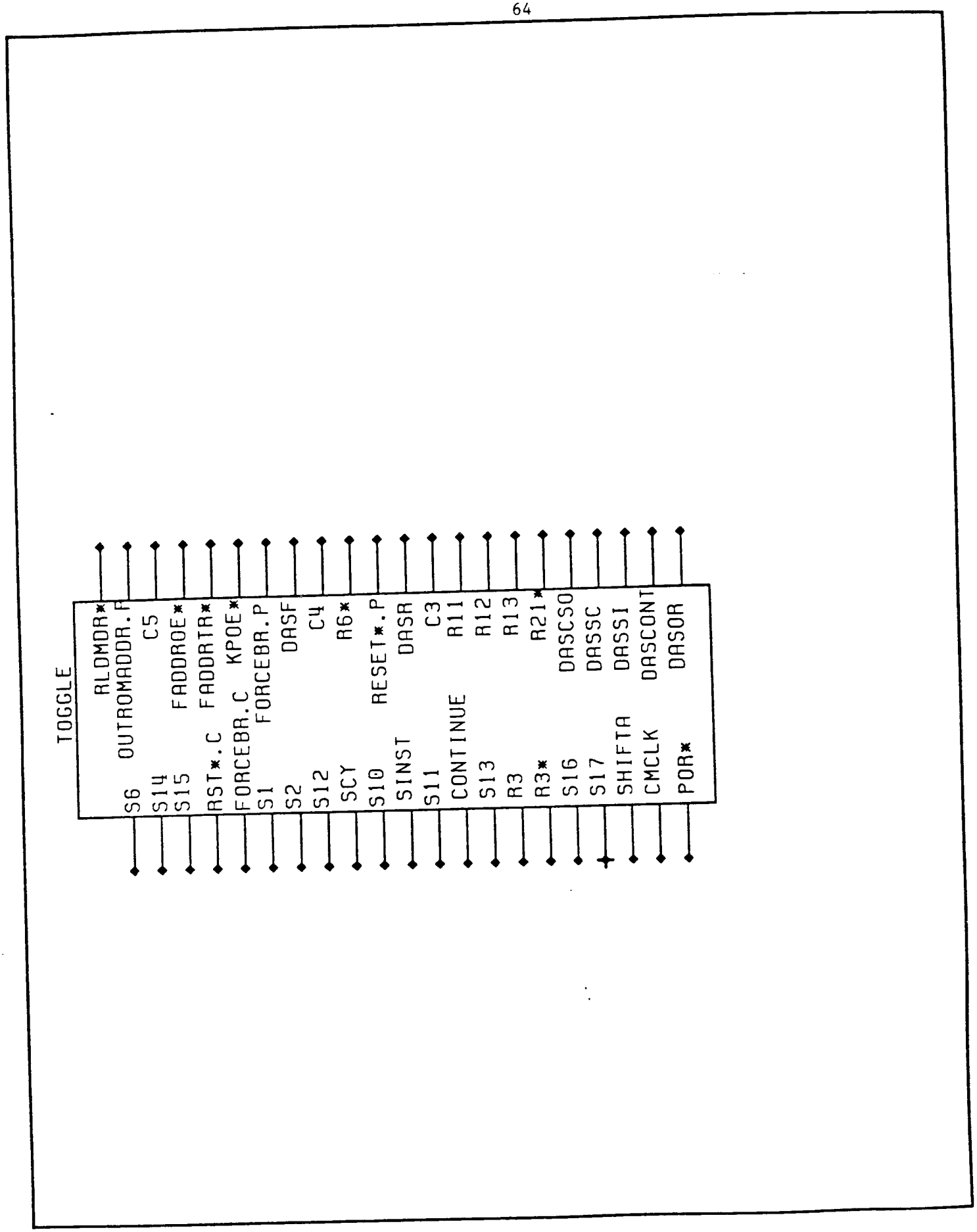


Figure 7 Toggle Symbol

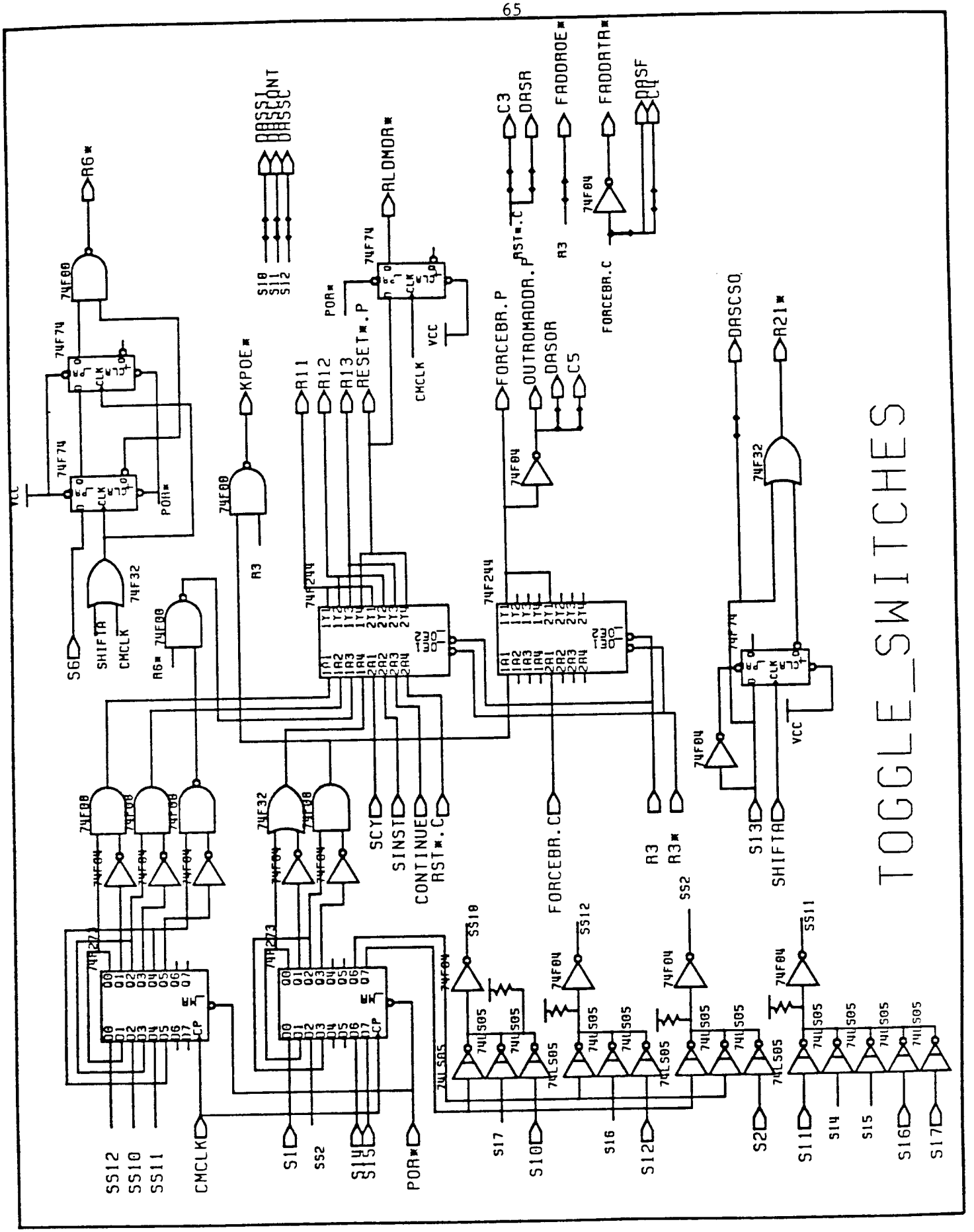


Figure 8 Toggle Schematic

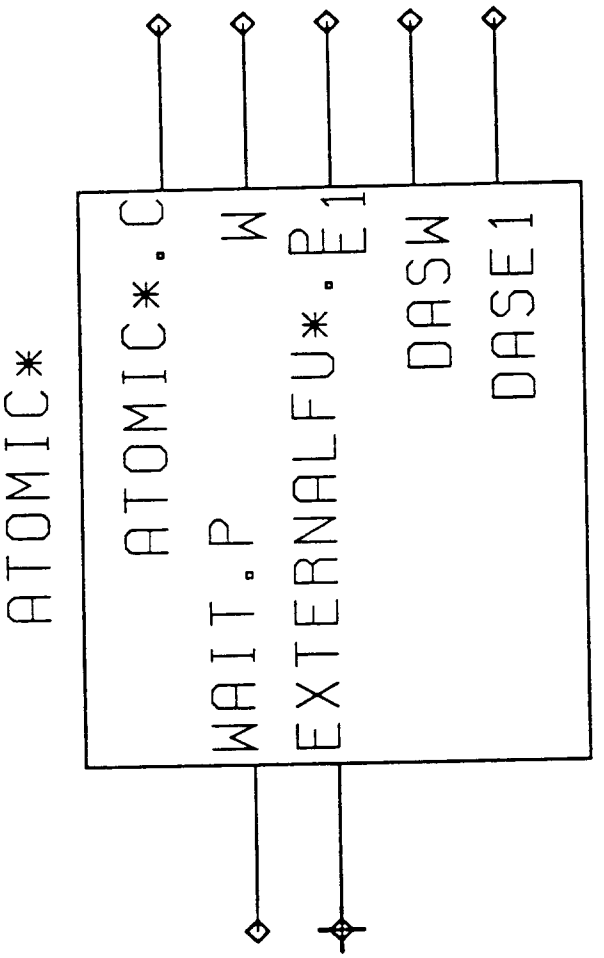


Figure 9 Atomicb Symbol

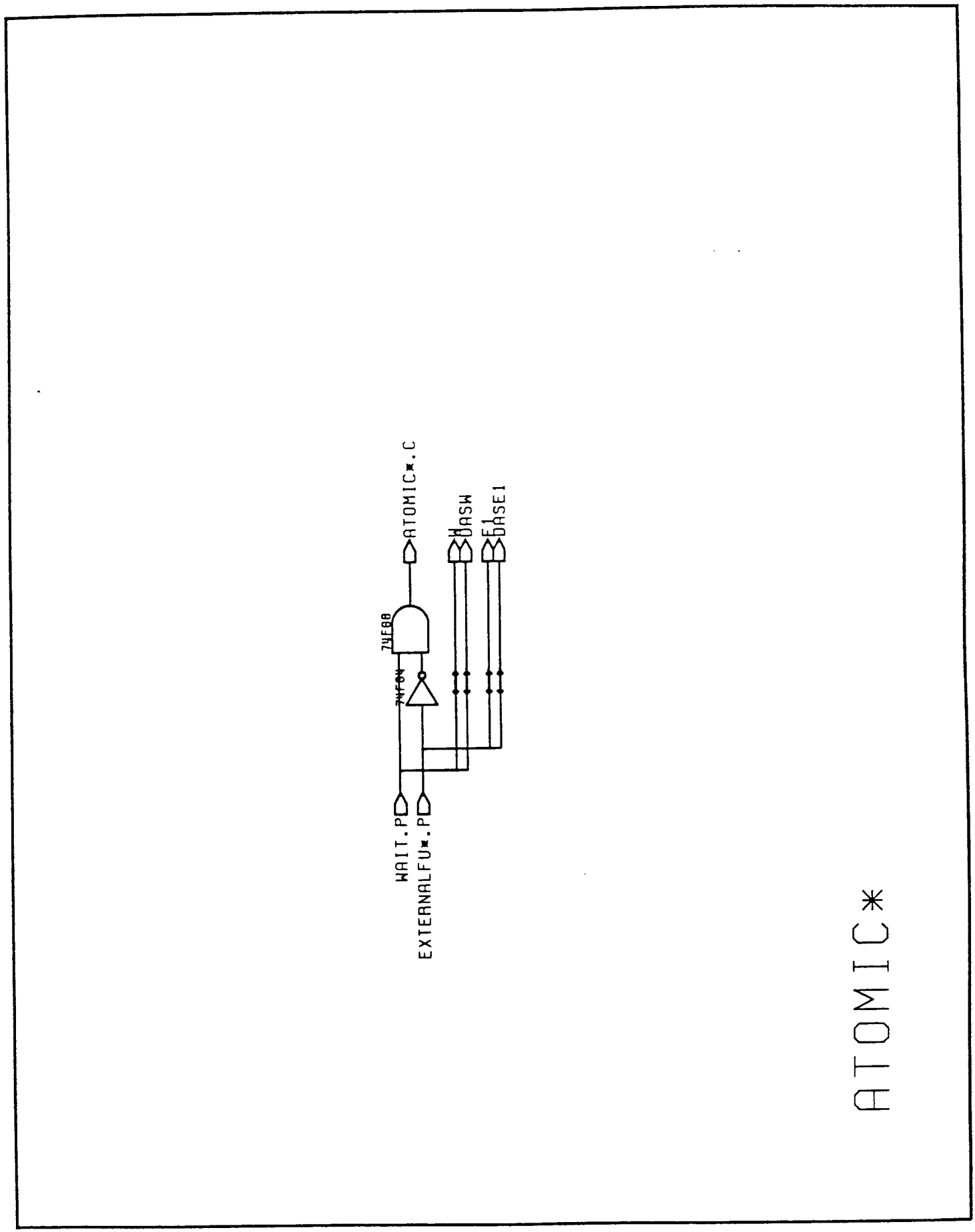


Figure 10 Atomicb Schematic

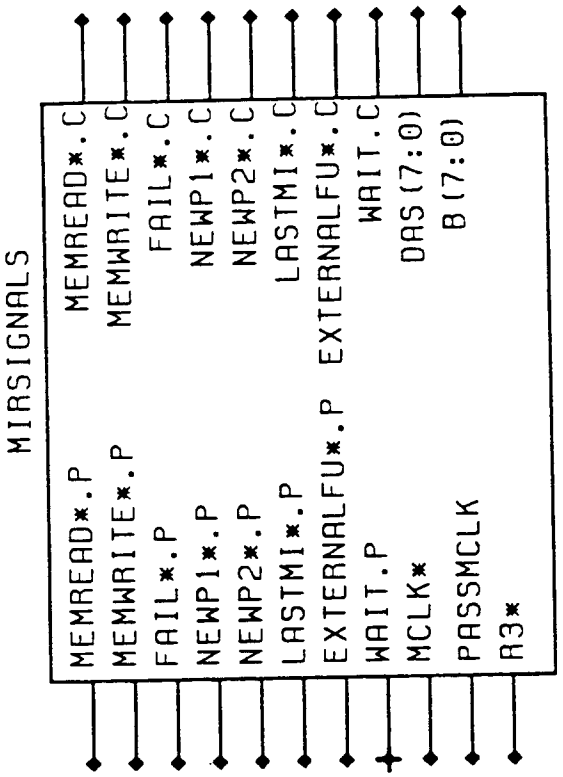


Figure 11 MIRsignal Symbol

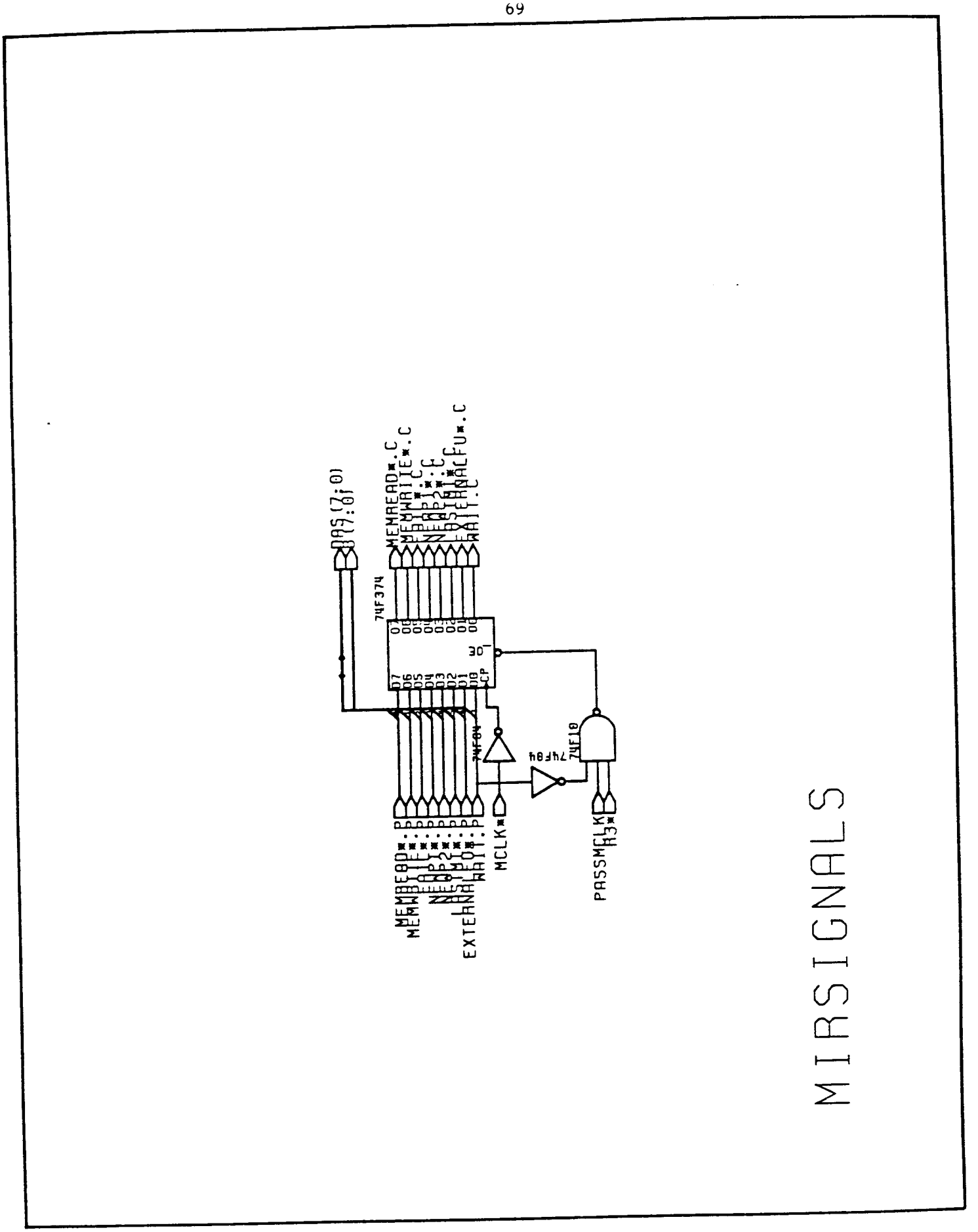


Figure 12 MIRsignal Schematic

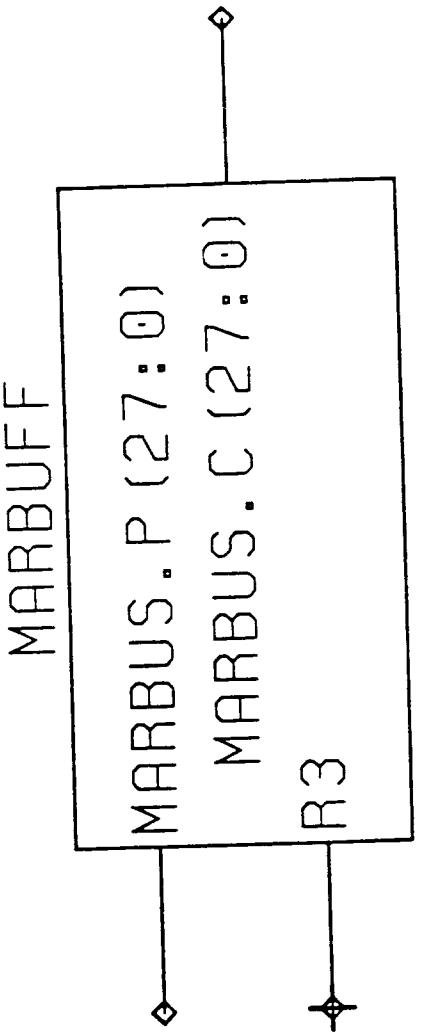


Figure 13 MARbuff Symbol

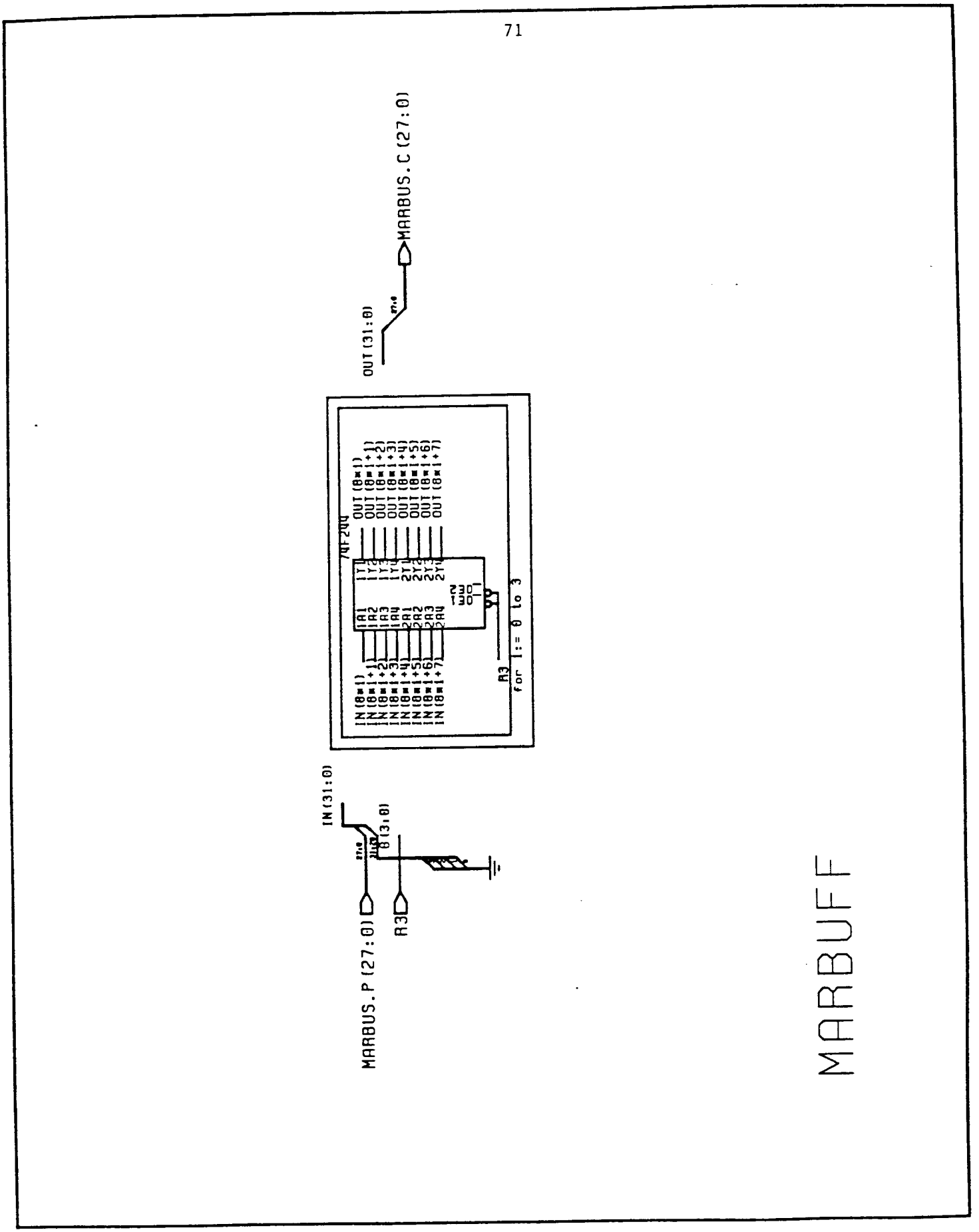


