

Naval Research Laboratory

Washington, DC 20375-5000



NRL Memorandum Report 6775

AD-A234 127

Testing the One-Port Random Access Memory (1PRAM) Module of TRW's CPUAX Signal Processing Superchip

GREGORY C. TAVIK

Radar Division

April 16, 1991

DTIC
ELECTE
APR 22 1991
S B D

Approved for public release; distribution unlimited.

91 4

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 1991 April 16	3. REPORT TYPE AND DATES COVERED Final	
4. TITLE AND SUBTITLE Testing the One-Port Random Access Memory (1PRAM) Module of TRW's CPUAX Signal Processing Superchip		5. FUNDING NUMBERS PE - 63452F	
6. AUTHOR(S) Gregory C. Tavik			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory Radar Division Code 5305 Washington, DC 20375-5000		8. PERFORMING ORGANIZATION REPORT NUMBER NRL Memorandum Report 6775	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Wright Research and Development Center Wright-Patterson AFB, OH 45433-6543		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The CPUAX Signal Processing Superchip has been developed by TRW under the Very High Speed Integrated Circuit (VHSIC) Phase II program. The chip utilizes a 0.5 um CMOS process and achieves a maximum throughout of 200 Mflops. Among the chip's 61 active macrocells, are 39 identical 4Kx1-bit one port random access memory (1PRAM) modules. This report describes the testing of the functional performance and VHSIC Hardware Description Language (VHDL) model of the 1PRAM macrocell by NRL, describes the testing procedures, the problems encountered, and the ramifications of these. An algorithm is also included for future testing of similar devices.			
14. SUBJECT TERMS VHSIC 1PRAM VHDL Daisy PMX CPUAX Superchip TRW			15. NUMBER OF PAGES 51
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR

CONTENTS

I. INTRODUCTION	1
II. PREPARATION FOR TESTING WITH THE DAISY PMX	2
III. TESTING THE 1PRAM	10
TRW TESTS	10
TEST_ADEC	10
TEST_PDEC	10
TEST_PROFF	10
TEST_OUTMUX	12
TEST_RWEN.1	12
TEST_RWEN.2	12
TEST_RWEN.3	12
TEST_WRITE	12
TEST_READ	12
IV. TESTING RESULTS AND IMPLICATIONS	17
V. 1PRAM HARDWARE PROBLEMS	18
VI. CONCLUSIONS	18
REFERENCES	19
NOTES	20
APPENDIX	21

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
Distribution/	
Availability Codes	
Dist	Special
A-1	

TESTING THE ONE-PORT RANDOM ACCESS MEMORY (1PRAM) MODULE OF TRW'S CPUAX SIGNAL PROCESSING SUPERCHIP

I. INTRODUCTION

The CPUAX Signal Processing Superchip has been developed by TRW under the VHSIC Phase II program. The chip utilizes a 0.5 um CMOS process and achieves a maximum throughput of 200 Mflops. The chip consists of 61 active macrocells, including 2 floating point arithmetic logic units (ALU's), 2 multiply accumulators (MAC's), read and write address generators, 6 storage elements, read and write memory interfaces, and 39 identical 4Kx1-bit one port random access memory (1PRAM) modules. Each macrocell is provided with a chip specification and an IEEE 1076 VHSIC Hardware Description Language (VHDL) functional model.

This report describes the testing of the functional performance and VHDL model of the 1PRAM macrocell by NRL. Testing was performed for two reasons: to verify that the macrocell behaves as specified and to validate the VHDL model. Three 1PRAM macrocells were received, each mounted on a 28-pin DIP test chip. All testing described herein was carried out on a Daisy Megalogician connected to a Physical Modeling Extension (PMX) using the 5.03 version of the Daisy DNIX operating system.

This report describes the testing procedures, the problems encountered, and the ramifications of these. An algorithm is also included for future testing of similar devices on the Daisy CAD system with use of the PMX. It is shown that the 1PRAM functioned as TRW described in its design publications, which can be found in the References. However, the test vectors provided by TRW, which were delivered with the VHDL model, did not activate the chip as expected. This VHDL model was also shown to be invalid. These issues are discussed in Section IV, Testing Results and Implications.

Some prior experience with the Daisy CAD system is recommended for a better understanding of this report. The manuals entitled Design Compilation and Verification Support System II by Daisy will be helpful.

II. PREPARATION FOR TESTING WITH THE DAISY PMX

This section is intended for anyone testing a device using the Daisy Simulator (MDLS2) on a Megalogician in conjunction with the Physical Modeling Extension (PMX).

A short description of the PMX is in order. The PMX of the Daisy CAD system may be used for two general purposes. The first is to allow a device to "model itself" in a design simulation. That is, the PMX can save the designer the trouble of creating a complicated Simulator Parameter Compiler (SPARC), or Daisy Behavioral Language (DABL) model for an available part. The PMX may also double as a functional test instrument. It is this role of the PMX which was utilized for the testing of the 1PRAM. Instead of sending "signals" to a software model, signals are sent directly to the actual device. Here they are processed by the hardware and the outputs are sent back to the simulator. In this way, test vectors may be sent to the device from user written software and real outputs can be recorded.

Once a pinout of the device is acquired, the preparation for testing may begin. The first job is to create a drawing with ACE, the Daisy schematic editor, of the component along with any additional logic needed that the ACE libraries may provide. Figure 1 shows the drawing created for the 1PRAM with a comparator included. The use of this comparator will be explained in Section III.

If the part to be tested is not contained in an ACE library, an ACE library drawing of the part needs to be generated. This may be accomplished in the following way.

1. While in ACE, click on "create_comp" in the "LOGIC" pop-up menu.
2. Draw the part as desired.
3. NAME each pin. (e.g. - A0, A1, D0, D1, RWEN)
4. NUMBER each pin. (NUMBER is a Type of PARAMETER)
5. SAVE the component in an appropriate existing library, or create a new library.

Once this has been completed, call the component from the ACE library and finish the drawing. After the ACE drawing of the test layout is complete, the following commands should be executed in the context of the top level drawing:

DANCE -T -ERR (CR)¹

This will traverse the design tree and create a file called ERR.DFR which contains all error messages concerning the schematics.

DRINK (CR)

This command writes a file called TREE.DFR which contains all DRINK error messages. All DANCE errors must be corrected before the execution of DRINK.

Next, create a SPARC control file. This file is specific to the component created in ACE and will be accessed whenever the component is used in the simulator. This file, which may be written with Daisy's text editor, TEC, contains such information as the ACE library name of the device, its technology, delays, a correlation between pin NAME and pin NUMBER, and whether the pin is an INPUT, OUTPUT, or BIDIREctional pin. Figure 2 shows the SPARC file for the 1PRAM.

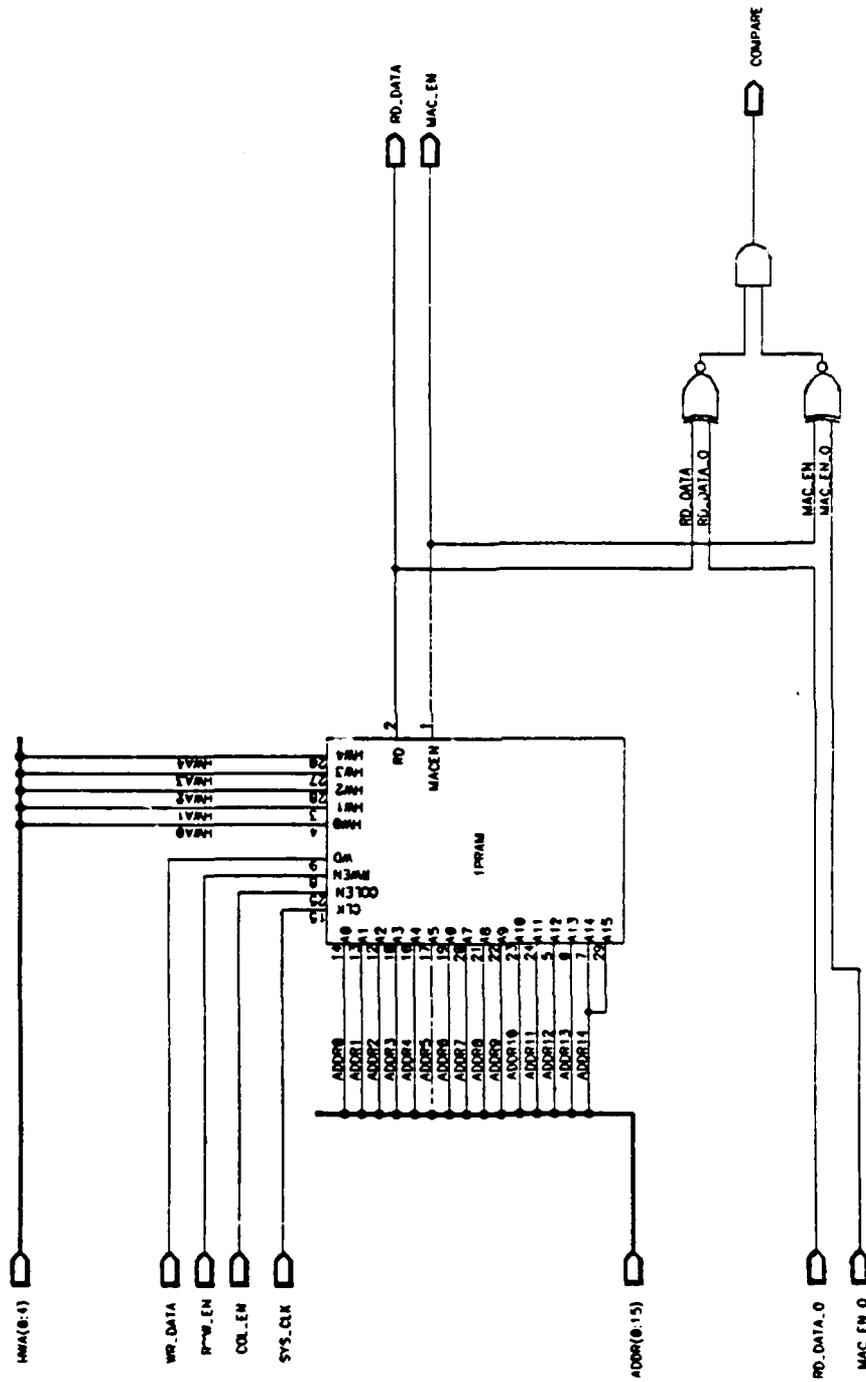


Figure 1
Schematic Editor (ACE) Circuit Drawing

Next, the SPARC file must be linked with its ACE drawing and assigned to a library. Execute the command below while in the context of the top drawing page.

SPARC file-name TO library (CR)

The "TO library" is optional, but if it is not specified, the Daisy will create a new library in the current context called "file-name.LIB". If this is done, this new library and its path name must be entered into the PROFILE file in the user's home directory under the heading \$SIFT_COMPONENT_LIB_REF\$. This is necessary so that when the SIFT command (considered in a moment) is executed, the simulator may locate this new library.

```

$TYPE_INFO
1PRAM: XDEV C B B
<INPUT: A0 PARAMS [PIN(14)]
A1 PARAMS [PIN(13)]
A2 PARAMS [PIN(12)]
A3 PARAMS [PIN(10)]
A4 PARAMS [PIN(16)]
A5 PARAMS [PIN(17)]
A6 PARAMS [PIN(19)]
A7 PARAMS [PIN(20)]
A8 PARAMS [PIN(21)]
A9 PARAMS [PIN(22)]
A10 PARAMS [PIN(23)]
A11 PARAMS [PIN(24)]
A12 PARAMS [PIN(5)]
A13 PARAMS [PIN(6)]
A14 PARAMS [PIN(7)]
A15 PARAMS [PIN(29)]
CLK PARAMS [PIN(15)]
WD PARAMS [PIN(9)]
RWEN PARAMS [PIN(8)]
COLEN PARAMS [PIN(25)]
HW0 PARAMS [PIN(4)]
HW1 PARAMS [PIN(3)]
HW2 PARAMS [PIN(28)]
HW3 PARAMS [PIN(27)]
HW4 PARAMS [PIN(26)];
OUTPUT: MACEN PARAMS [PIN(1)];
RD PARAMS [PIN(2)];
$END

```

Figure 2
SPARC Control File

After the SPARC command has been successfully executed, Simulator Intermediate File Translator (SIFT) must be invoked in the context of the ACE drawing.

SIFT (CR)

This command will catch any current errors that may be found in the SPARC file, and display them on the screen.

One pitfall of this last procedure is the mistake of confusing an ACE library file with a SIFT library file and accidentally overwriting one with a new version of the other. These are not at all the same, and one of the files will have to be rewritten or redrawn if this

misunderstanding occurs. One way to prevent such an error is to keep all ACE library files under a directory called ACE_LIB in the user's home directory. Likewise, the user may create a specific directory for SIFT library files.

Another detail which may cause problems at the SIFT run time is the order in which the PROFILE variable parameters are arranged within a file called "logfile". This file can also be found in the home directory. The path name of a personal PROFILE file should come before the location of any other PROFILE file (such as /PROJECT) which the schematic may be accessing. This is necessary so that the special pointers assigned in the PROFILE file may take precedence over all others being used.