Figure 14 MARbuff Schematic

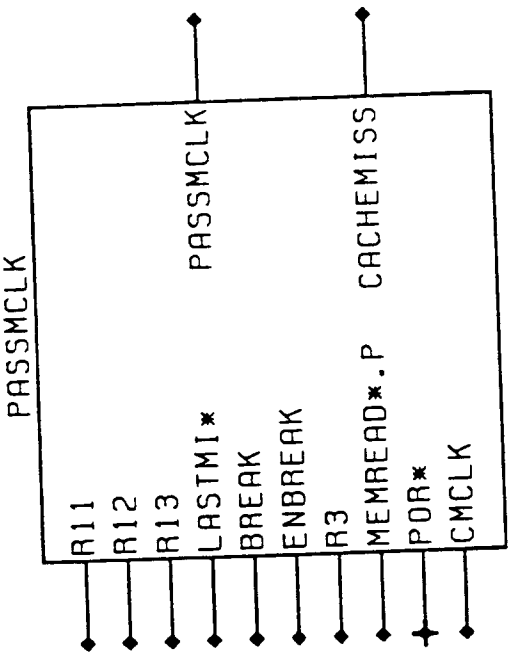


Figure 15 Passmclk Symbol

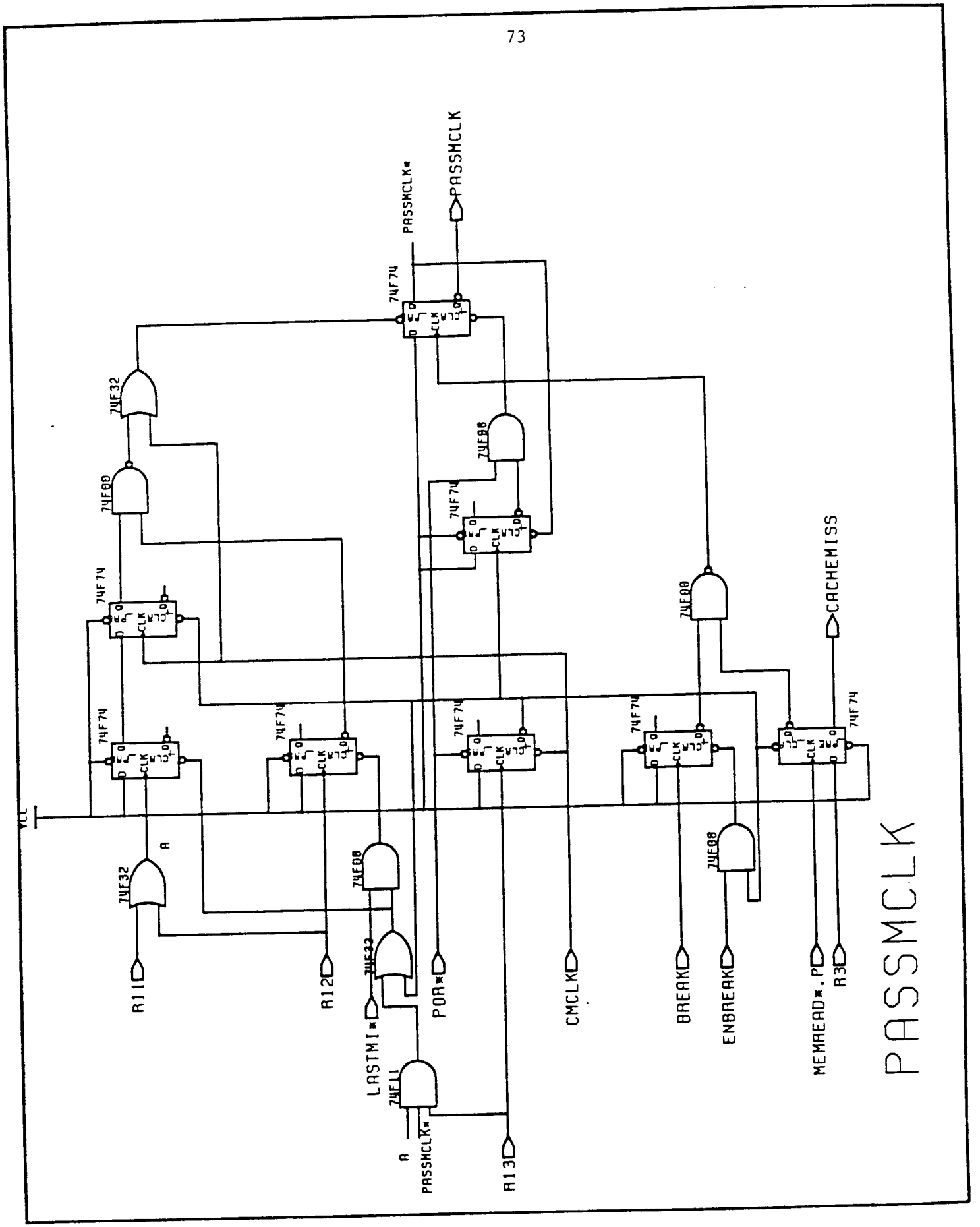


Figure 16 Passmclk Schematic

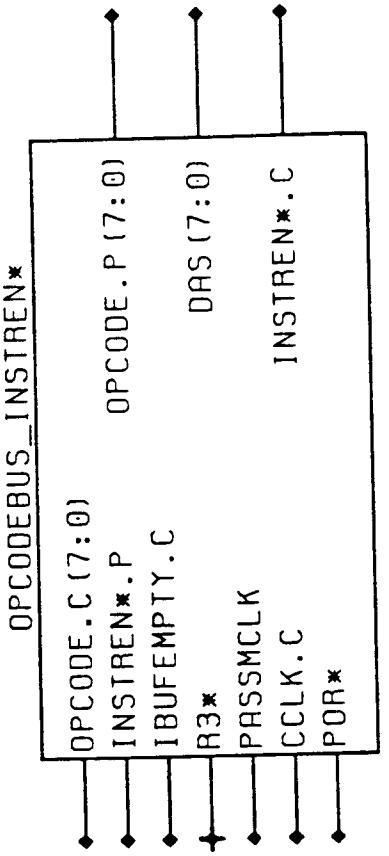


Figure 17 Opcodebus Symbol

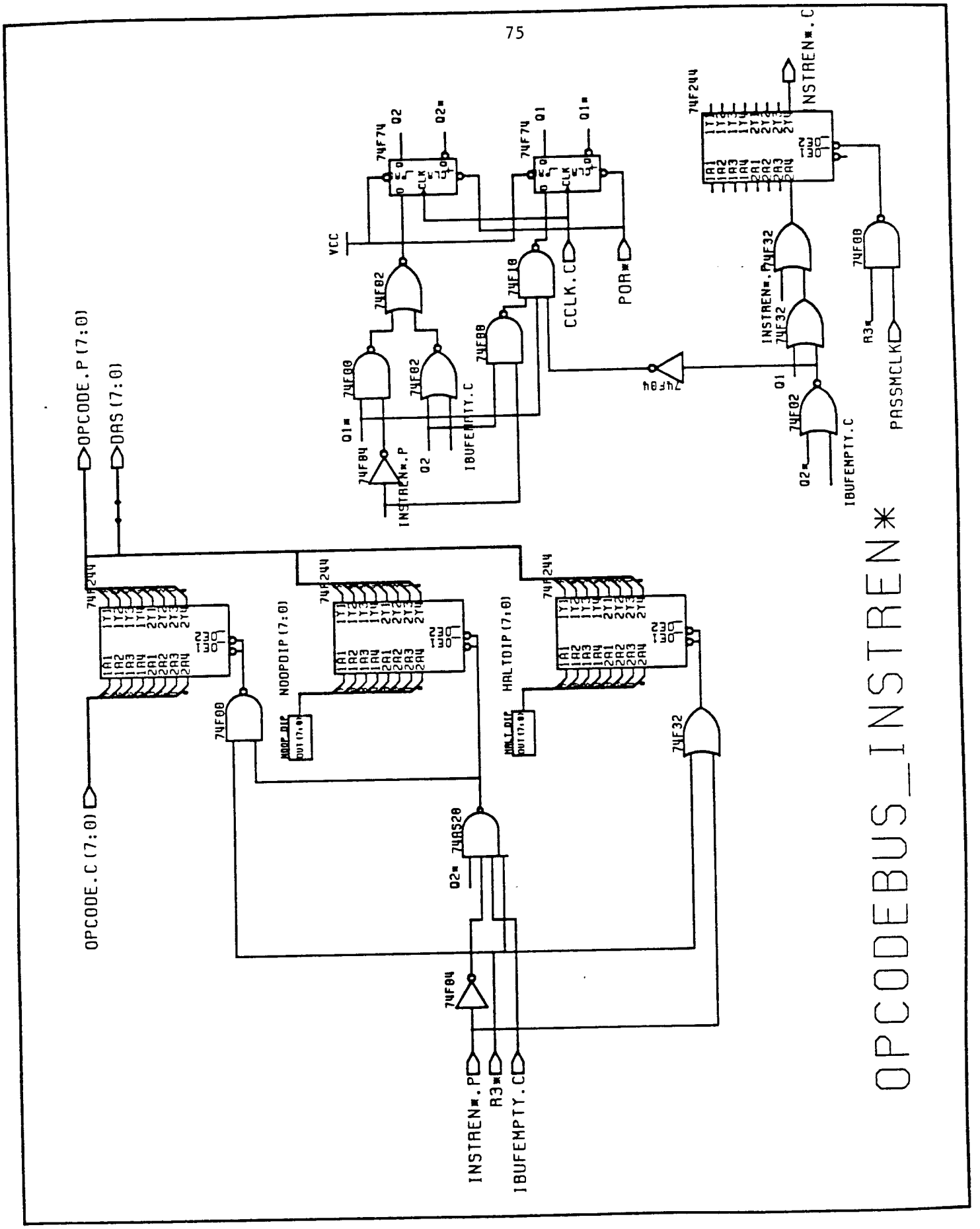


Figure 18 Opcodebus Schematic

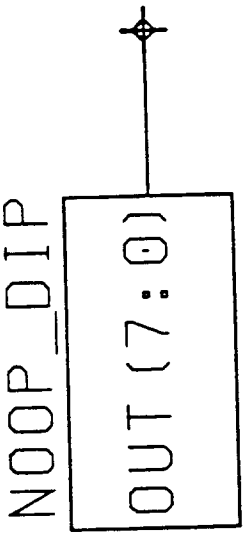
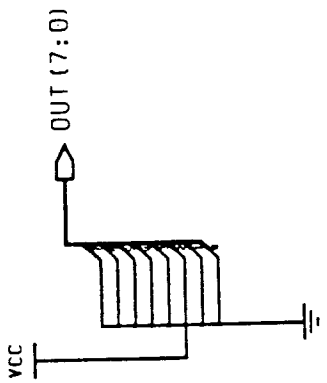


Figure 19 Noop_dip Symbol



NOOP_DIP

Figure 20 Noop_dip Schematic

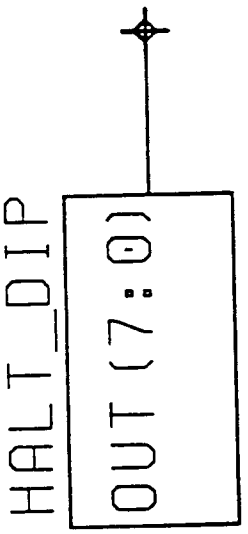
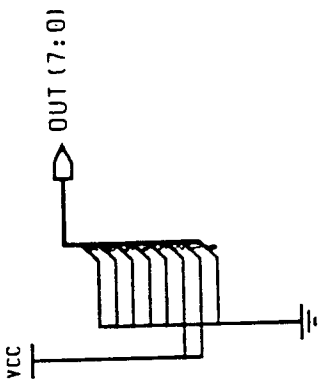


Figure 21 Halt_dip Symbol



HALT_DIP

Figure 22 Halt_dip Schematic

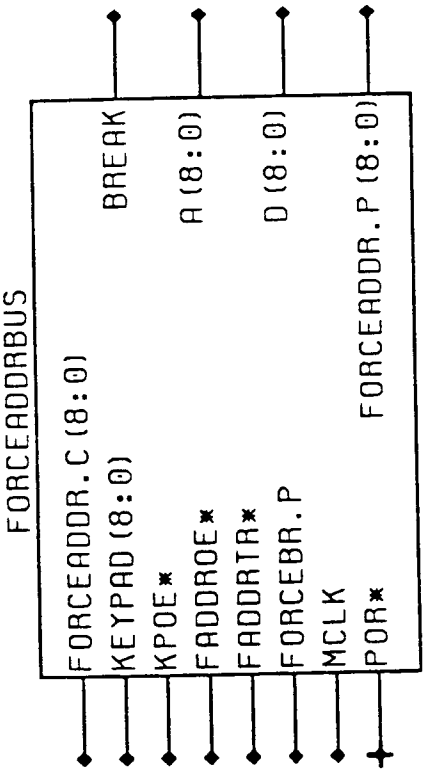


Figure 23 Forceaddr Symbol

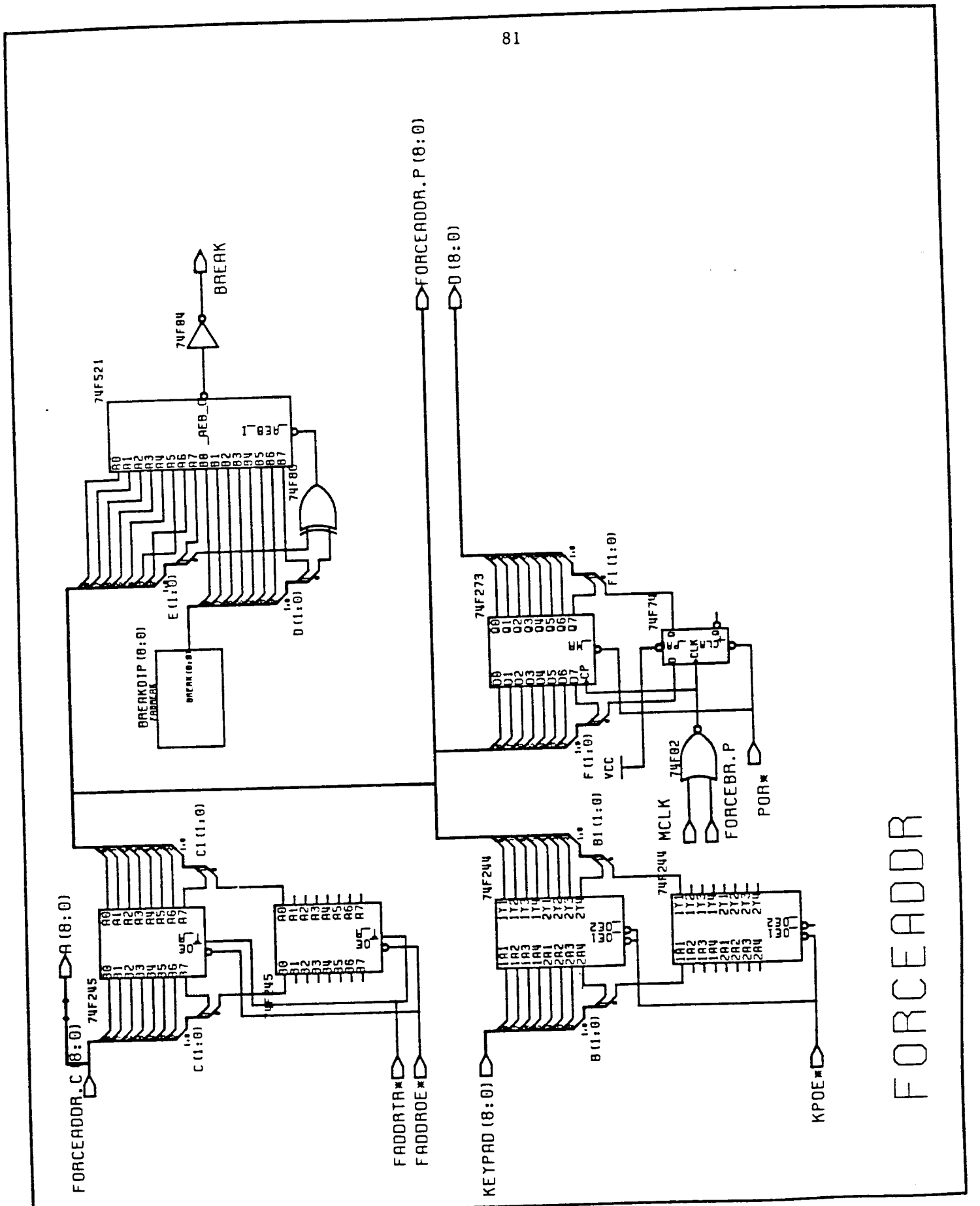


Figure 24 Forceaddr Schematic

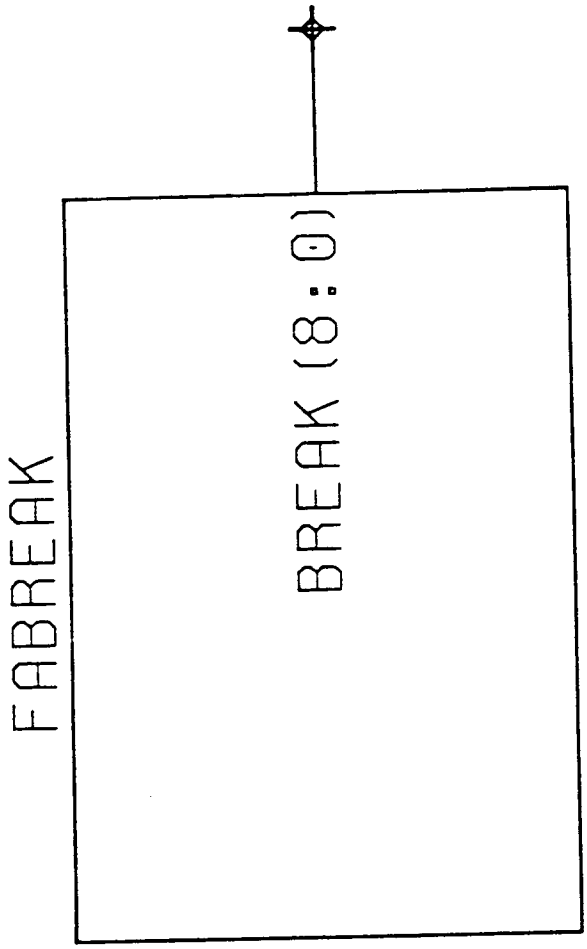
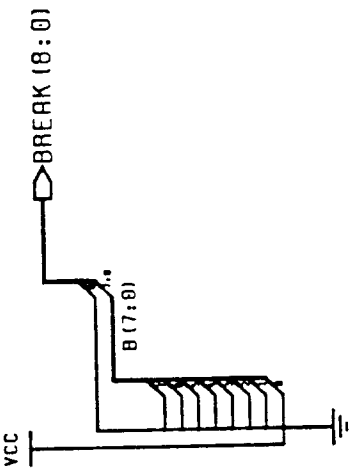