The creation of a Simulator Object Module (SOM) control file particular to a specific test is the next step in examining the component. But first it is necessary to lay out the PMX Daughter Card which will connect the hardware under test to the Daisy CAD system.

PMX channels are the enumerated pins and sockets provided on the Daughter Card. They are the means by which the hardware interfaces with the Daisy simulator. Since the states of these channels are unknown at power-up, all inputs and outputs were buffered to protect the 1PRAM from possible damage. Four MM74C244's were used; their positions along with the 1PRAM's location on the Daughter Card are shown in Figure 3. Note that asterisks indicate pins where jumpers may be employed to connect that particular channel to the hardware. Circles indicate socket holes where 300, 400, and 600 mil DIP chips may be plugged into the card. Solid lines indicate ground wires. The dashed line denotes the power line. Power was brought onto the board by connecting channel 80 to the power supply, 3.3 volts in this case, and channel 33 to the ground of the power supply. This was done because the 1PRAM drew the most current (104 mA at maximum) of all the chips mounted on the card. A number between two pins denotes which pin of the 1PRAM is connected there. The card is divided into two columns, left and right. Always wire wrap to the inner pins of the column, since these are wired to the chips. The outer pins are connected to the PMX board which leads to the simulator. A small thick line drawn between pins denotes that a jumper is installed there. All these channels, except 129 and 131, are pulled down by the resistor packs shown at the bottom of the card. This was done so that the nominal high output from the PMX to the chip of 5 volts could be limited to approximately 2.8 volts. This effective high input to the buffers functioned well with the supply voltage of 3.3 volts. The wires for these are not shown.

When this wiring of the Daughter Card has been finished, a SOM control file needs to be created. The \$PMX_INFO section of the file contains two parts called PMX_BOARD and COMPONENT. Under PMX_BOARD, the board type and base address are given. For a static board, the only type presently owned by the Radar Division, the type number is 0F0A7EFH. The "H" signifies that the number is in hexadecimal format. The base address of the board to be used may be found in the following manner.

1. Attach jumpers to any of the sites labeled W1 through W16 on the Daughter Card. This is a 16-bit word which associates the Daughter Card with an ID number. A jumper employed on a pair of pins denotes a 1 for that digit.
2. Plug the card into one of the slots and turn on the PMX.
3. Type these commands from DNIX:
CD /PROJECT/DIAG (CR)
PMXDIAGM (CR)

A list appears on the screen of the PMX boards with their base addresses and their associated

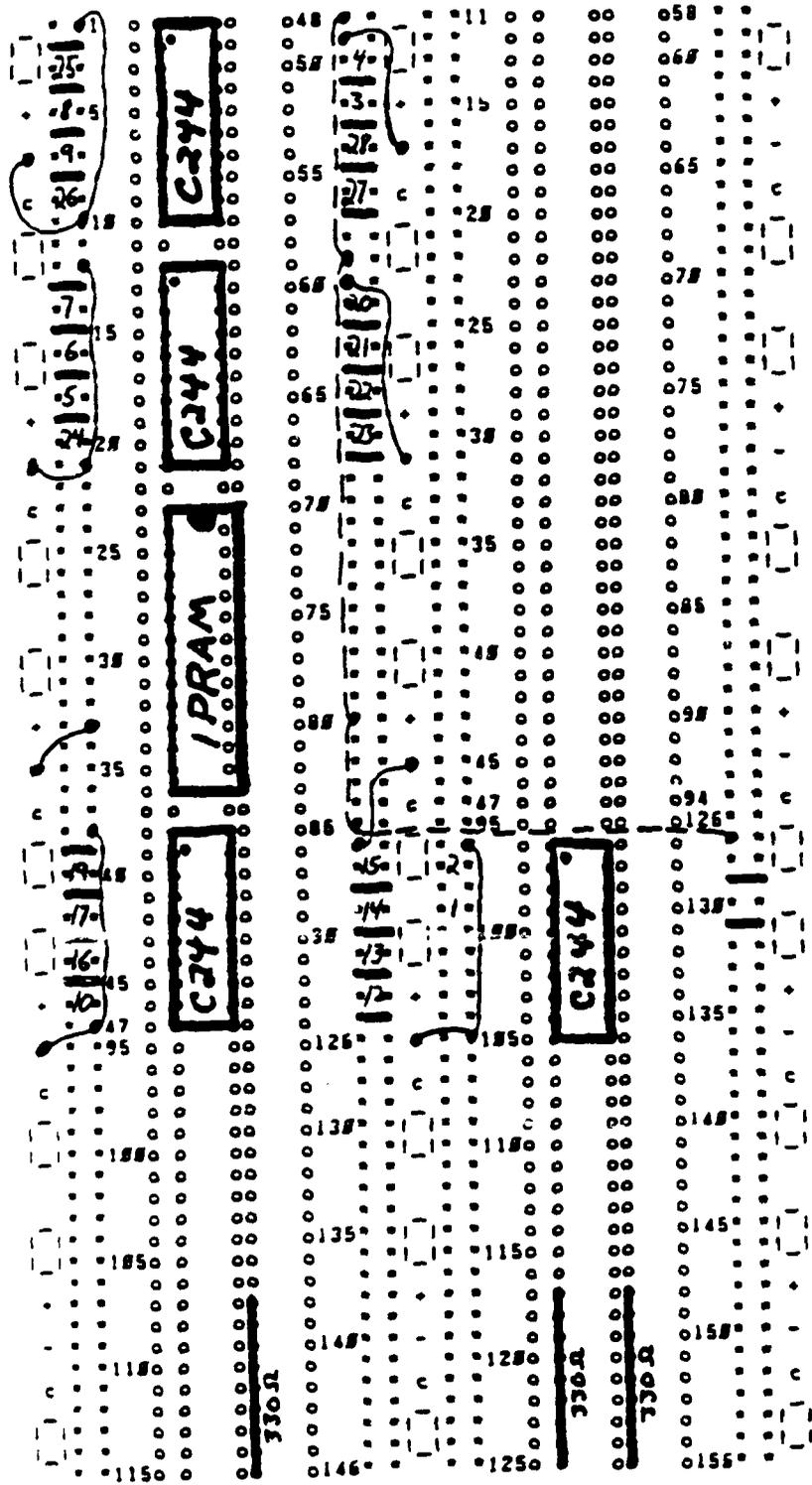


Figure 3
PMX Daughter Card Layout

1. KERMIT (CR)
2. CTRL-]-B
3. net (CR) (CR)
4. c 128.60.3.4 (CR) (number specifies the Sun's address)
5. kermit -s filename (CR)
6. CTRL-]-C (CR)
7. r (CR)
8. conn (CR) (reconnect to the Sun)
9. logout (CR)
10. CTRL-]-B (CR many times until menu appears)
11. rel (CR)

The files were then saved onto a floppy disk, the floppy was placed into the Daisy drive, and the following DNIX commands were executed:

1. DOSTYPE A:FILE1 >FILE2 (CR)
FILE2 may not have the same name as FILE1. FILE2 will go into the current directory.
2. NLINE -LF <FILE2 >FILE3 (CR)
This command removes the line feed character (␣) from ASCII files. FILE2 and FILE3 may not have the same name.
3. colrm startcol# endcol# <FILE3 >FILE4 (CR)
This removes all unwanted columns in the ASCII file and moves the remaining columns to the left. FILE3 and FILE4 may not have the same name.
4. TEC FILE4 (CR)

Once in TEC, character substitutions may be performed. For instance, the files received from TRW contained some vectors in hexadecimal format. Hexadecimal is not a readable format for an INPUT file, only binary is acceptable. Hence, all the hex numbers had to be converted to binary. This was achieved by using the SUB command while in TEC. If this last procedure is necessary then be sure to substitute each hex (or octal, etc.) digit in ascending order. That is, substitute 0000 for 0 before replacing 8 with 1000. This prevents recursive substitution which would corrupt the file. If extensive substitution is required, it may be more practical to write a small program to do the conversion.

This process will convert any UNIX file to DNIX. If the file came from PC-DOS use the -CR option instead of -LF with the NLINE command. Generating input files from a PC program is an easy method to produce such things as addressing input vectors. This technique was employed in the creation of the input files for TEST_WRITE and TEST_READ discussed later.

The final command to be invoked in the directory of the top level drawing is:

MDLS2 (CR)

This invokes the simulator and opens any input or output files called by the SOM control file.

All of the above was completed in preparation for testing of the 1PRAM. Each test will be described in detail in the remaining pages.

III. TESTING THE 1PRAM

Three packaged 1PRAM IC's were received from TRW/Motorola, the manufacturers. They were each labeled with a number. All initial testing was done with IC #57, and then later with IC #59 and #60. A brief summary of each test is presented below.

TRW TESTS

All these test vectors were written by TRW. The procedure shown in the section above was used to convert all these vectors (input and output) into readable Daisy DNIX ASCII files. This file was given a header and used as an input file to the 1PRAM on the PMX. The two simulated outputs, RD and MACEN, were used as inputs (called RD_DATA_O and MAC_EN_O in the ACE schematic) to a 2-bit comparator so that the real outputs (called RD_DATA and MAC_EN in the ACE schematic) could be compared to the simulated outputs. When the signal, COMPARE, was low, a discrepancy had occurred. All of the following test waveforms are given in the appendix.

TEST_ADEC (Address Decode)

This test checks the read, write, and addressing capabilities of the 1PRAM. It does this by first attempting to flush out the page register and set it to zero. This procedure will be discussed in detail in Section IV. While walking a one across the address lines, ADDR(11:0), it writes 0's and 1's to the addressed memory cells (i.e. - ADDR0 = 0, ADDR1 = 1, ADDR2 = 0, etc.). These addresses are then read back and the RD and MACEN outputs are monitored. The RD line should begin repeating the strobed pattern after two SYSCLK cycles from the falling edge of the RWEN line. MACEN should go high after 16 falling edges of the clock beginning at the rising edge of COLEN.

TEST_PDEC (Page Decode)

This test checks the page register and the shifting procedure needed to load the macrocell with a new page address. As long as the internal signal, SHIFTEN, is high, ADDR10 functions as the data line into the page register (see Figure 6). As data is shifted in, the data shifted out of PR(0) should appear on the RD line. MACEN is observed while a 1 is marched across the page address, ADDR(15:12), following each shift. The specifics of this test, since it deals mainly with the operation of the page register, will be discussed in Section IV.

TEST_PROFF (Page Register Off)

This test is similar to the PDEC test in that the page register's value is changed many times throughout the procedure. Once again, MACEN is observed while a 1 is marched across the page address, following each shift. However, in this test, the value shifted into the page register always leaves PR(4) high, driving MACEN low. The RD line is also observed to catch the data being shifted out of the page register as each new address is shifted into the register.

The next seven tests inspect various aspects of the chip. The set-up pattern for the 1PRAM (zeroing out the page register) was written in accordance with the Motorola set-up

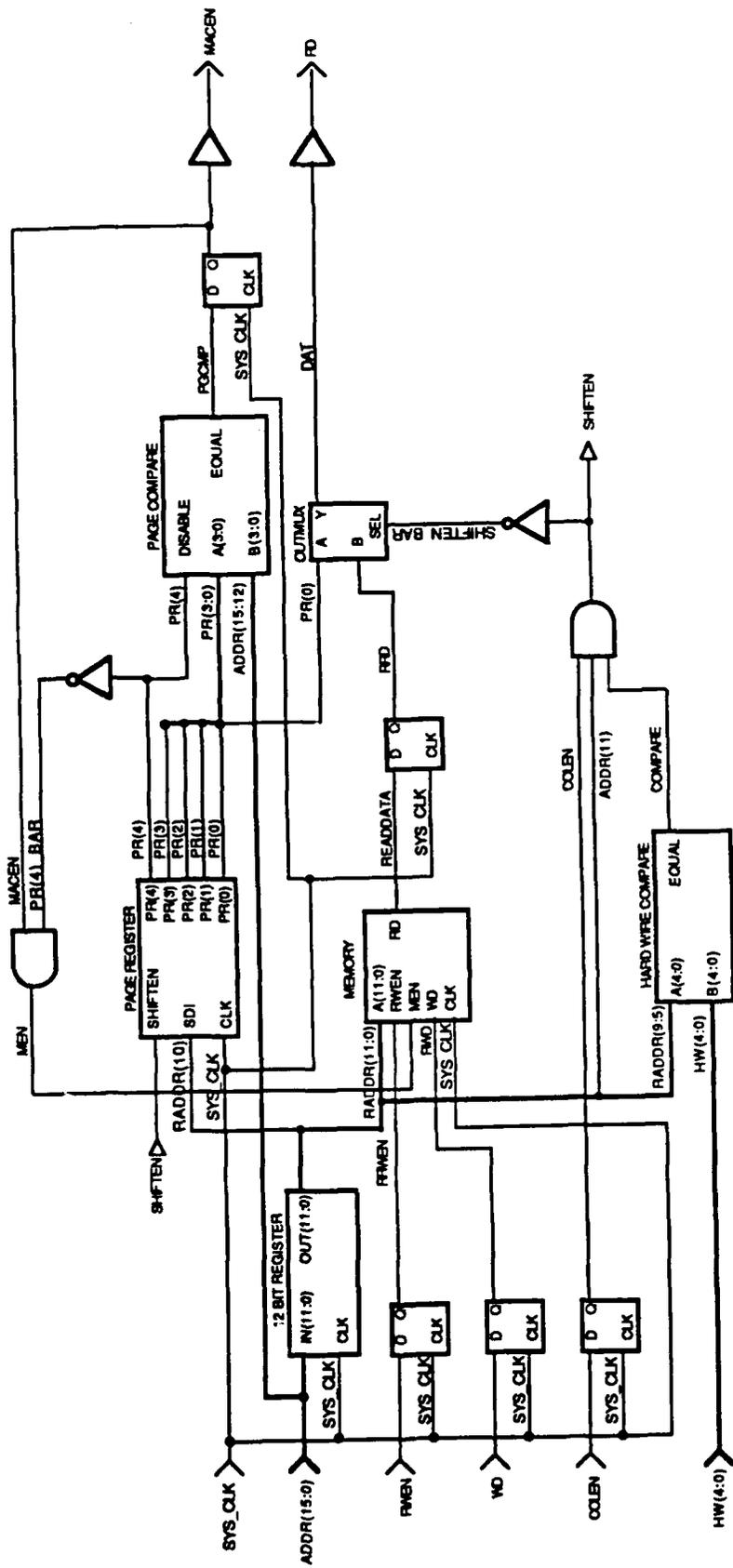


Figure 6
1PRAM Block Diagram

pattern (see Figure 8). All tests assume the chip has already been set-up.

TEST_OUTMUX

This test checks the operation of the multiplexer which selects between the internal signals, PR(0) and RRD, by way of the select, SHIFTEN_BAR (see Figure 6). By setting ADDR11 high and obtaining an address match from the hardwired comparator, COLEN is left as the control signal that determines the state of SHIFTEN.

The test writes a 1 to a particular address (8C0₁₆ in this case), then strobes the COLEN line which in turn toggles the SHIFTEN line. Therefore, an alternating pattern of 1's and 0's should be observed at the RD output as the multiplexer selects between the memory cell and PR(0) (which is set to 0 due to the set-up pattern).

TEST_RWEN.1

This test monitors the functionality of the read-write enable line, RWEN. An address is presented to the macrocell for 3 system clock cycles at a time. During the first cycle, RWEN is held high and a write occurs. For the next two cycles RWEN is held low and the data previously written is read back. The address lines change, and the operation is repeated. The write data line, WD, remains high throughout the test. MACEN should remain high, and RD should do the same after two system clock falling edges from the presentation of the first address to the chip.

TEST_RWEN.2

This is the same test as RWEN.1, except for the action of the WD line. Instead of letting this line remain high, it is toggled every 3 system clock cycles. The address lines change synchronously with the WD line. In this way, an alternating pattern of 1's and 0's can be observed at the RD output.

TEST_RWEN.3

This test is essentially the same as the former, except for the fact that the cycle for writing and reading is shortened to 2 system clocks. That is, the time for reading is limited to one SYSCLK cycle.

TEST_WRITE

This test writes (RWEN = 1) a strobed pattern to every four memory cells (i.e. - ADDR0 - ADDR3 = 0, ADDR4 - ADDR7 = 1, etc.).

TEST_READ

This test is performed immediately after TEST_WRITE and reads (RWEN = 0) all the memory cells in the array. The same pattern that is written should be observed on the RD line.

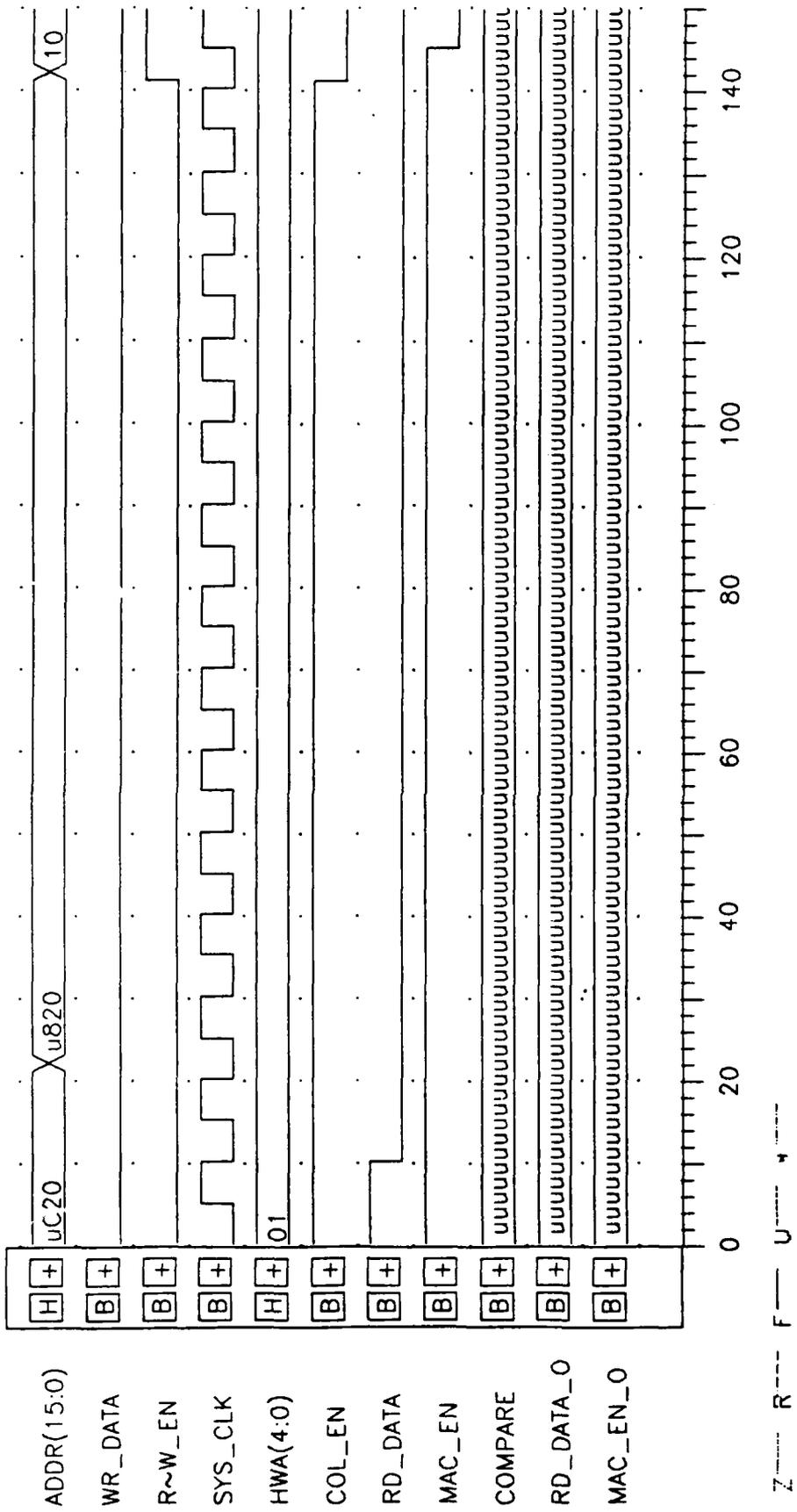


Figure 7.1
Improper Setup Routine for 1PRAM and Results (0 ns- 150 ns)

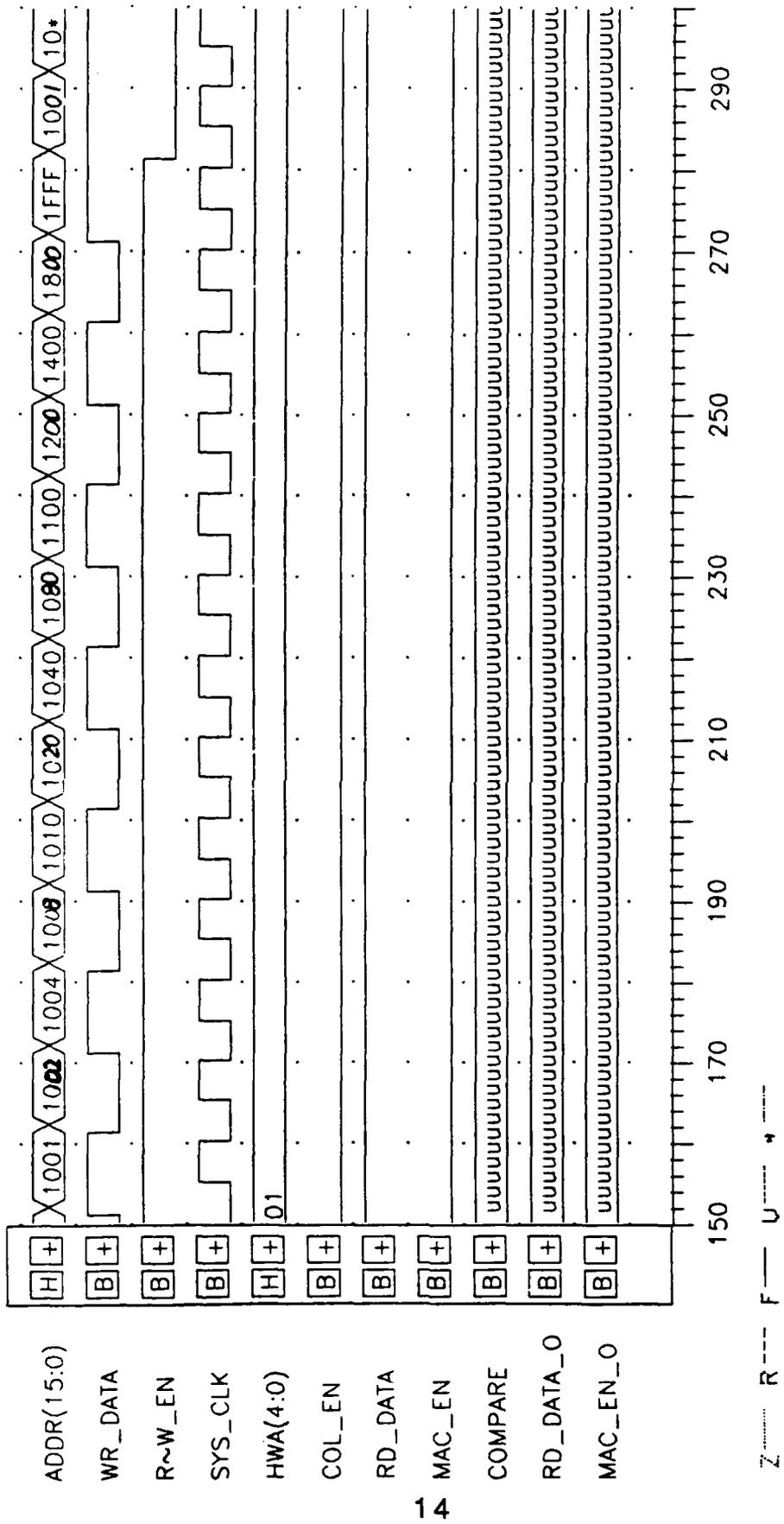


Figure 7.2
Improper Setup Routine for 1PRAM and Results (150 ns - 300 ns)

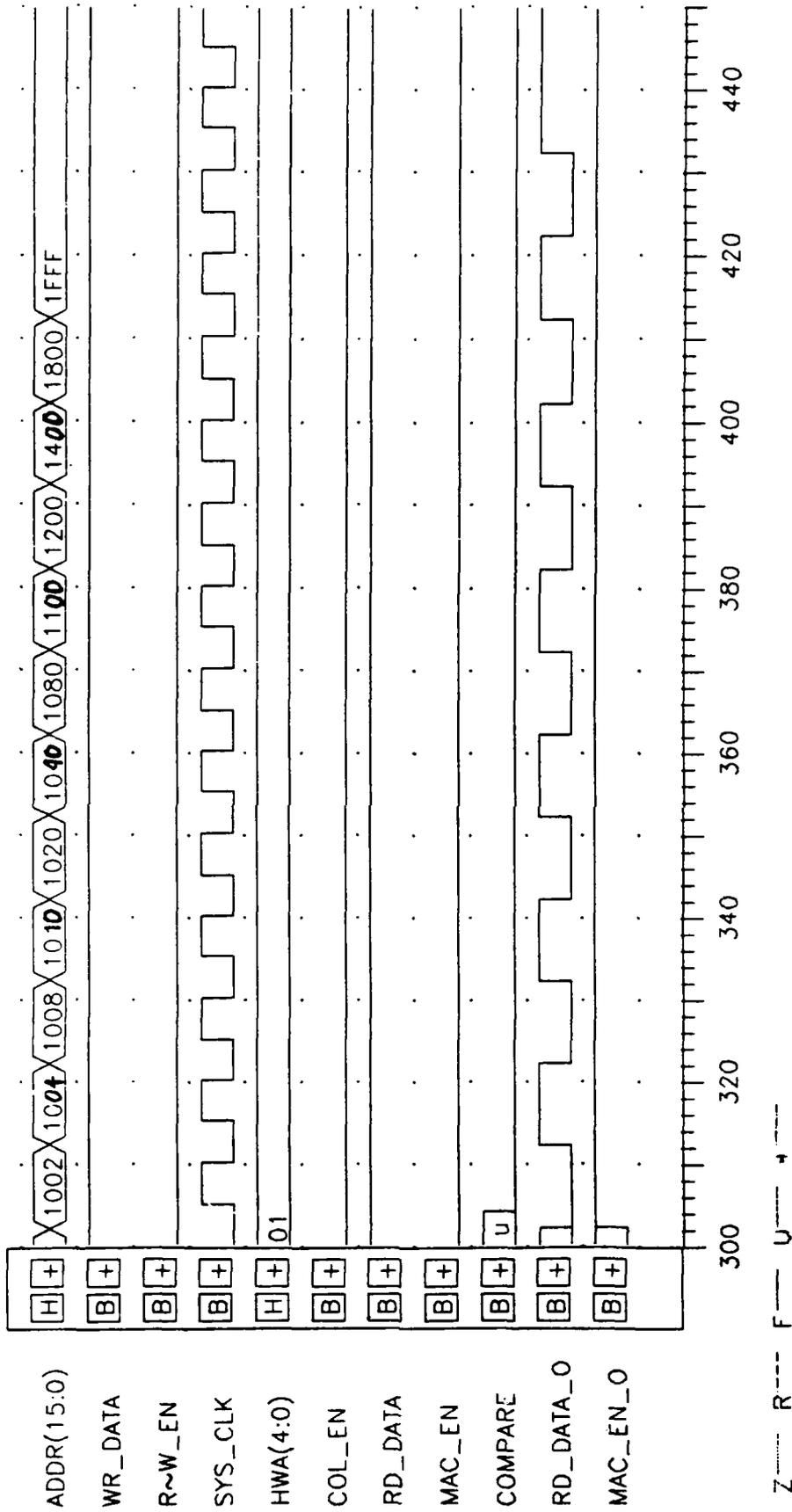


Figure 7.3
Improper Setup Routine for 1PRAM and Results (300 ns - 450 ns)