Figure 25 Fabreak Symbol



FABREAK

Figure 26 Fabreak Schematic

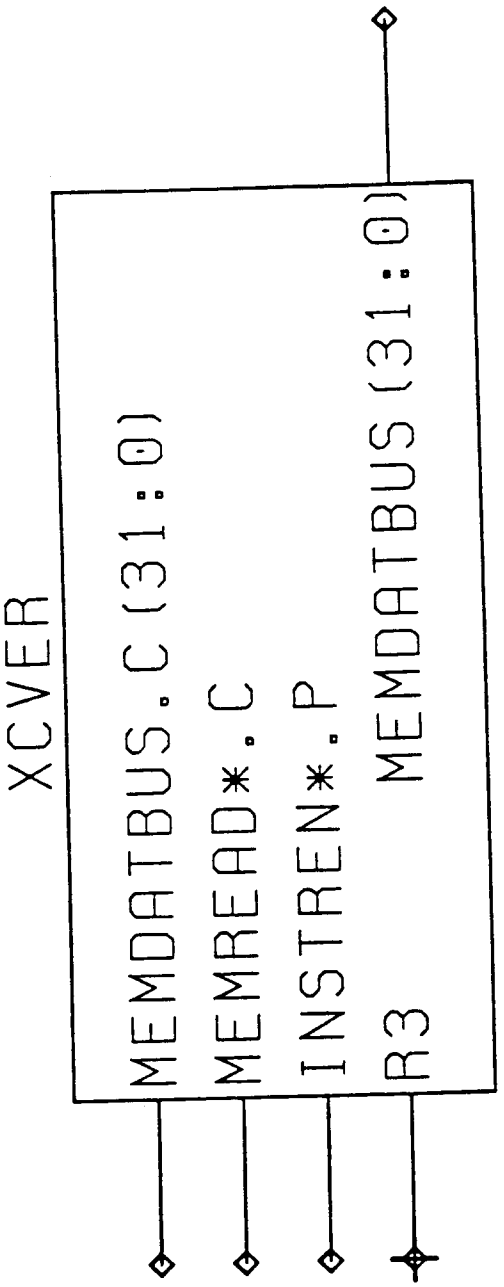


Figure 27 Xcver Symbol

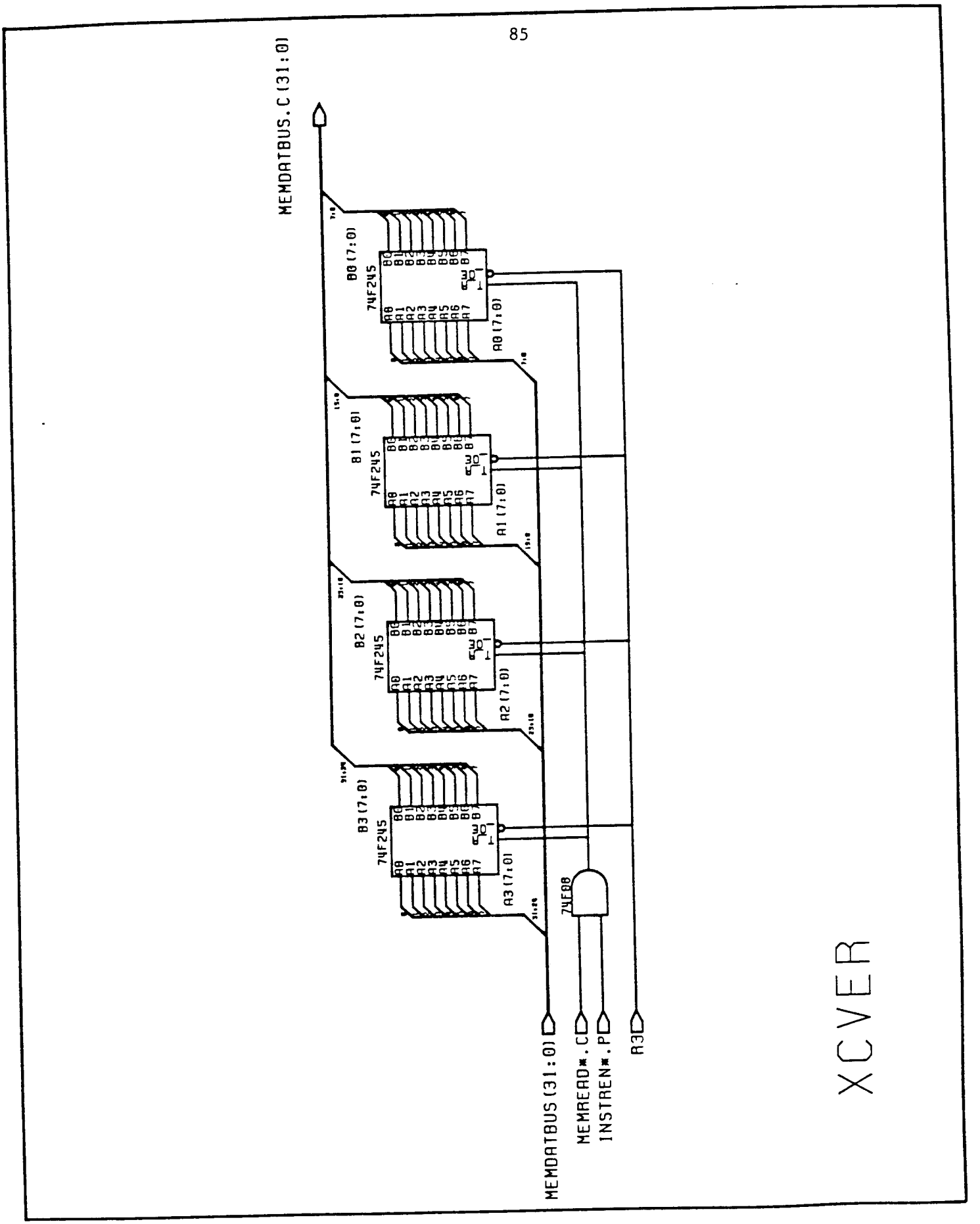


Figure 28 Xcver Schematic

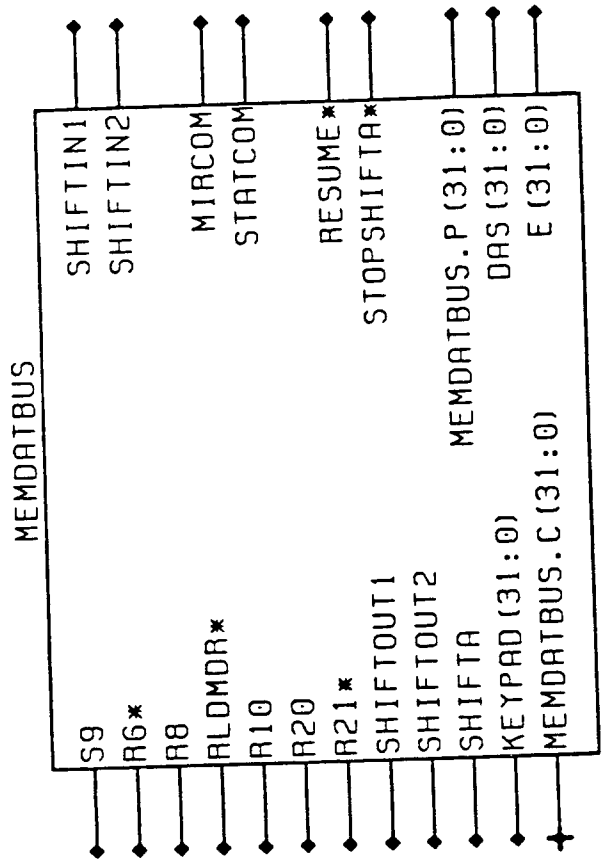


Figure 29 Memdatbus Symbol

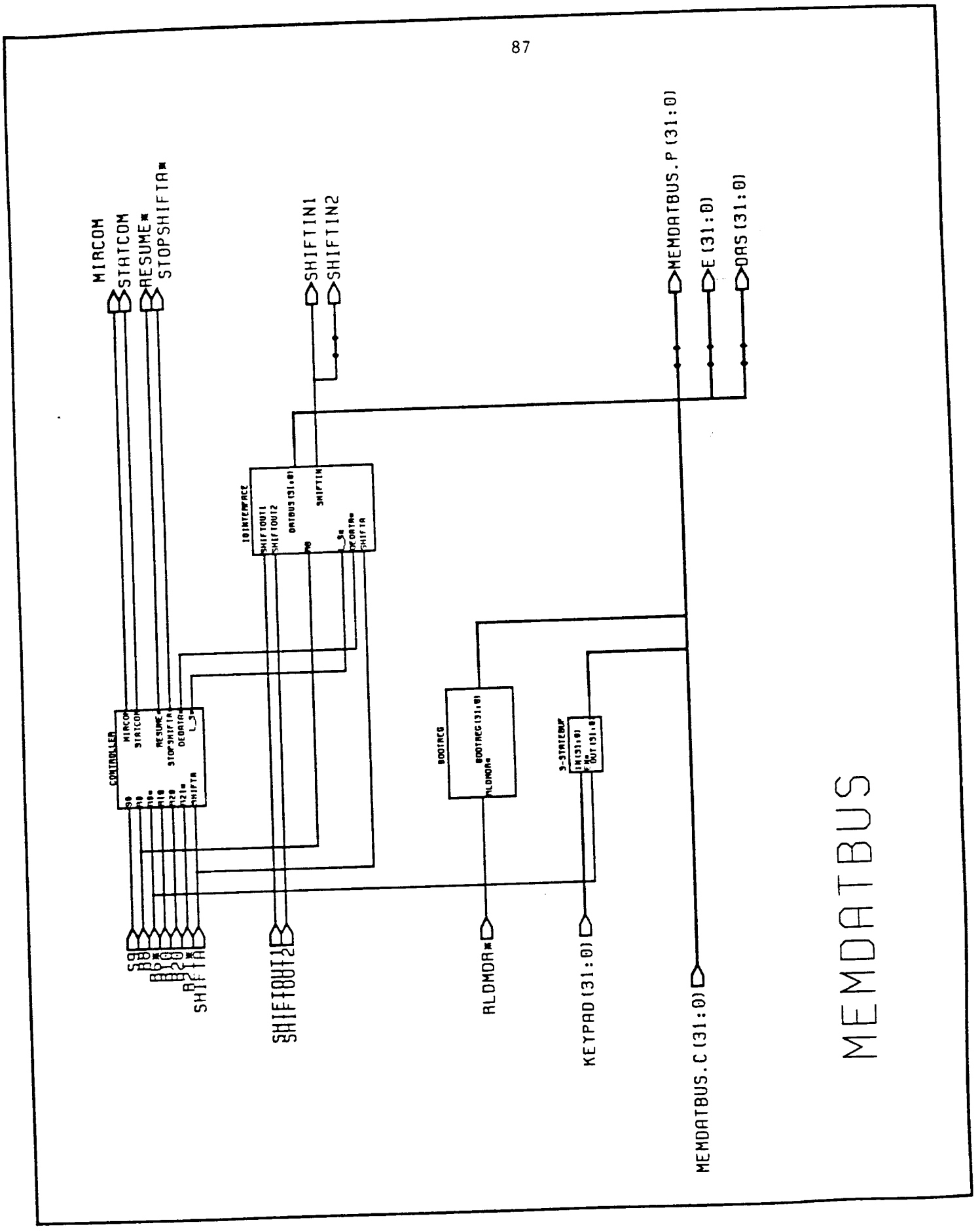


Figure 30 Memdatbus Schematic

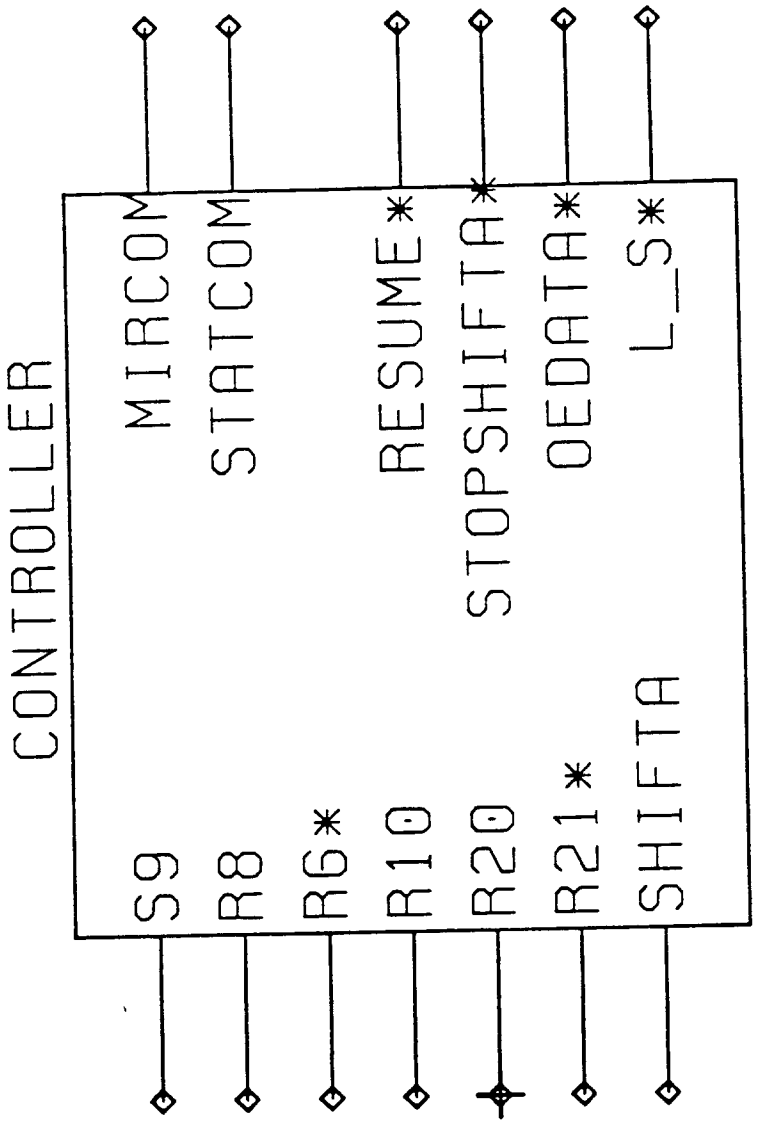


Figure 31 Controller Symbol

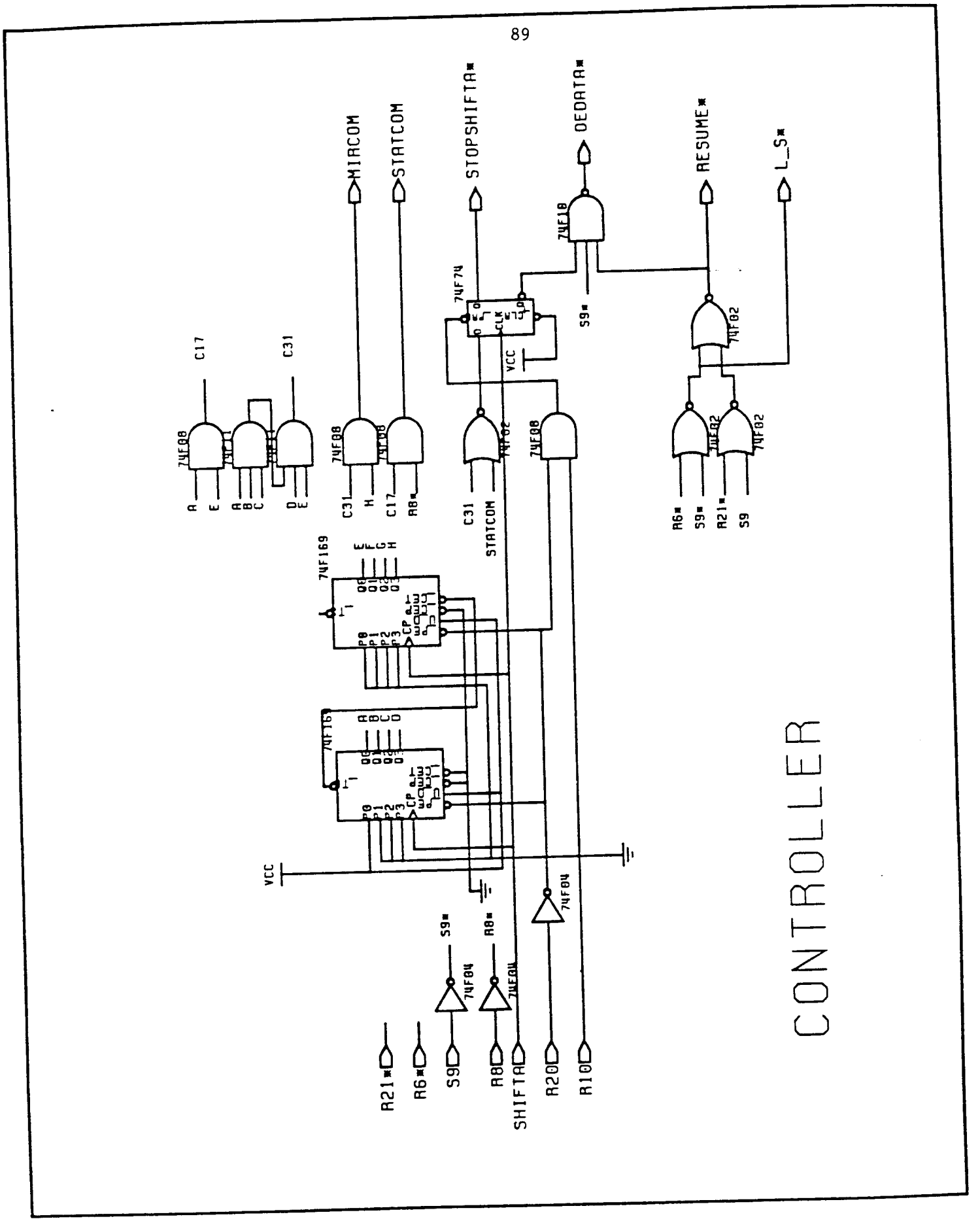


Figure 32 Controller Schematic

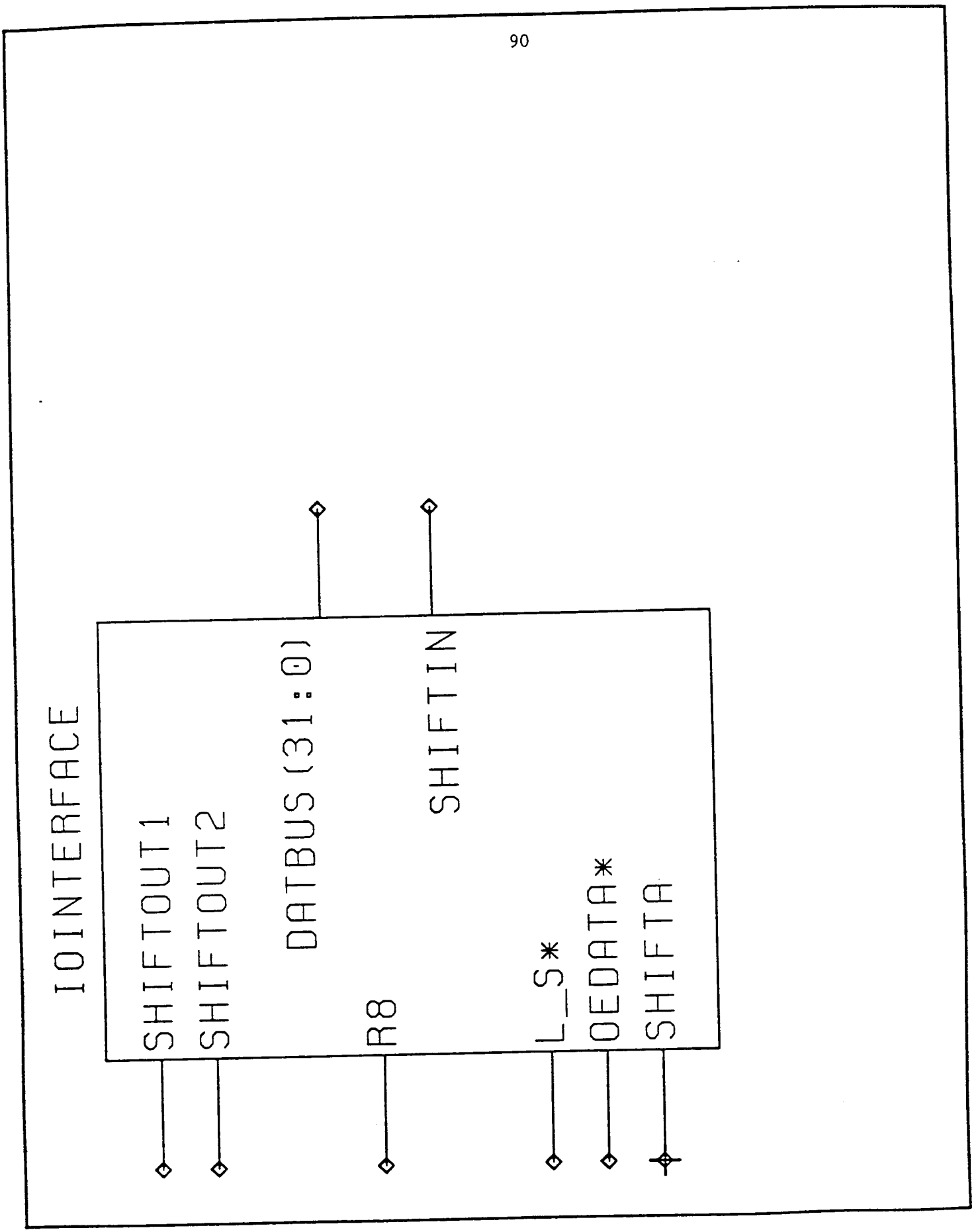


Figure 33 Iointerface Symbol

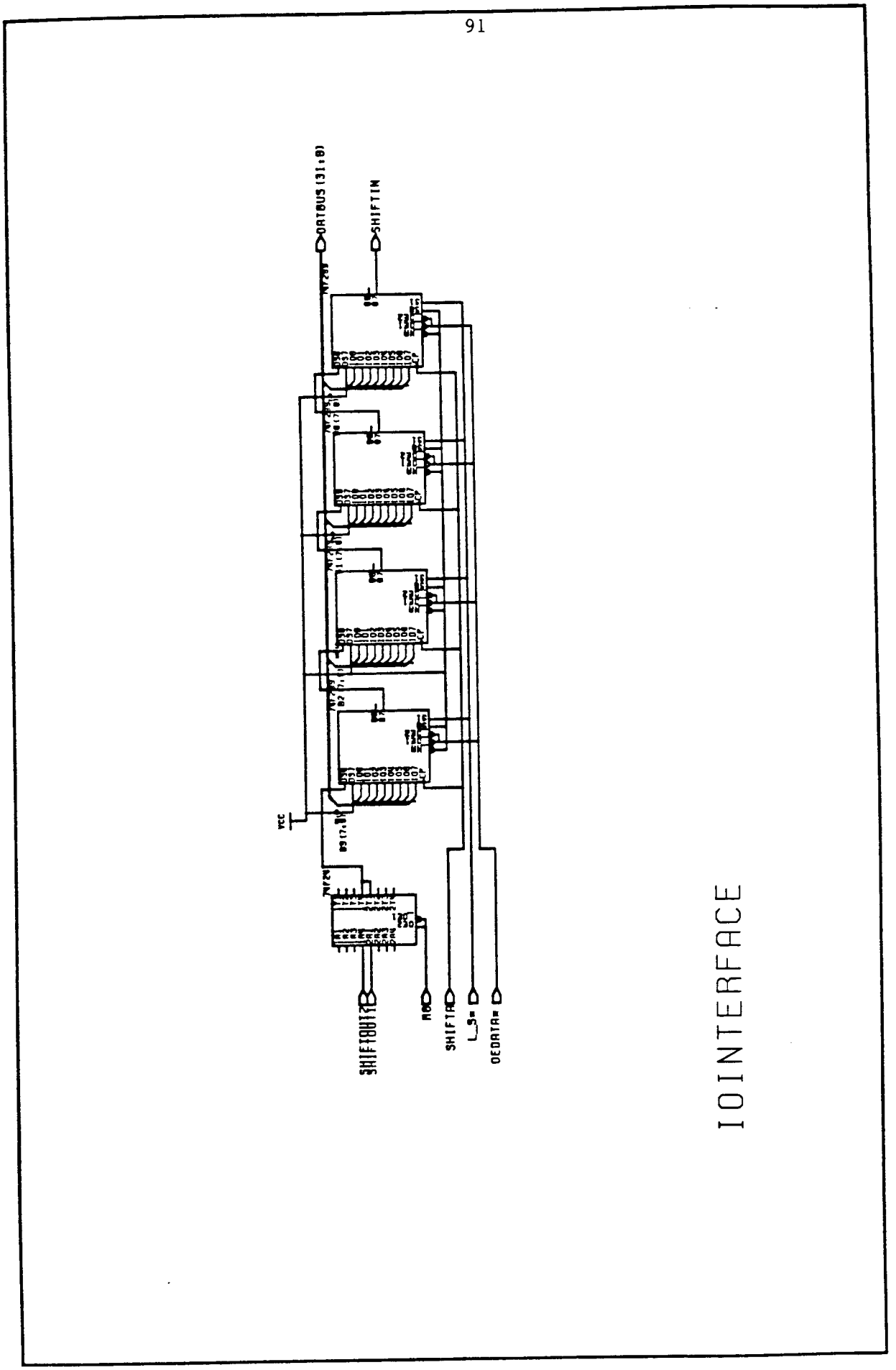


Figure 34 Iointerface Schematic

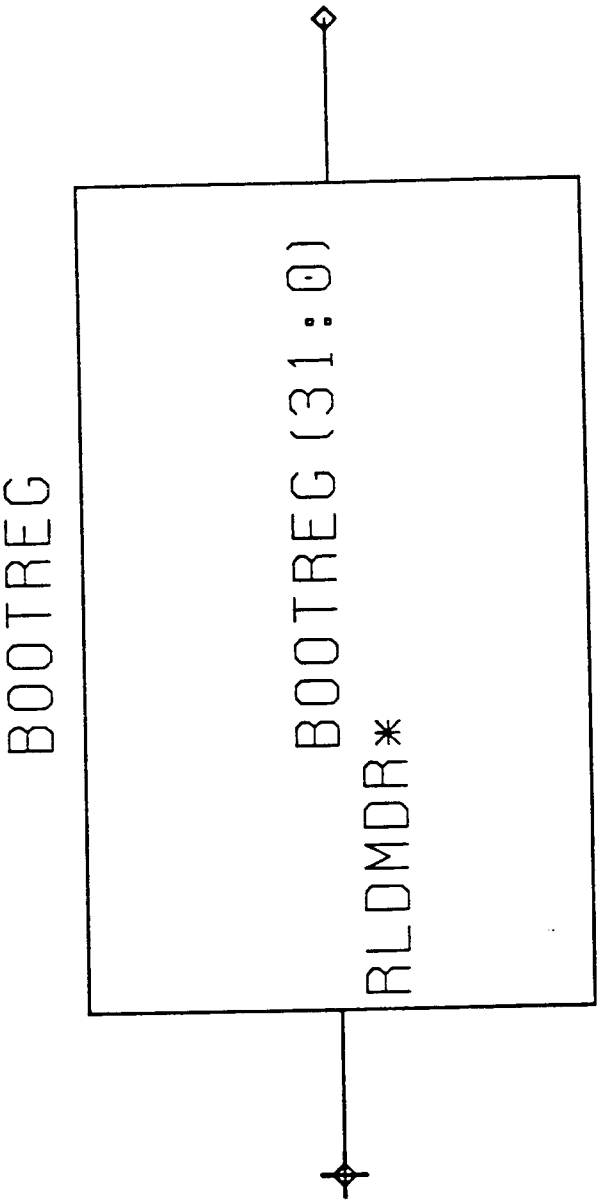


Figure 35 Bootreg Symbol