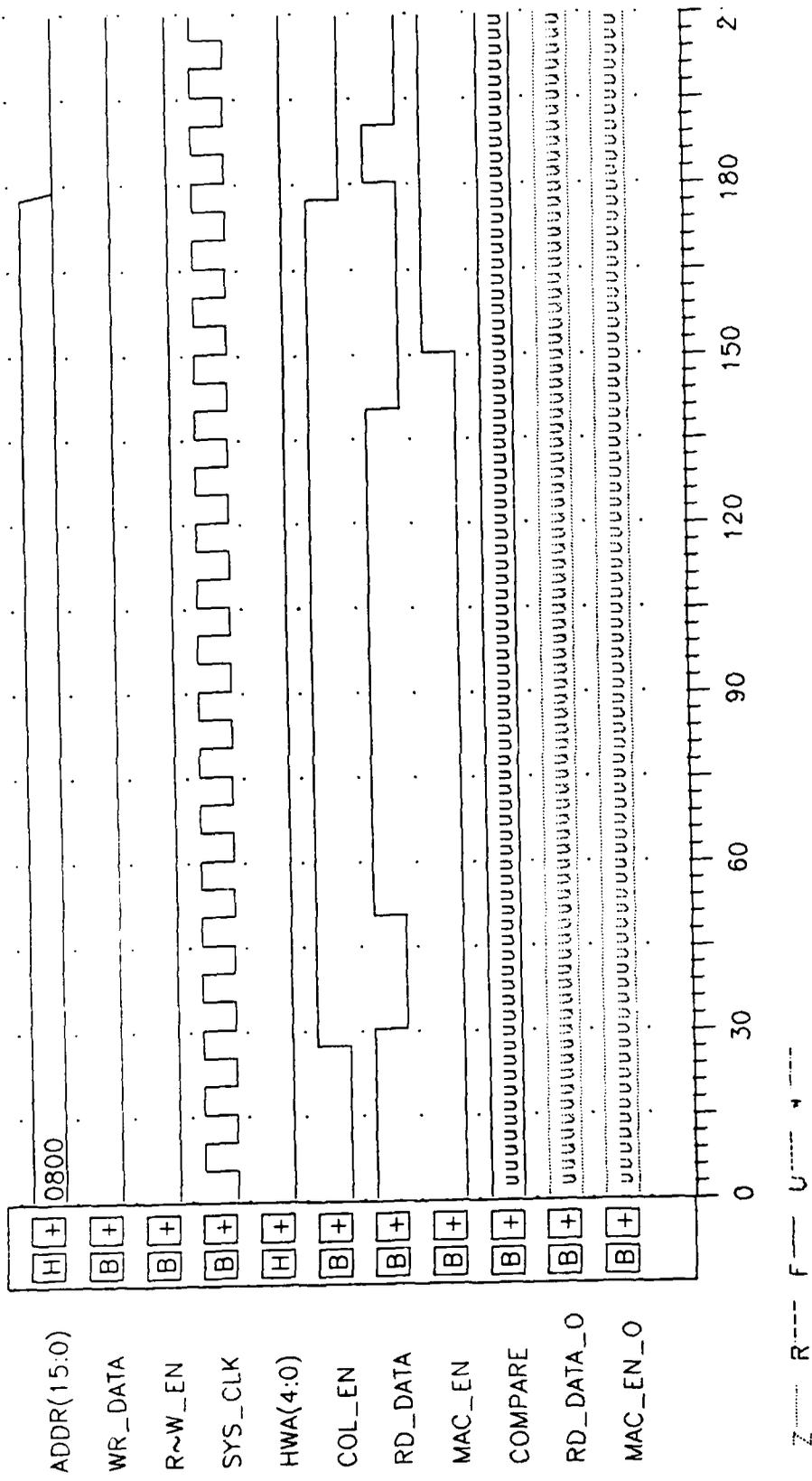


Figure 8
Correct Setup Routine for 1PRAM

IV. TESTING RESULTS AND IMPLICATIONS

With respect to the TRW tests and the VHDL generated outputs, all tests contain a common error in the timing. The set-up timing and all shift operations are one SYSCLK cycle short. In the TRW set-up shown in Figure 7.1, COLEN is held high for only 14 system clocks. This does not allow for the two initial SYSCLK cycles needed to activate the page register clock. This problem was noticed when testing the 1PRAM with TEST_ADEC. As can be seen from the TRW version of the ADEC test (see Figures 7.2 and 7.3), the RD and MACEN signals do not reflect the specifications for the IC (see Section III - TEST_ADEC). Upon consulting with Motorola's test engineers, it was suggested that their set-up routine, shown in Figure 8, be used instead.²

After replacing TRW's set-up vectors with Motorola's, the test operated correctly. The same was done for the PDEC and PROFF tests. However, these tests performed similar shifting operations throughout. So in all places where a shift of the page register was required, the original signals were delayed by 1 system clock cycle beginning at the rising edge of COLEN. This means that the first digit to be shifted into the register was no longer stable for only 2 falling edges of the system clock. It was now kept stable for at least 3 falling edges of the system clock. To be more specific, COLEN would now go high during a high SYSCLK, remain high for 14 full cycles, and finally go low during the next high SYSCLK.

The main problem with the TRW-created test vectors was that ADDR10 was not being presented at the correct time to the page register, but occurred 20 ns (simulation time) too early. This observation has serious ramifications with respect to the VHDL model. These same incorrect vectors were fed to the TRW model which operated as if it had received the correct timing. This means that TRW VHDL model of the 1PRAM is defective. In particular, a problem exists in the code which describes the operation of the page register.

Another problem with the TRW VHDL model was discovered regarding the MACEN line in the ADEC and PDEC tests. The MACEN signal must go high on the next falling edge of the clock immediately after a page match occurs between PR(3:0) and ADDR(15:12). This means that the macrocell has been enabled and is ready to perform read and write operations. The VHDL output for MACEN shows, incorrectly, that the MACEN line is unknown until reading occurs.

After these changes had been made to the TRW tests, all three chips passed all of the tests mentioned in Section III. However, a problem did arise when TEST_ADEC was run on chip #60. Consequently, new tests were developed to investigate the situation further. This is the discussion of the final section of this report.

It should be noted that in the 28-pin DIP version of the chip which was tested, ADDR14 is tied to ADDR15 due to a lack of pins. Hence, only 8 (in hex: 0, 1, 2, 3, C, D, E, F) possible page addresses are allowed instead of the normal 16. The remaining 8 addresses will register as one of the above (4 will register as C, 5 as D, etc.). To check that this was indeed true, the fifth batch of shift data in the PDEC test was changed from 00100₂ to 01100₂ to show that as a 1 was marched across ADDR14, MACEN would be forced high. It was. It was also shown that if PR(3:0) equals a 4 or an 8, MACEN is not forced high as expected by the VHDL outputs in the PDEC and PROFF tests. Address 4xxx₁₆ registers as Cxxx₁₆, and address 8xxx₁₆ registers as 0xxx₁₆, and when a 1 is marched across the address lines, a match never occurs. Therefore, MACEN stays low.

V. 1PRAM HARDWARE PROBLEMS

As mentioned above, chip #60 did not pass the ADEC test. It failed to successfully read a 1 written to address 400₁₆. An investigation was begun to determine how the chip could fail this test yet seem to pass the READ test. The Test Plan shows that ADDR10 and ADDR11 control the quadrant of the memory to be addressed.³ The memory cells are arranged in 64 rows by 64 columns. These columns are divided into 4 quadrants of 16.

A few tests were written to see if these two address lines were working correctly. The most significant of these tests is called TEST_QUAD - Walking Ones. This test, after set-up, writes to addresses 076₁₆, 476₁₆, 876₁₆, and C76₁₆. These four memory cells are all in the same row and column, differing only by their quadrant. A 1 is written to one of the four cells and a 0 to the other three. Then the cells are read to check for errors. In the next write-read cycle the 1 is written to the next cell of the four and 0's to the other three. This is repeated until the 1 has been written to all four cells. Address x8xx worked well, but the other 3 quadrants seemed to be tied together as if the data written to x0xx was also written to x4xx and xCxx. Whatever was written to quadrant 0 showed up in quadrants 1 and 3. It seems that there may be a malfunction in the quadrant decoder, or perhaps some sort of crosstalk is occurring between the different internal WR_SEL signals. This problem was intermittent.

IC #60 may have seemed to pass the READ test because this test wrote to one quadrant at a time. In other words, the pattern that was written to the first quadrant was exactly the same as the one written to all the others. Therefore, the chip could appear to pass this test successfully. As an added note, the other test packages, #57 and #59, passed these QUAD tests.

VI. CONCLUSIONS

It was shown that the 1PRAM does function as expected by TRW's and Motorola's specifications at a low speed (the PMX generated a SYSCLK signal of approximately 150 Hz in real time). The hardware problems mentioned above are particular to one IC and most likely a manufacturing glitch due to the new technology implemented. However, it was also shown that problems do exist in the vectors generated by TRW. These vectors were altered, and the correct outputs were obtained.

With regard to the VHDL model, design errors were found in the code provided by TRW for the 1PRAM. Unless these problems are resolved, the VHDL model will be rendered useless to designers in future applications.

REFERENCES

Adlhoch, Richard, et al., (1987), "Architecture Specification: 1-Port Memory Macrocell," TRW Report, October 1987.

Adlhoch, Richard, et al., (1989), "Final Design Report: 1-Port Memory Macrocell," TRW Report, July 1989.

Adlhoch, Richard, et al., (1989), "Test Plan: 1-Port Memory Macrocell," TRW Report, August 1989.

"Logician Design Compilation," (1986), Daisy Systems Corporation, California.

"Logician Verification Support System II," (1986), Daisy Systems Corporation, California.

Sunamoto, Steve, (1989), "CPUAX Superchip Design," TRW VHSIC Semiannual Review, TRW Report, July 1989.

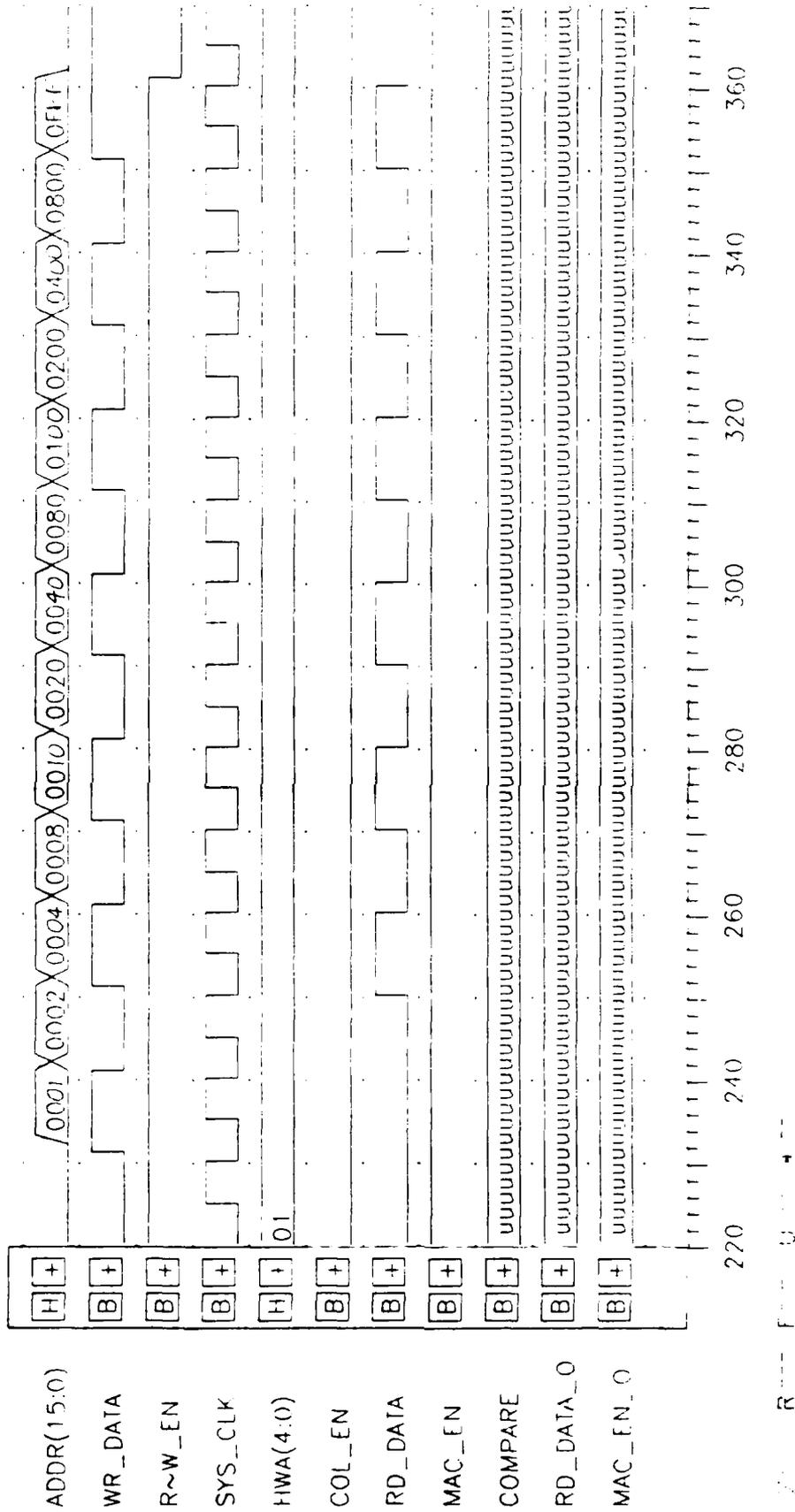
NOTES

- 1 "(CR)" indicates a carriage return.
- 2 Brigham Rees is a test engineer for Motorola who also performed tests on the 1PRAM.
- 3 Richard Adlhoch, et al., "Test Plan: 1-Port Memory Macrocell," (TRW Report, 1989)

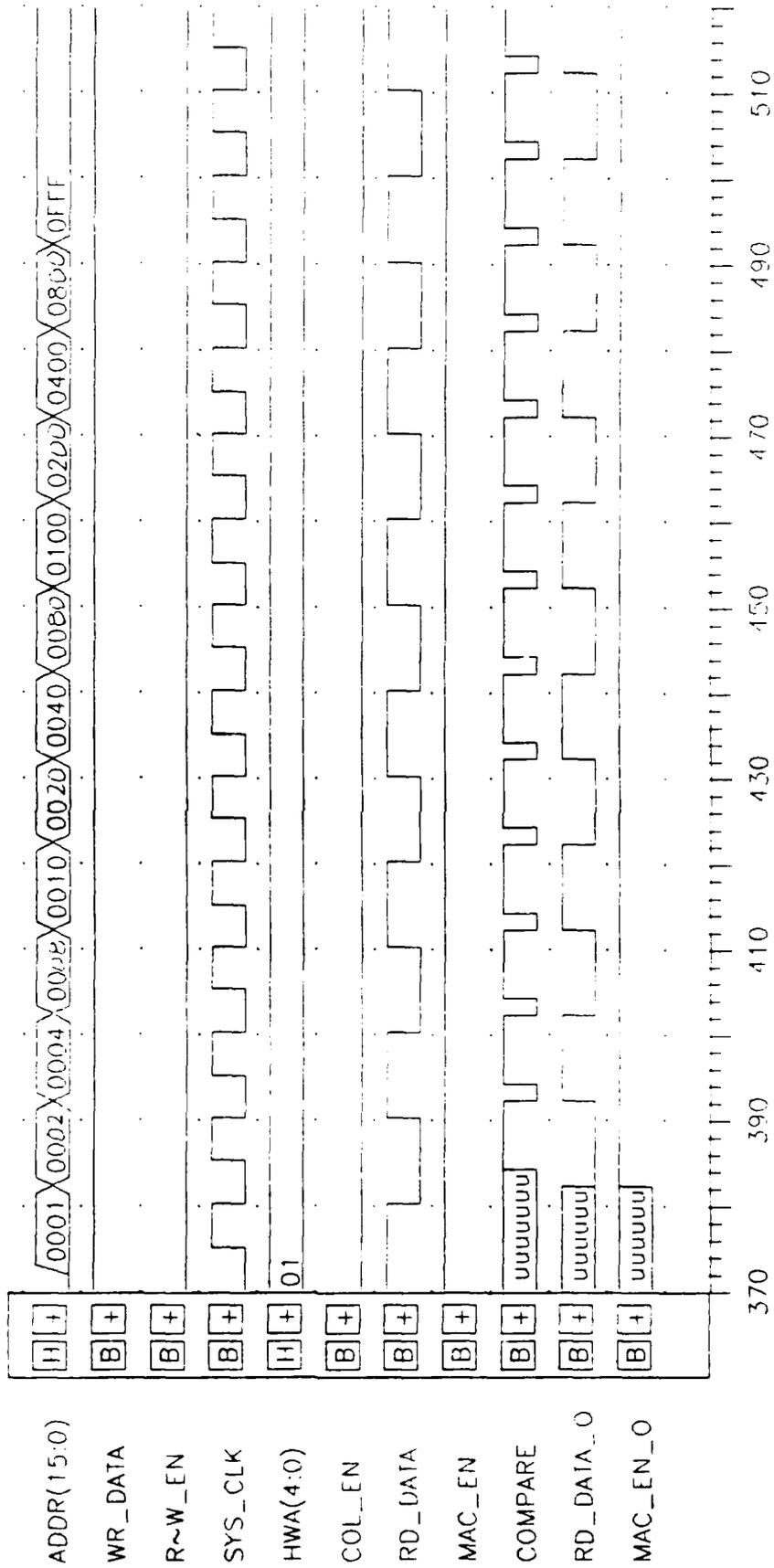
APPENDIX

This appendix contains the waveforms created by executing the tests considered. They are rendered below in the same order as explained above. Please note that a recurring "u" means that the signal(s) is unknown at that instant. The scale time is given in nanoseconds (simulation time), and all signal values are given in hexadecimal format.

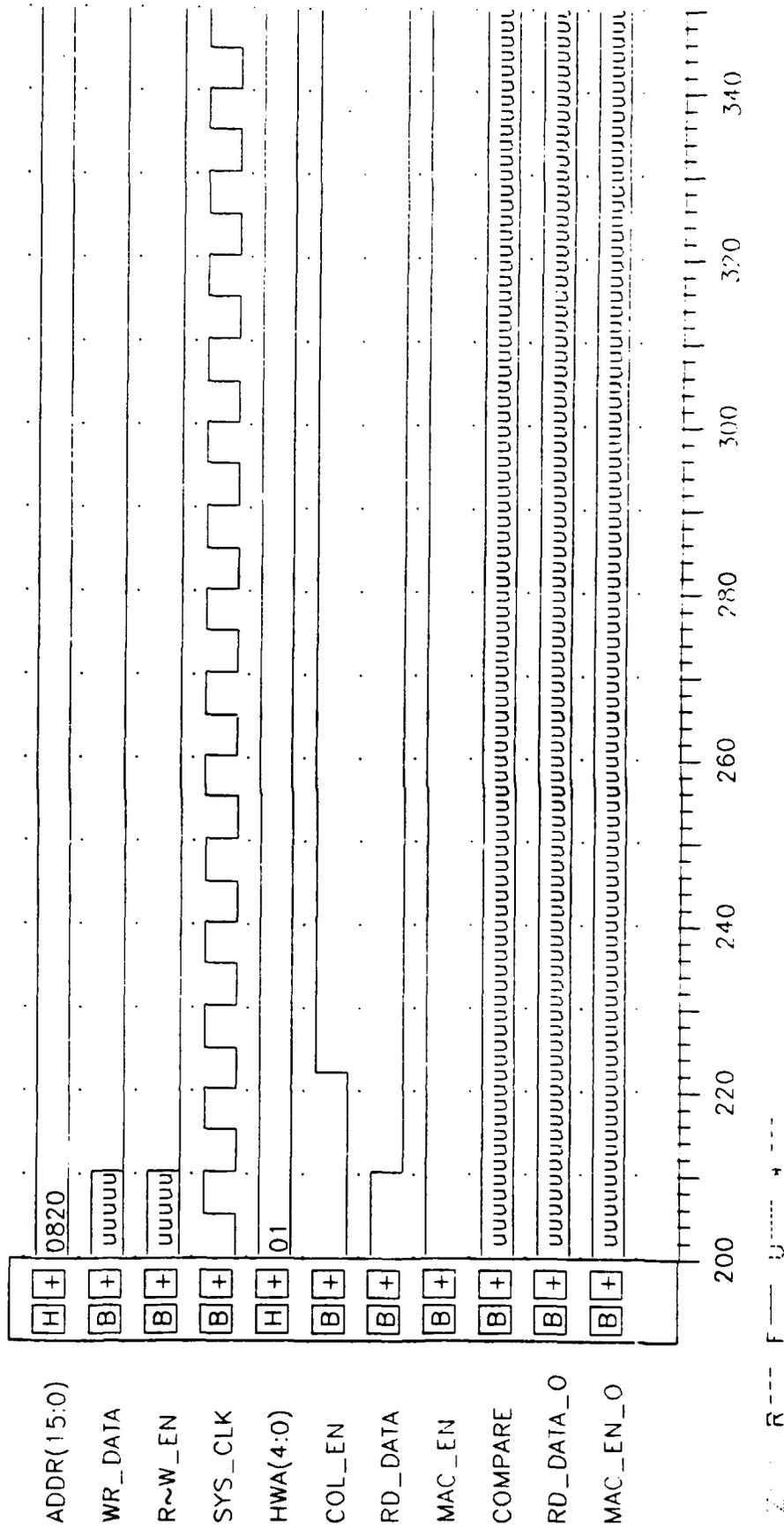
With regard to RD_DATA_O and MAC_EN_O, these signals may seem nonsensical at times. This is due to the additional timing added to the original vectors received from TRW. Recall that these additions were made so that the desired responses would occur.



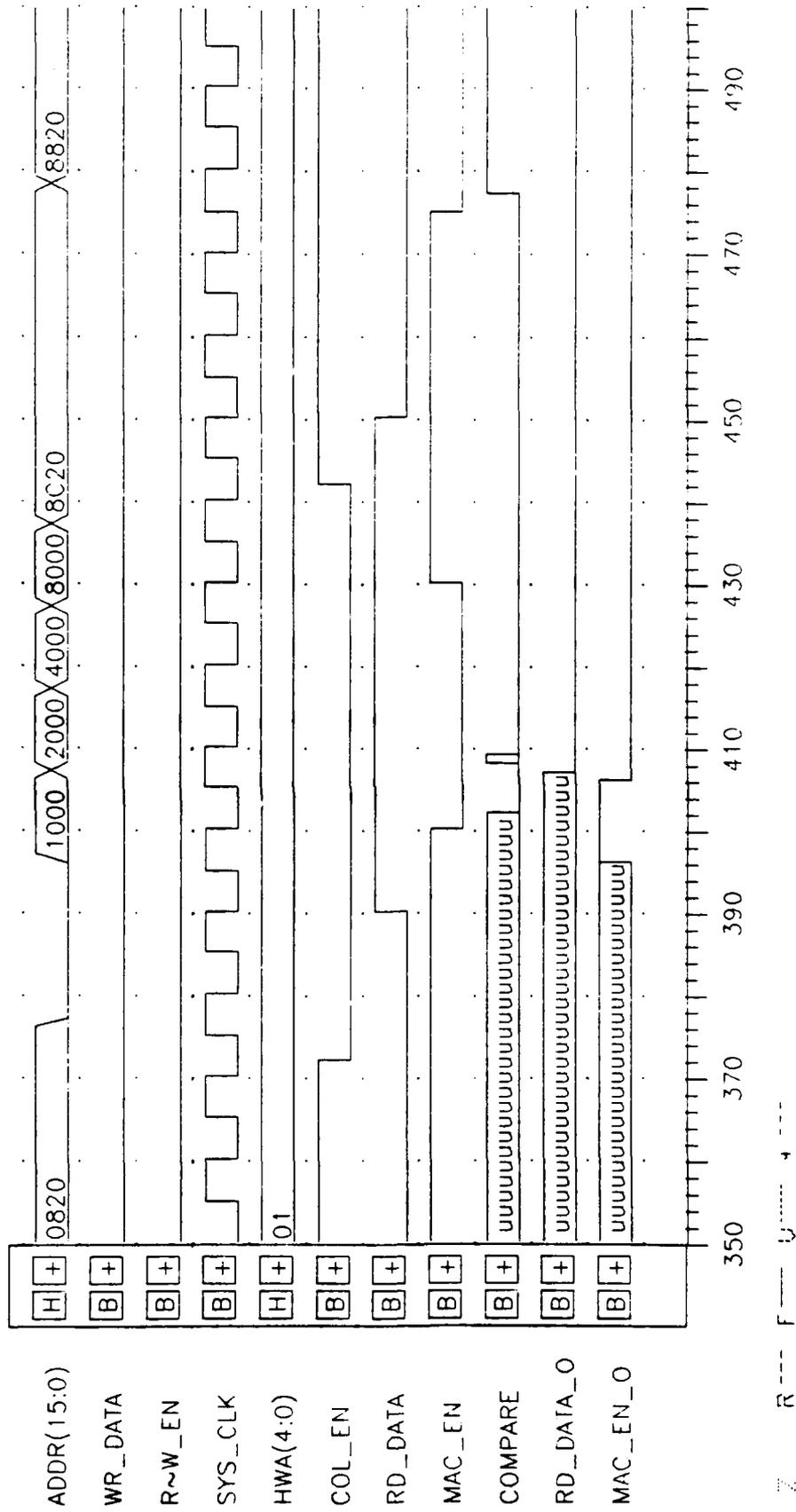
TEST_ADEC (220-370)