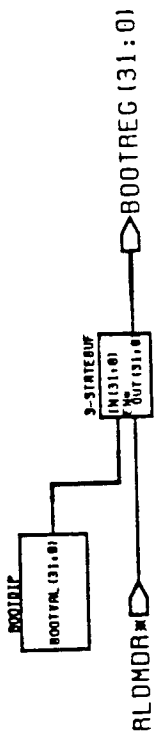


Figure 36 Bootreg Schematic

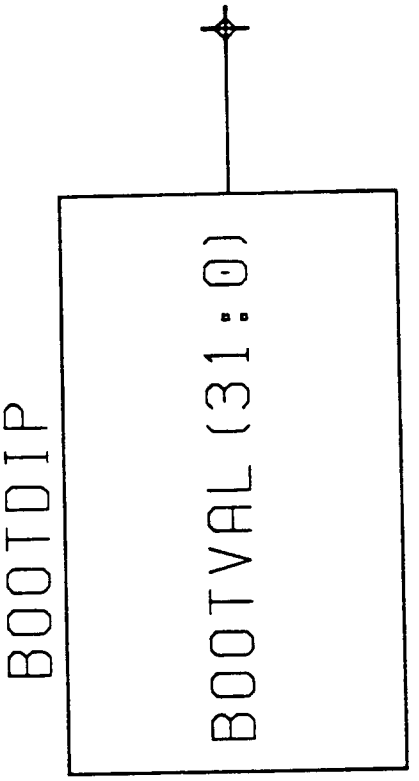


Figure 37 Bootdip Symbol

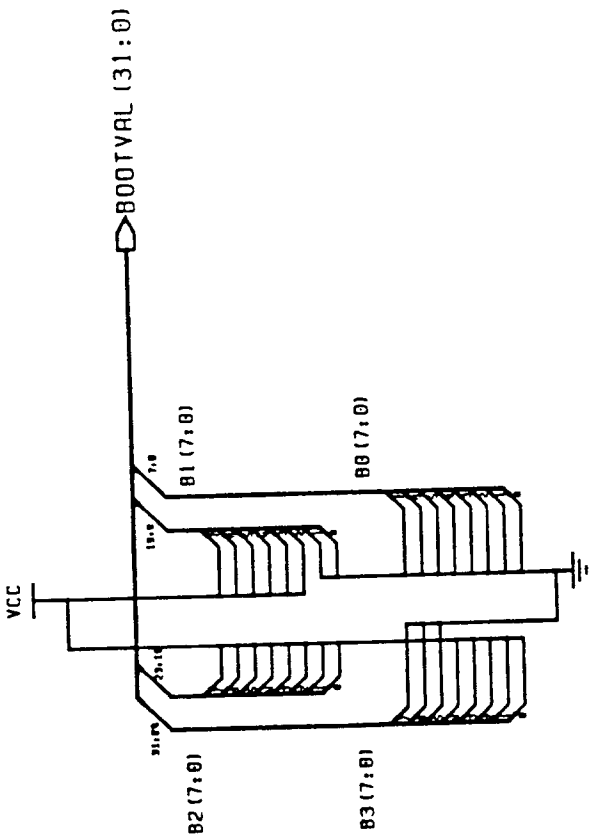


Figure 38 Bootdip Schematic

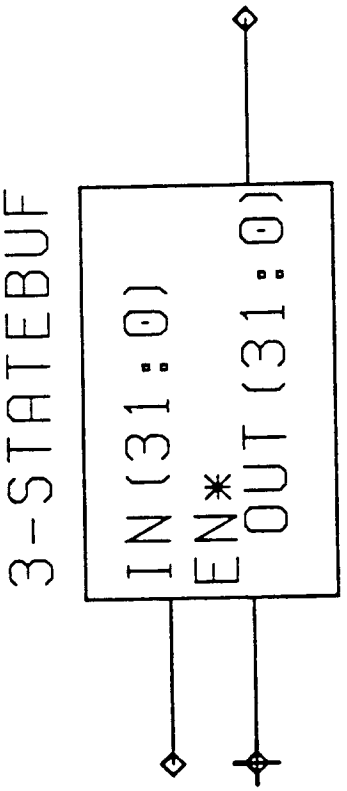


Figure 39 Tri-State Buffer Symbol

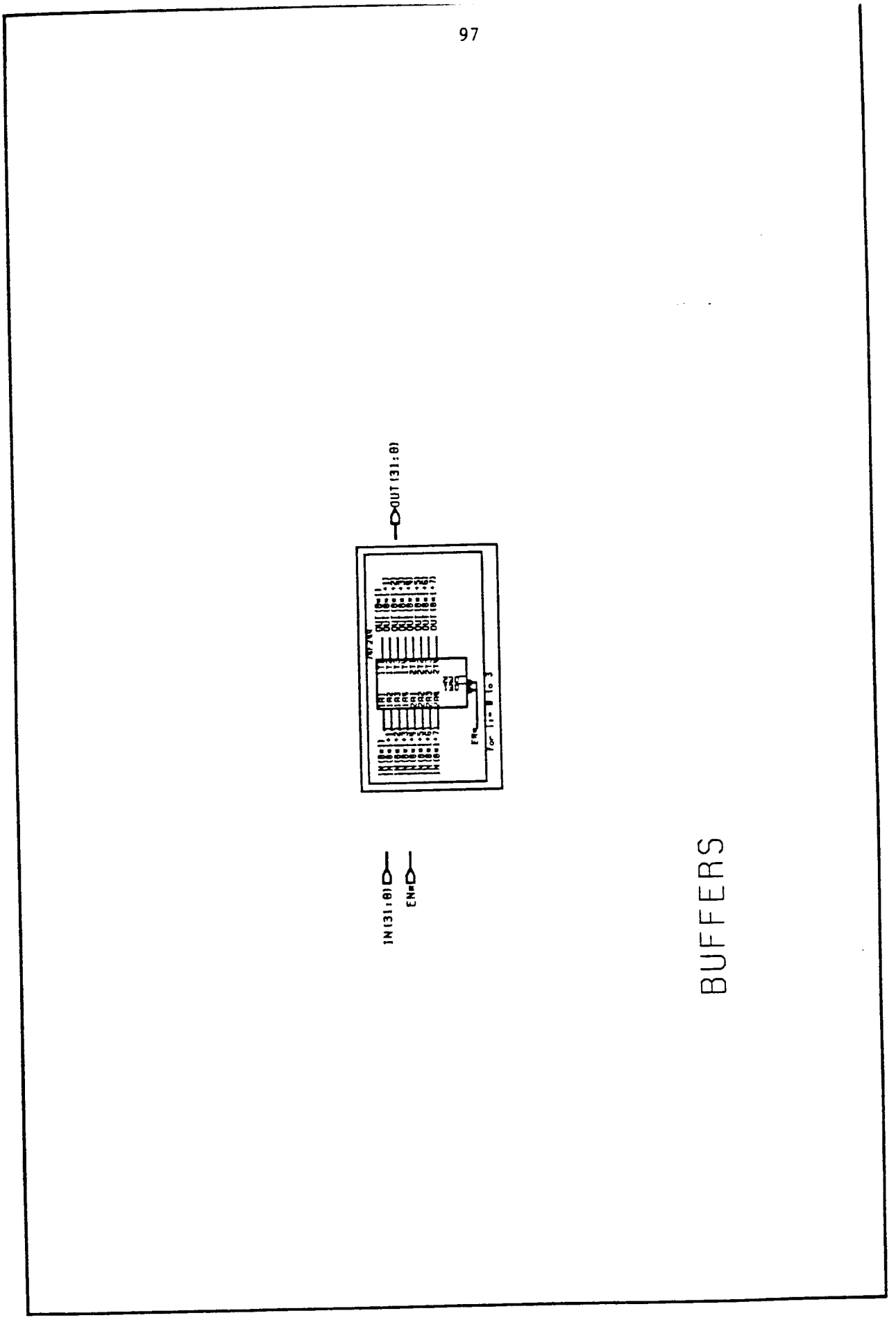
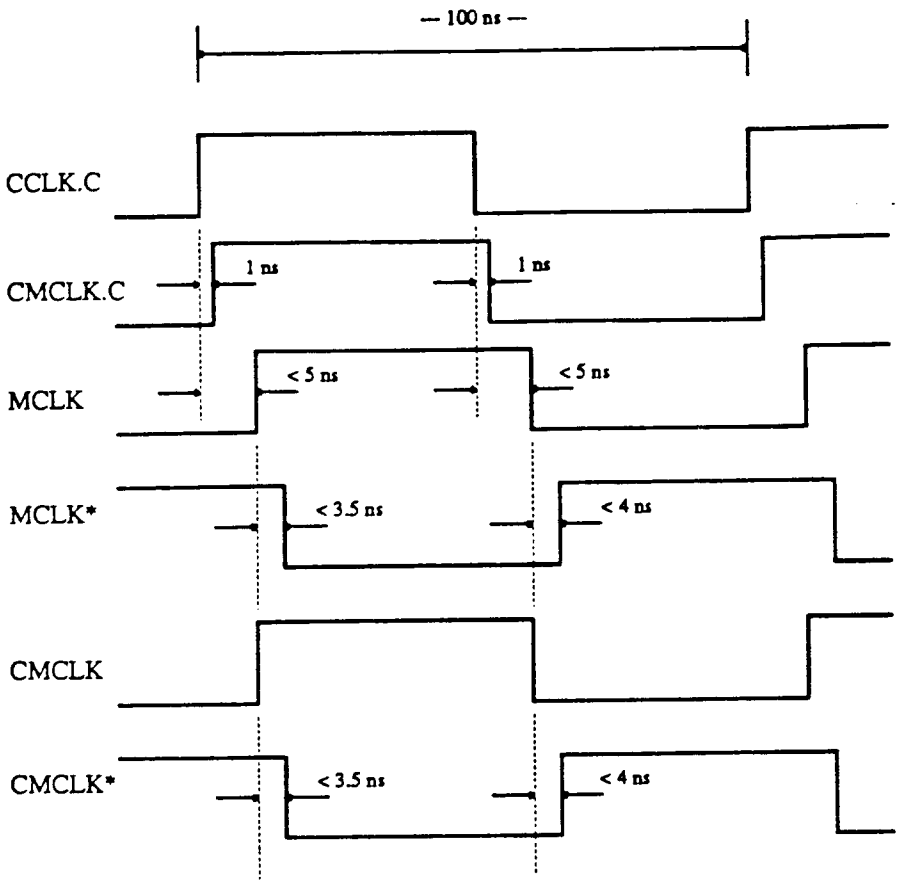