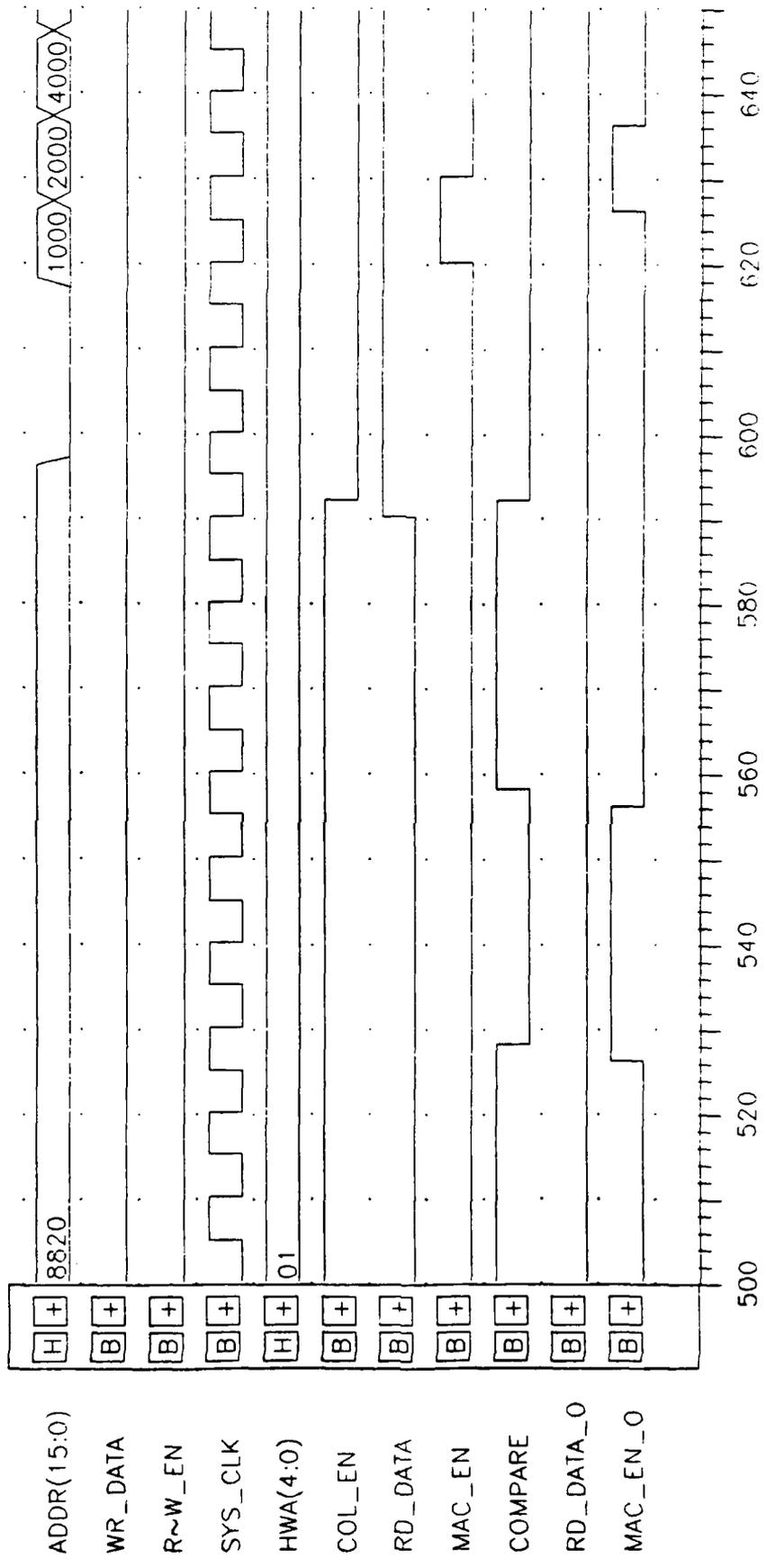
TEST_ADEC (370-520)



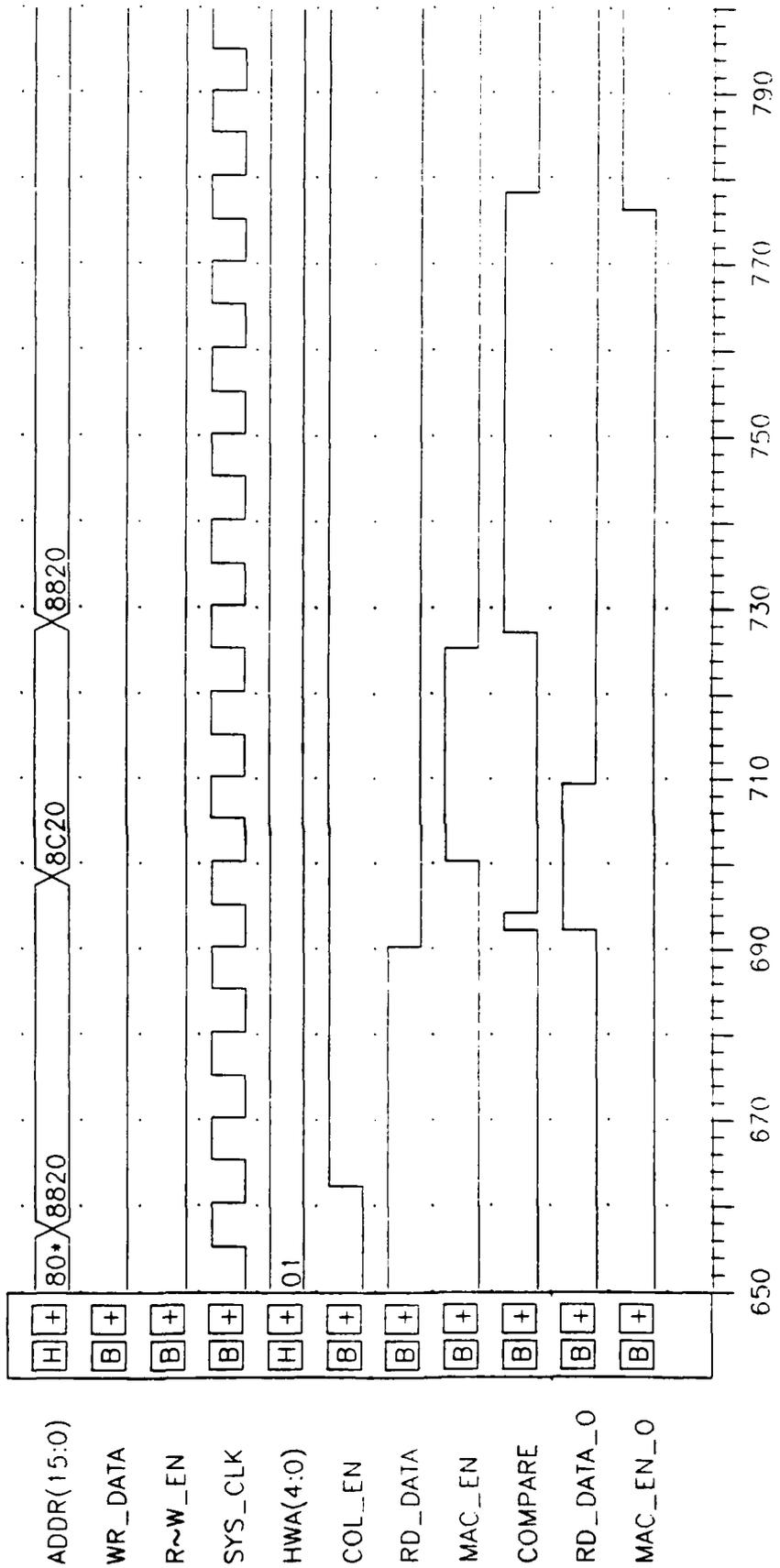
TEST_PDEC (200-350)



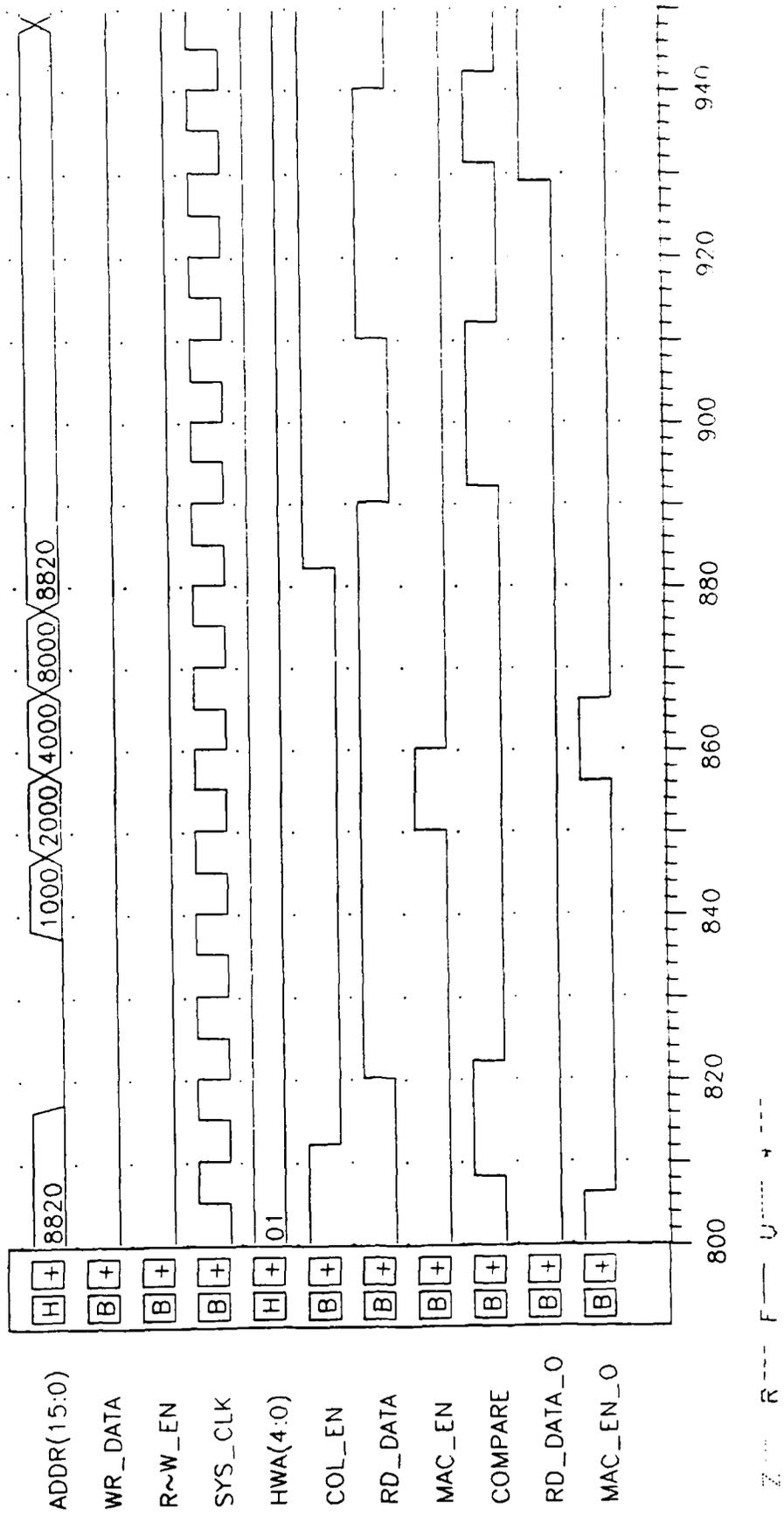
TEST_PDEC (350-500)



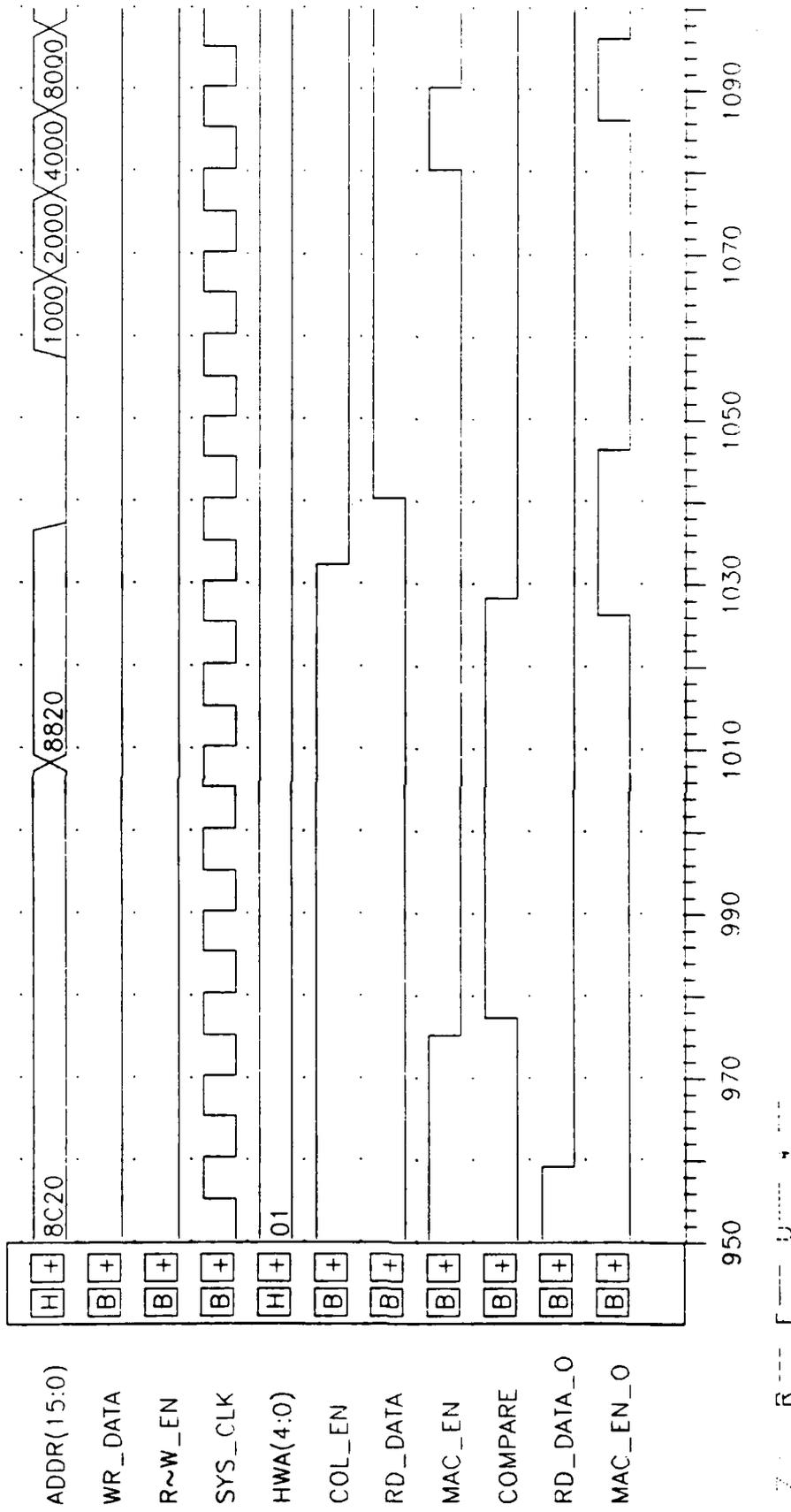
TEST_PDEC (500-650)



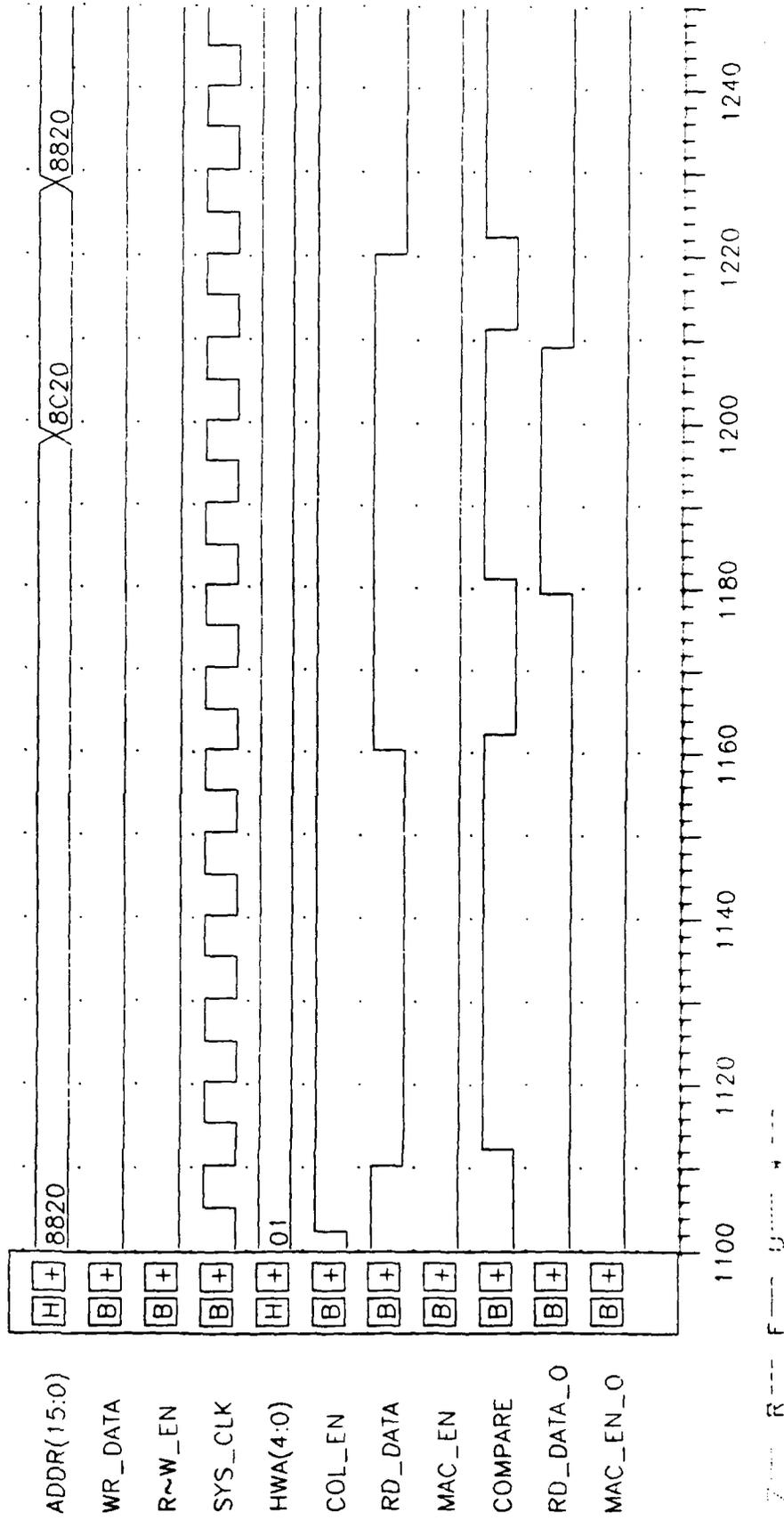
TEST_PDEC (650-800)



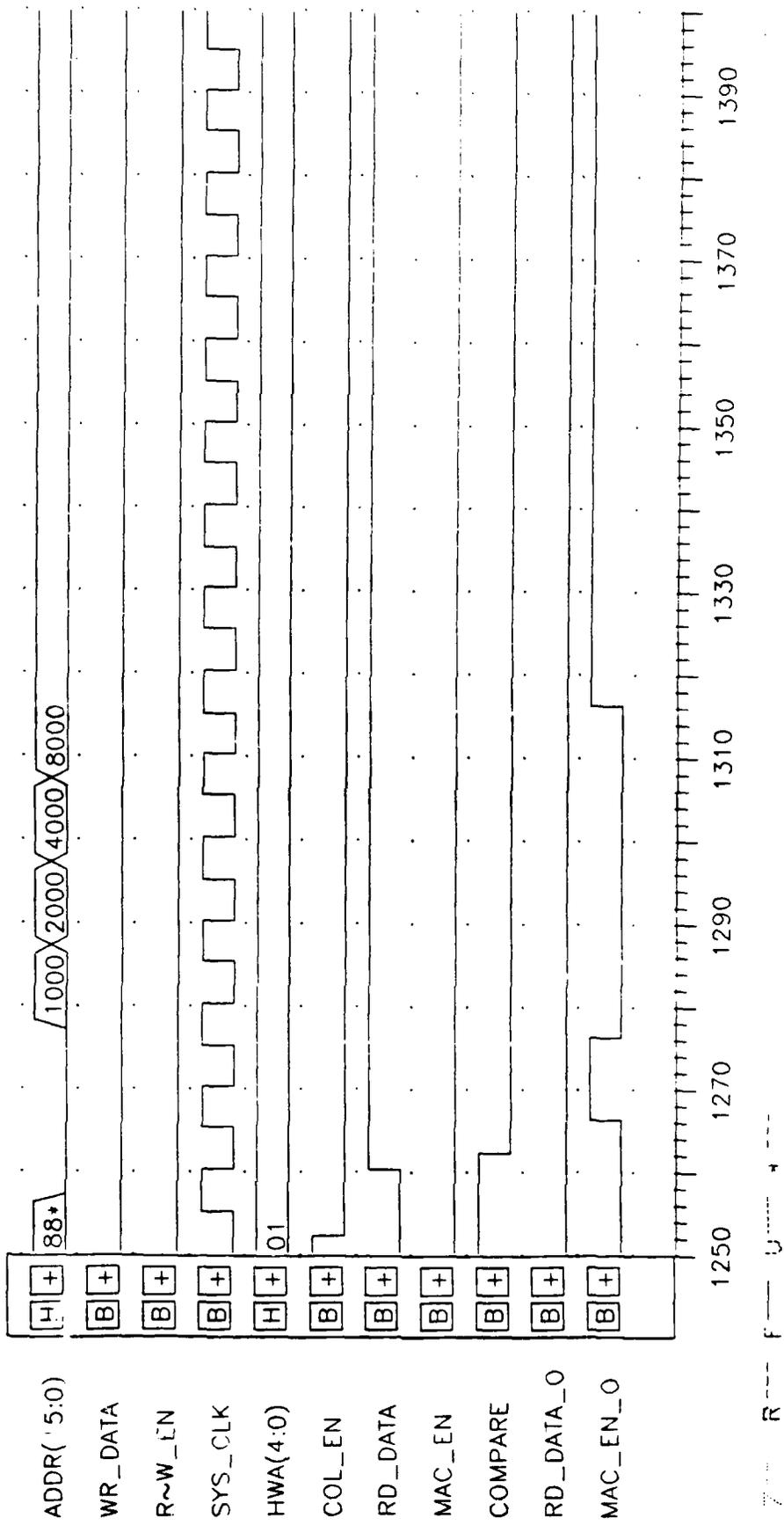
TEST_PDEC (800-950)



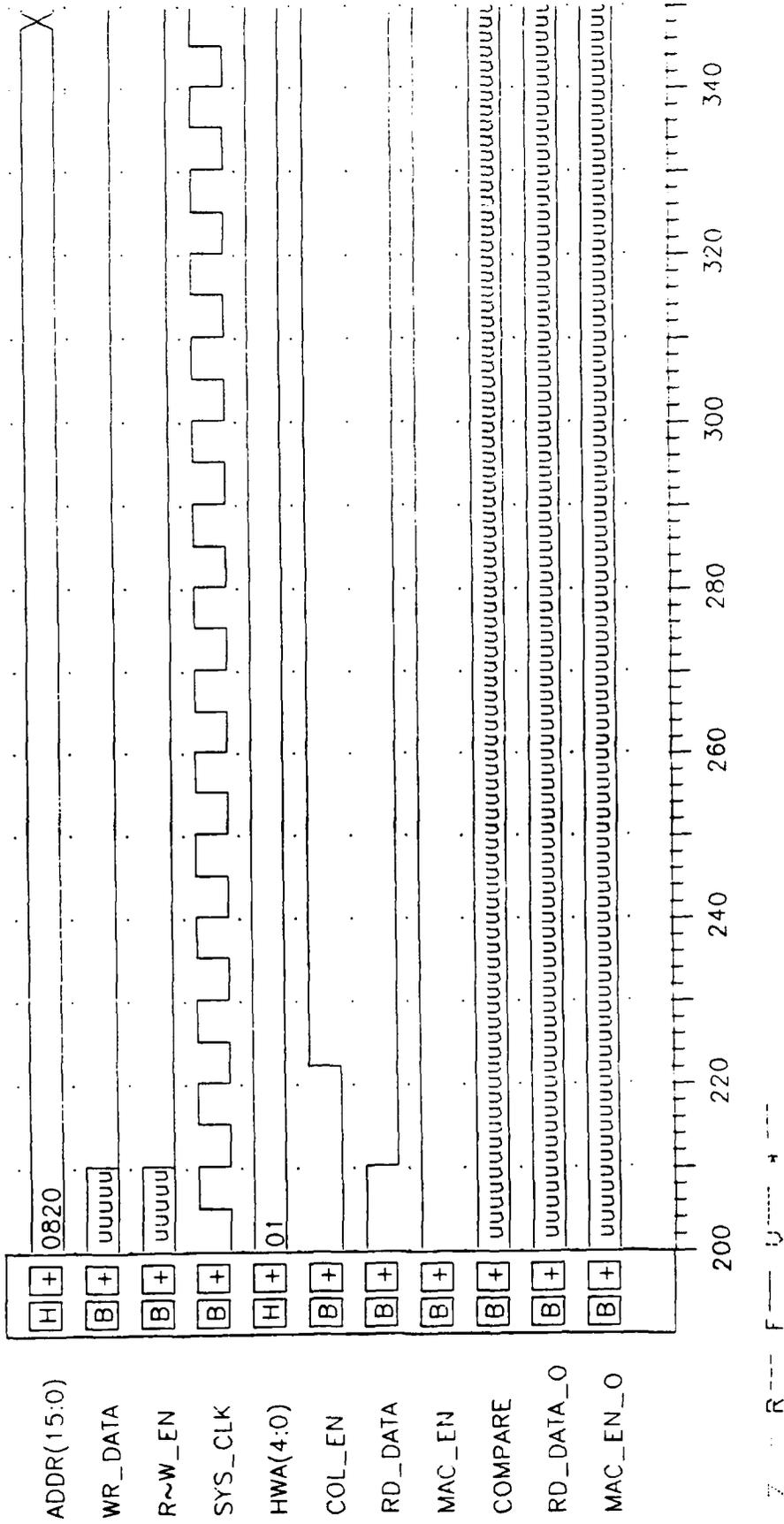
TEST_PDEC (950-1100)