Figure 40 Tri-State Buffer Schematic

CLOCK SKEW



Notes

CCLK
- CB internal clock (always running)

CMCLK.C
- CB output clock
- Is called MCLK by Robert Yung

MCLK/MCLK*
- to VLSI-PLM
- is called PCLK by Robert Yung

CMCLK/CMCLK*
- used internally in VPB

Figure 41 Clock Skew

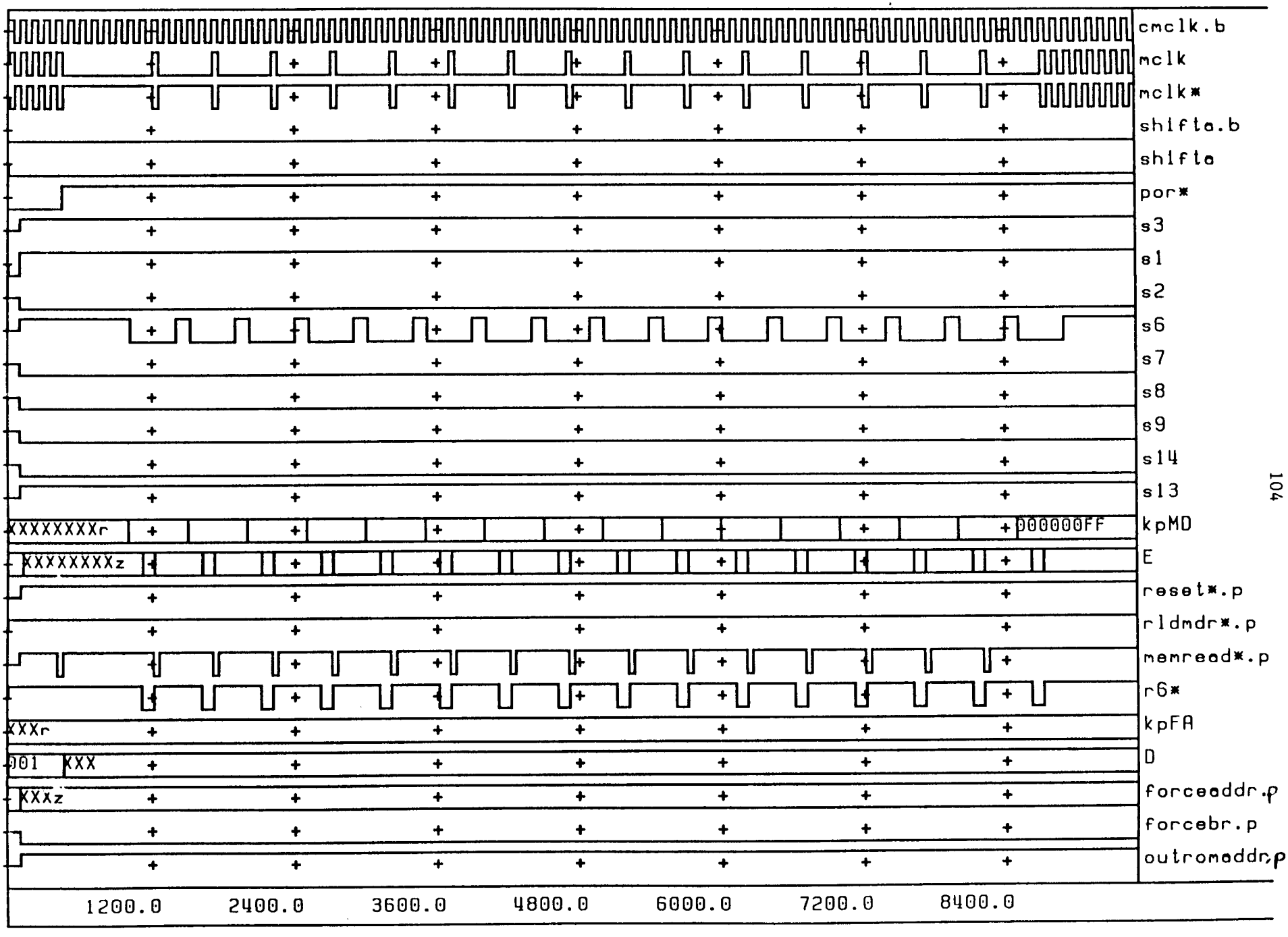
```
# RESET1.do simulation input for testing of the VPB
#
# Reset sequence:
# CMCLK.B is at 10 MHz.
#
force POR* 0 0
force POR* 1 450

# Reset the switches:
# All switches are initialized so that we are in debug mode.
#
force s3 1 100
force r3* 0 100 -f
force s4 0 100
force s5 1 100
force s7 0 100
force s8 0 100
force s9 0 100
force s1 1 100
force s2 0 100
force s6 1 100
force s10 0 100
force s11 0 100
force s12 0 100
force s13 1 100
force s14 0 100
force s15 0 100
force s16 0 100
force s17 0 100

# Set miscellaneous signals:
# VLSI-PLM MEMREAD*.P is initially high.
#
force memread*.p 1 100
force lastmi*.p 1 100
force instren*.p 1 100

# Set the clocks:
# The board is receiving CMCLK.C from the cache board.
# The offset between CMCLK.C and CMCLK.B is just to show
#      that they are not synchronized.
#
clo p 100
force cmclk.c 1 10 -r
force cmclk.c 0 60 -r

# Set on board clocks:
# To save hardware, we use the same clock for MCLK.B and SHIFTA.B.
# This is possible since they are not on at the same time.
# Initially SHIFTA.B is 0 and CMCLK.B is 10 MHz.
#
clo p 100
force cmclk.b 1 0 -r
force cmclk.b 0 50 -r
force shifta.b 0 0

# Hard reset:
# During cycle 1, we press S1.
# In cycle 2, the board will send RESET* to VLSI-PLM,
#      which branches to location 0x077.
# In cycle 3, the board sends RLDMDR* to VLSI-PLM, enables
#     MEMDAT, and VLSI-PLM starts executing boot00.
# VLSI-PLM executes boot01 in cycle 4
# VLSI-PLM executes init00 in cycle 5 and waits
#      for user to enter constant RAM data
# Note lines in caps indicate VLSI-PLM outputs
```

```
#
force s1 0 10
force s1 1 300
FORCE MEMREAD*.P 0 410
FORCE MEMREAD*.P 1 460
run 1000

# Supply data for INIT00
#
force kpmd 0ffffc00 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT01
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT01
#
force kpmd 0ffffc10 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT02
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT02
#
force kpmd 0fffffff 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT03
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT03
#
force kpmd 0ffffe00 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT04
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500
```

```
# Supply data for INIT04
#
force kpmd 00000001 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT05
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT05
#
force kpmd 00000020 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT06
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT06
#
force kpmd  00001000 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT07
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT07
#
force kpmd cfffffff 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT08
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT08
#
force kpmd 00040000 0

# Press ENTER* switch
```

```
#
force s6 0 20
force s6 1 400

# Start INIT09
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT09
#
force kpmd 00080000 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT10
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT10
#
force kpmd 00000004 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT11
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT11
#
force kpmd 0000000f 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT12
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT12
#
force kpmd 00000002 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT13
#
```

```
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT13
#
force kpmd 00000000 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT14
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT14
#
force kpmd fffffffd 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400

# Start INIT15
#
FORCE MEMREAD*.P 0 210
FORCE MEMREAD*.P 1 260
run 500

# Supply data for INIT15
#
force kpmd  000000ff 0

# Press ENTER* switch
#
force s6 0 20
force s6 1 400
run 500

# Run some more cycles to make sure that VLSI-PLM
#      can finish with the reset states.
# VLSI-PLM should halt after it is done with the
#      Hard Reset sequence.
#
run 500

# Save state and list window
#
save state all/reset1 -r
write list reset1.list -r
```



Trace of Simulation Results of Reset1 test

```
  0.0 Xr 1 0 Xi X Xi X 0 Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
  4.6 Xr 1 0 Xi X Xi X 0 Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
  7.0 Xr 1 0 Xi X Xi X 0 Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
  8.1 Xr 1 0 Xi 0 Xi X 0 Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 10.0 1  1 0 Xi 0 Xi X 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 10.7 1  1 0 Xi 0 Xi X 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 12.4 1  1 0 1  0 Xi X 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 16.8 1  1 0 1  0 1  X 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 20.0 1  1 0 1  0 1  0 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 50.0 1  0 0 1  0 1  0 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 54.0 1  0 0 0  0 1  0 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 58.1 1  0 0 0  0 0  0 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 60.0 0  0 0 0  0 0  0 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
 61.8 0  0 0 0  0 0  1 0 Xr Xr Xr Xr 0  Xr Xr Xr Xr Xr Xr Xr Xr XXXXXXXXr XXXXXX
100.0 0  1 0 0  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
104.0 0  1 0 1  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
105.4 0  1 0 1  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
108.2 0  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
109.0 0  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
109.4 0  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
110.0 1  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
111.4 1  1 0 1  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
113.1 1  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
131.2 1  1 0 1  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
150.0 1  0 0 1  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
154.0 1  0 0 0  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
158.1 1  0 0 0  0 0  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
160.0 0  0 0 0  0 0  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
161.8 0  0 0 0  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
200.0 0  1 0 0  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
204.0 0  1 0 1  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
208.2 0  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
210.0 1  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
211.4 1  1 0 1  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
250.0 1  0 0 1  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
254.0 1  0 0 0  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
258.1 1  0 0 0  0 0  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
260.0 0  0 0 0  0 0  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
261.8 0  0 0 0  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
300.0 0  1 0 0  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
304.0 0  1 0 1  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
308.2 0  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
310.0 1  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
311.4 1  1 0 1  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
350.0 1  0 0 1  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
354.0 1  0 0 0  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
358.1 1  0 0 0  0 0  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
360.0 0  0 0 0  0 0  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
361.8 0  0 0 0  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
400.0 0  1 0 0  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
404.0 0  1 0 1  0 0  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
408.2 0  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
410.0 1  1 0 1  0 1  1 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
411.4 1  1 0 1  0 1  0 0 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
450.0 1  0 0 1  0 1  0 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
454.0 1  0 0 0  0 1  0 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
458.1 1  0 0 0  0 0  0 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
460.0 0  0 0 0  0 0  0 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
461.8 0  0 0 0  0 0  1 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
470.0 0  0 0 0  0 0  1 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
500.0 0  1 0 0  0 0  1 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
504.0 0  1 0 1  0 0  1 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
510.0 1  1 0 1  0 0  1 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
550.0 1  0 0 1  0 0  1 1 1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
554.0 1  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
560.0 0  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
600.0 0  1 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
604.0 0  1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
610.0 1  1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
650.0 1  0 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
654.0 1  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
660.0 0  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
700.0 0  1 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
704.0 0  1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
710.0 1  1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
750.0 1  0 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
754.0 1  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
760.0 0  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
800.0 0  1 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
804.0 0  1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
810.0 1  1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
850.0 1  0 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
854.0 1  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
860.0 0  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
900.0 0  1 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
904.0 0  1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
910.0 1  1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
950.0 1  0 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
954.0 1  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
960.0 0  0 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  XXXXXXXXr XXXXXX
1000.0 0 1 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  0FFFFC00 XXXXXX
1004.0 0 1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  0FFFFC00 XXXXXX
1010.0 1 1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  0FFFFC00 XXXXXX
1020.0 1 1 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1050.0 1 0 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1054.0 1 0 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1060.0 0 0 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1100.0 0 1 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1104.0 0 1 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1110.0 1 1 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1116.7 1 1 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1121.0 1 1 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XFFFFX
1122.1 1 1 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1150.0 1 0 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1154.0 1 0 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1160.0 0 0 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1200.0 0 1 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1204.0 0 1 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1208.2 0 1 0 1  0 1  1 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1210.0 1 1 0 1  0 1  1 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1211.4 1 1 0 1  0 1  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1218.1 1 1 0 1  0 1  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 0FFFFC
1222.6 1 1 0 1  0 1  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1250.0 1 0 0 1  0 1  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1254.0 1 0 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1258.1 1 0 0 0  0 0  0 0 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1260.0 0 0 0 0  0 0  0 0 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1261.8 0 0 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1300.0 0 1 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1304.0 0 1 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1310.0 1 1 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1350.0 1 0 0 1  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1354.0 1 0 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1360.0 0 0 0 0  0 0  0 1 1  1  0  0  0  1  0  0  0  0  0  1  0  0FFFFC00 XXXXXX
1400.0 0 1 0 0  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  0FFFFC00 XXXXXX
1404.0 0 1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  0FFFFC00 XXXXXX
1410.0 1 1 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  0FFFFC00 XXXXXX
1450.0 1 0 0 1  0 0  0 1 1  1  0  0  0  1  0  1  0  0  0  1  0  0FFFFC00 XXXXXX
```

1454.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC00	XXXXXX
1460.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC00	XXXXXX
1500.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC10	XXXXXX
1504.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC10	XXXXXX
1510.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC10	XXXXXX
1520.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1550.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1554.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1560.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1600.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1604.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1610.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1616.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1621.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XFFFFX
1622.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1650.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1654.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1660.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1700.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1704.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1708.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1710.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1711.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1718.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	0FFFFC
1722.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1750.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1754.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1758.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1760.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1761.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1800.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1804.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1810.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1850.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1854.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1860.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFC10	XXXXXX
1900.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC10	XXXXXX
1904.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC10	XXXXXX
1910.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC10	XXXXXX
1950.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC10	XXXXXX
1954.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC10	XXXXXX
1960.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFC10	XXXXXX
2000.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFFFF	XXXXXX
2004.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFFFF	XXXXXX
2010.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFFFF	XXXXXX
2020.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2050.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2054.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2060.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2100.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2104.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2110.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2116.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2121.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XFFFFF
2122.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF
2150.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF
2154.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF
2160.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF
2200.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF
2204.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF
2208.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF
2210.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF
2211.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF
2218.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	0FFFFF

2222.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2250.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2254.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2258.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2260.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2261.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2300.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2304.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2310.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2350.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2354.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2360.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFFFF	XXXXXX
2400.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFFFF	XXXXXX
2404.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFFFF	XXXXXX
2410.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFFFF	XXXXXX
2450.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFFFF	XXXXXX
2454.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFFFF	XXXXXX
2460.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFFFF	XXXXXX
2500.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFE00	XXXXXX
2504.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFE00	XXXXXX
2510.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFE00	XXXXXX
2520.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2550.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2554.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2560.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2600.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2604.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2610.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2616.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2621.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XFFFFX
2622.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2650.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2654.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2660.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2700.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2704.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2708.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2710.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2711.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2718.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	0FFFFE
2722.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2750.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2754.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2758.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2760.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2761.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2800.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2804.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2810.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2850.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2854.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2860.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0FFFFE00	XXXXXX
2900.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFE00	XXXXXX
2904.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFE00	XXXXXX
2910.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFE00	XXXXXX
2950.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFE00	XXXXXX
2954.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFE00	XXXXXX
2960.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0FFFFE00	XXXXXX
3000.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000001	XXXXXX
3004.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000001	XXXXXX
3010.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000001	XXXXXX
3020.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3050.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3054.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX

reset1.list Mon Jan 30 16:47:24 1989 3

3060.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3100.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3104.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3110.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3116.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3121.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3122.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3150.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3154.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3160.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3200.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3204.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3208.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3210.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3211.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3218.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	000000
3222.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3250.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3254.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3258.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3260.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3261.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3300.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3304.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3310.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3350.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3354.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3360.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000001	XXXXXX
3400.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000001	XXXXXX
3404.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000001	XXXXXX
3410.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000001	XXXXXX
3450.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000001	XXXXXX
3454.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000001	XXXXXX
3460.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000001	XXXXXX
3500.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000020	XXXXXX
3504.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000020	XXXXXX
3510.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000020	XXXXXX
3520.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3550.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3554.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3560.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3600.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3604.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3610.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3616.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3621.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3622.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3650.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3654.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3660.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3700.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3704.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3708.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3710.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3711.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3718.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	000000
3722.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3750.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3754.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3758.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3760.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3761.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3800.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3804.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3810.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3850.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3854.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3860.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000020	XXXXXX
3900.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000020	XXXXXX
3904.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000020	XXXXXX
3910.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000020	XXXXXX
3950.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000020	XXXXXX
3954.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000020	XXXXXX
3960.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000020	XXXXXX
4000.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00001000	XXXXXX
4004.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00001000	XXXXXX
4010.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00001000	XXXXXX
4020.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4050.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4054.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4060.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4100.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4104.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4110.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4116.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4121.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4122.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4150.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4154.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4160.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4200.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4204.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4208.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4210.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4211.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4218.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	000010
4222.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4250.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4254.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4258.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4260.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4261.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4300.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4304.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4310.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4350.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4354.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4360.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00001000	XXXXXX
4400.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00001000	XXXXXX
4404.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00001000	XXXXXX
4410.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00001000	XXXXXX
4450.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00001000	XXXXXX
4454.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00001000	XXXXXX
4460.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00001000	XXXXXX
4500.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	CFFFFFFF	XXXXXX
4504.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	CFFFFFFF	XXXXXX
4510.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	CFFFFFFF	XXXXXX
4520.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4550.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4554.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4560.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4600.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4604.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4610.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4616.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4621.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XFFFFF
4622.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF
4650.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF

4654.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF
4660.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF
4700.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF
4704.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF
4708.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF
4710.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF
4711.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF
4718.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	CFFFFF
4722.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4750.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4754.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4758.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4760.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4761.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4800.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4804.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4810.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4850.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4854.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4860.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	CFFFFFFF	XXXXXX
4900.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	CFFFFFFF	XXXXXX
4904.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	CFFFFFFF	XXXXXX
4910.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	CFFFFFFF	XXXXXX
4950.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	CFFFFFFF	XXXXXX
4954.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	CFFFFFFF	XXXXXX
4960.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	CFFFFFFF	XXXXXX
5000.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00040000	XXXXXX
5004.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00040000	XXXXXX
5010.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00040000	XXXXXX
5020.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5050.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5054.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5060.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5100.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5104.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5110.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5116.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5121.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5122.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5150.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5154.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5160.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5200.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5204.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5208.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5210.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5211.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5218.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	000400
5222.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5250.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5254.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5258.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5260.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5261.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5300.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5304.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5310.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5350.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5354.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5360.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00040000	XXXXXX
5400.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00040000	XXXXXX
5404.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00040000	XXXXXX
5410.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00040000	XXXXXX
5450.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00040000	XXXXXX

5454.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00040000	XXXXXX
5460.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00040000	XXXXXX
5500.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00080000	XXXXXX
5504.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00080000	XXXXXX
5510.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00080000	XXXXXX
5520.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5550.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5554.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5560.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5600.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5604.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5610.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5616.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5621.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5622.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5650.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5654.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5660.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5700.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5704.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5708.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5710.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5711.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5718.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	000800
5722.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5750.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5754.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5758.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5760.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5761.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5800.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5804.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5810.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5850.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5854.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5860.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00080000	XXXXXX
5900.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00080000	XXXXXX
5904.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00080000	XXXXXX
5910.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00080000	XXXXXX
5950.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00080000	XXXXXX
5954.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00080000	XXXXXX
5960.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00080000	XXXXXX
6000.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000004	XXXXXX
6004.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000004	XXXXXX
6010.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000004	XXXXXX
6020.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6050.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6054.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6060.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6100.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6104.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6110.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6116.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6121.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6122.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000
6150.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000
6154.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000
6160.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000
6200.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000
6204.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000
6208.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000
6210.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000
6211.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000
6218.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	000000

6222.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6250.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6254.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6258.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6260.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6261.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6300.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6304.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6310.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6350.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6354.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6360.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000004	XXXXXX
6400.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000004	XXXXXX
6404.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000004	XXXXXX
6410.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000004	XXXXXX
6450.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000004	XXXXXX
6454.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000004	XXXXXX
6460.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000004	XXXXXX
6500.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0000000F	XXXXXX
6504.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0000000F	XXXXXX
6510.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0000000F	XXXXXX
6520.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6550.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6554.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6560.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6600.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6604.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6610.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6616.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6621.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6622.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6650.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6654.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6660.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6700.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6704.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6708.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6710.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6711.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6718.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	000000
6722.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6750.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6754.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6758.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6760.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6761.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6800.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6804.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6810.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6850.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6854.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6860.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0000000F	XXXXXX
6900.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0000000F	XXXXXX
6904.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0000000F	XXXXXX
6910.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0000000F	XXXXXX
6950.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0000000F	XXXXXX
6954.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0000000F	XXXXXX
6960.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0000000F	XXXXXX
7000.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000002	XXXXXX
7004.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000002	XXXXXX
7010.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000002	XXXXXX
7020.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7050.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7054.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX

7060.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7100.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7104.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7110.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7116.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7121.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7122.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7150.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7154.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7160.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7200.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7204.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7208.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7210.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7211.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7218.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	000000
7222.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7250.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7254.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7258.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7260.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7261.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7300.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7304.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7310.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7350.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7354.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7360.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7400.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000002	XXXXXX
7404.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	1	0	0	0	00000002	XXXXXX
7410.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000002	XXXXXX
7450.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000002	XXXXXX
7454.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000002	XXXXXX
7460.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000002	XXXXXX
7500.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000000	XXXXXX
7504.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000000	XXXXXX
7510.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000000	XXXXXX
7520.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7550.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7554.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7560.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7600.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7604.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7610.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7616.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7622.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7650.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7654.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7660.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7700.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7704.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7708.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7710.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7711.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7718.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	000000
7722.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7750.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7754.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7758.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7760.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7761.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7800.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7804.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7810.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX

7850.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7854.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7860.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	00000000	XXXXXX
7900.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000000	XXXXXX
7904.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000000	XXXXXX
7910.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000000	XXXXXX
7950.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000000	XXXXXX
7954.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000000	XXXXXX
7960.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	00000000	XXXXXX
8000.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	FFFFFFFD	XXXXXX
8004.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	FFFFFFFD	XXXXXX
8010.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	FFFFFFFD	XXXXXX
8020.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8050.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8054.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8060.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8100.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8104.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8110.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8116.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8121.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8122.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8150.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8154.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8160.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8200.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8204.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8208.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8210.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8211.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8218.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	FFFFFF
8222.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8250.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8254.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8258.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8260.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8261.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8300.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8304.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8310.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8350.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8354.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8360.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	FFFFFFFD	XXXXXX
8400.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	FFFFFFFD	XXXXXX
8404.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	FFFFFFFD	XXXXXX
8410.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	FFFFFFFD	XXXXXX
8450.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	FFFFFFFD	XXXXXX
8454.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	FFFFFFFD	XXXXXX
8460.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	FFFFFFFD	XXXXXX
8500.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8504.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8510.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8520.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8550.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8554.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8560.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8600.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8604.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8610.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8616.7	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8621.0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8622.1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8650.0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8654.0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8660.0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8700.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8704.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8708.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8710.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8711.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8718.1	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	000000
8722.6	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8750.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8754.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8758.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8760.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8761.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8800.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8804.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8808.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8810.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8811.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8850.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8854.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8858.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8860.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8861.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	000000FF	XXXXXX
8900.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8904.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8908.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8910.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8911.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8950.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8954.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8958.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8960.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
8961.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9000.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9004.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9008.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9010.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9011.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9050.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9054.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9058.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9060.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9061.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9100.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9104.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9108.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9110.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9111.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9150.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9154.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9158.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9160.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9161.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9200.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9204.0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9208.2	0	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9210.0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9211.4	1	1	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9250.0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9254.0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9258.1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9260.0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9261.8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX
9300.0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	000000FF	XXXXXX

```
9304.0 0  1  0  1  0  0  1  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9308.2 0  1  0  1  0  1  1  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9310.0 1  1  0  1  0  1  1  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9311.4 1  1  0  1  0  1  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9350.0 1  0  0  1  0  1  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9354.0 1  0  0  0  0  1  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9358.1 1  0  0  0  0  0  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9360.0 0  0  0  0  0  0  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9361.8 0  0  0  0  0  0  1  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9400.0 0  1  0  0  0  0  1  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9404.0 0  1  0  1  0  0  1  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9408.2 0  1  0  1  0  1  1  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9410.0 1  1  0  1  0  1  1  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9411.4 1  1  0  1  0  1  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9450.0 1  0  0  1  0  1  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9454.0 1  0  0  0  0  1  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9458.1 1  0  0  0  0  0  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9460.0 0  0  0  0  0  0  0  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX
9461.8 0  0  0  0  0  0  1  1  1  0  0  0  1  0  1  0  0  0  1  0  000000FF  XXXXXX

 TIME  ^cmclk.c    ^shifta     ^s3         ^s1         ^s11        ^kpMD
          ^cmclk.b    ^mclk        ^s7         ^s2         ^s12            ^E
             ^shifta.b   ^mclk*       ^s8         ^s6         ^s13
                ^cmclk      ^por*       ^s9         ^s10        ^s14
```