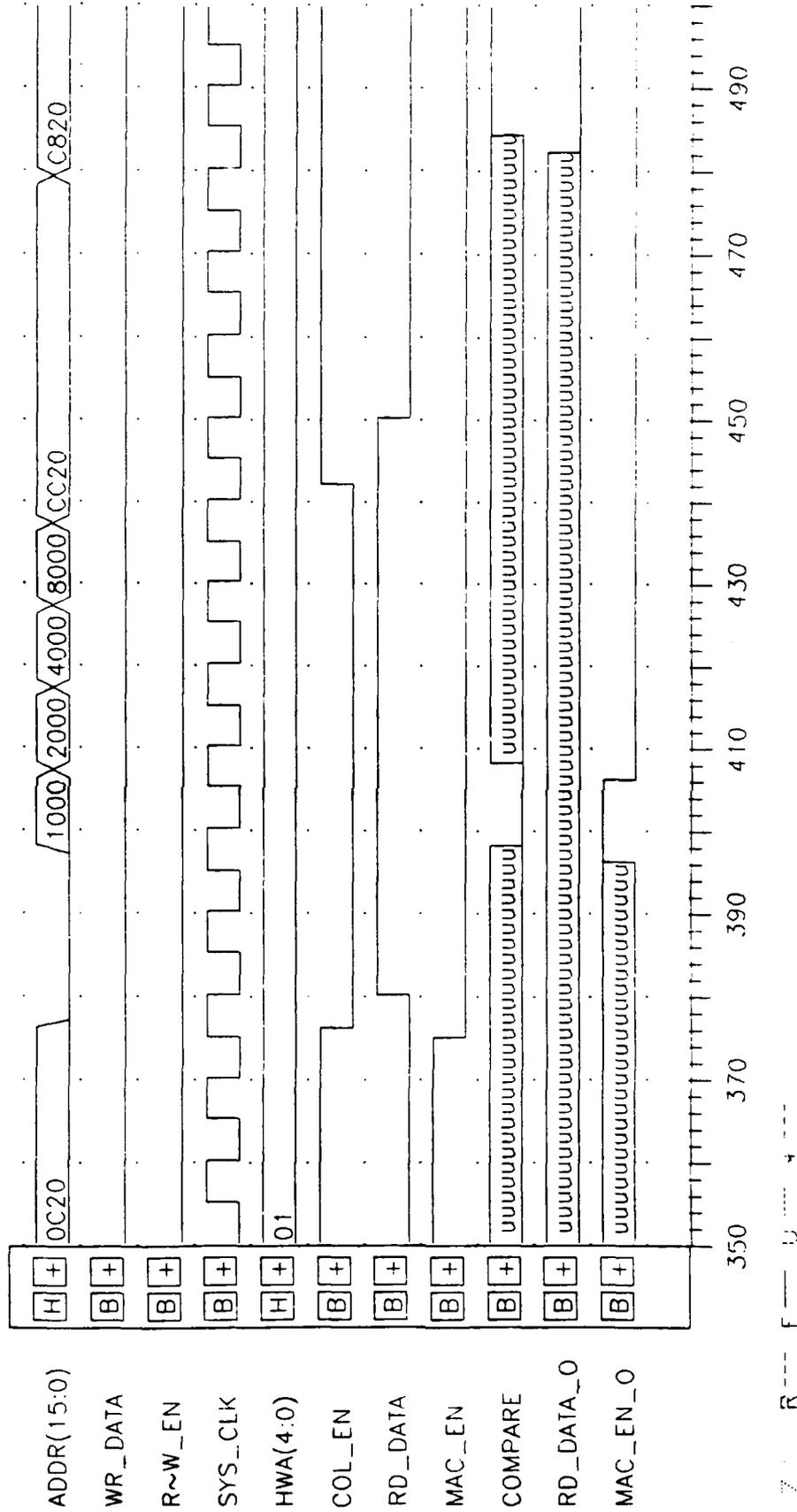
TEST_PDEC (1100-1250)



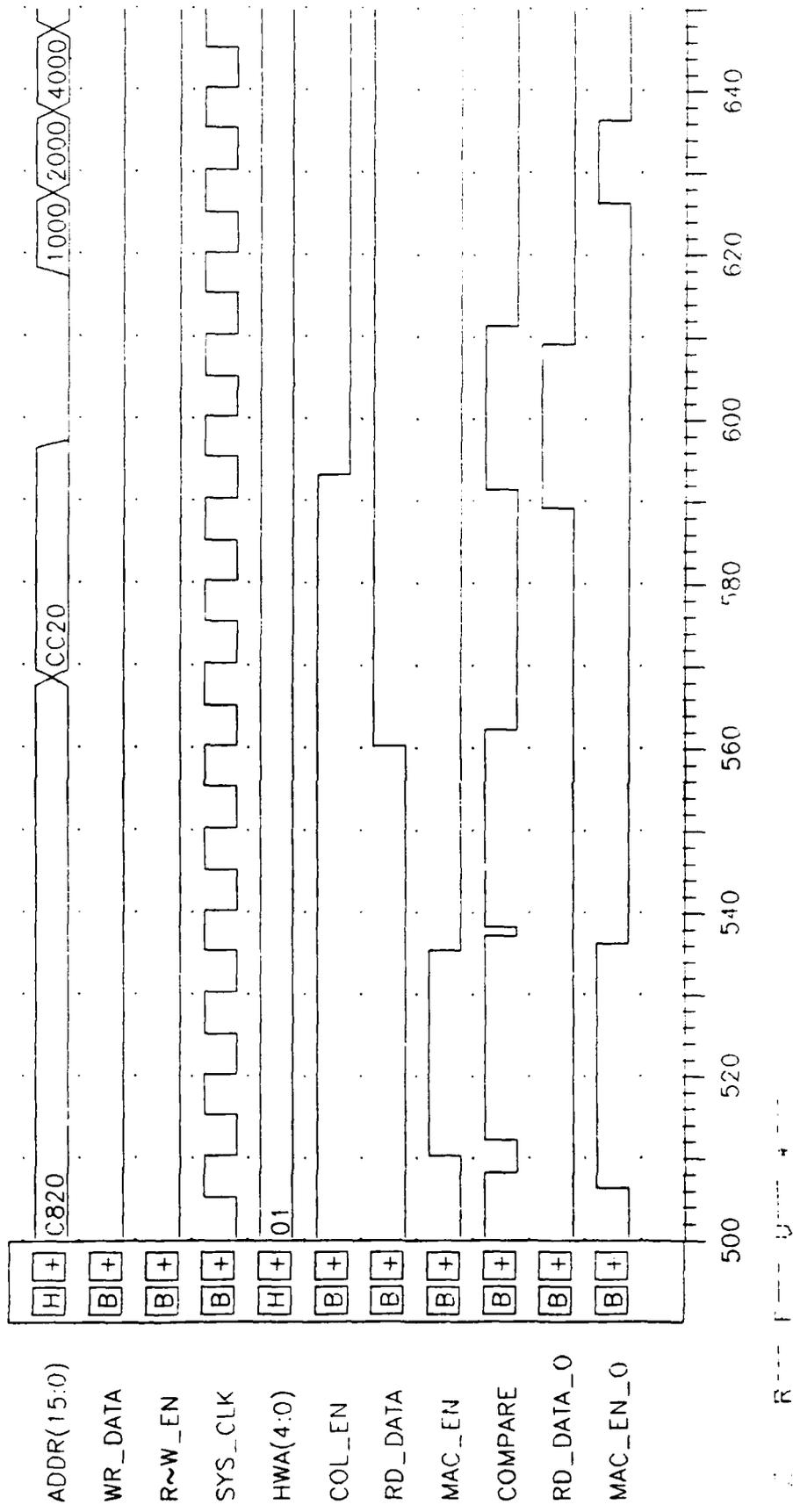
TEST_PDEC (1250-1400)



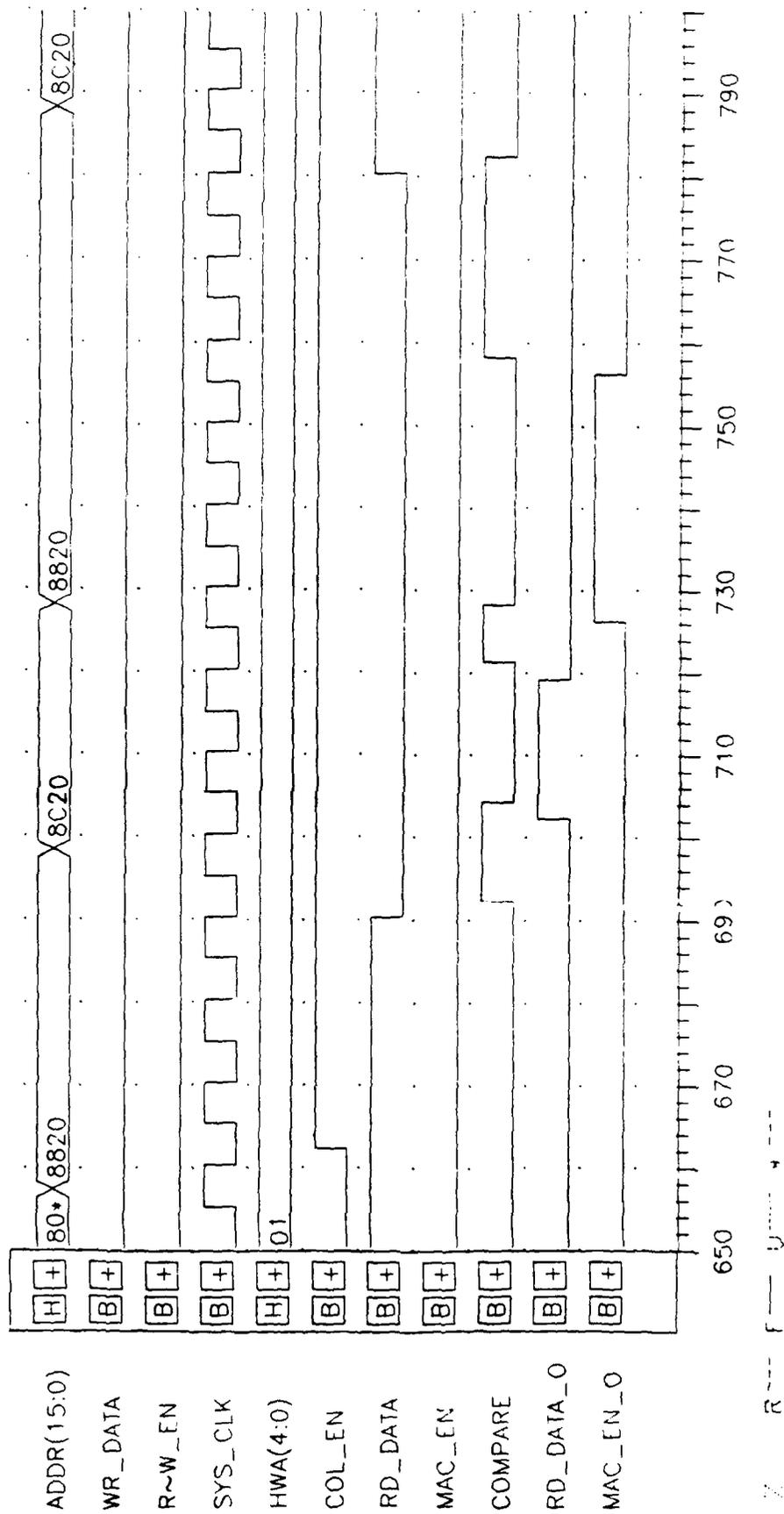
TEST_PROFF (200-350)



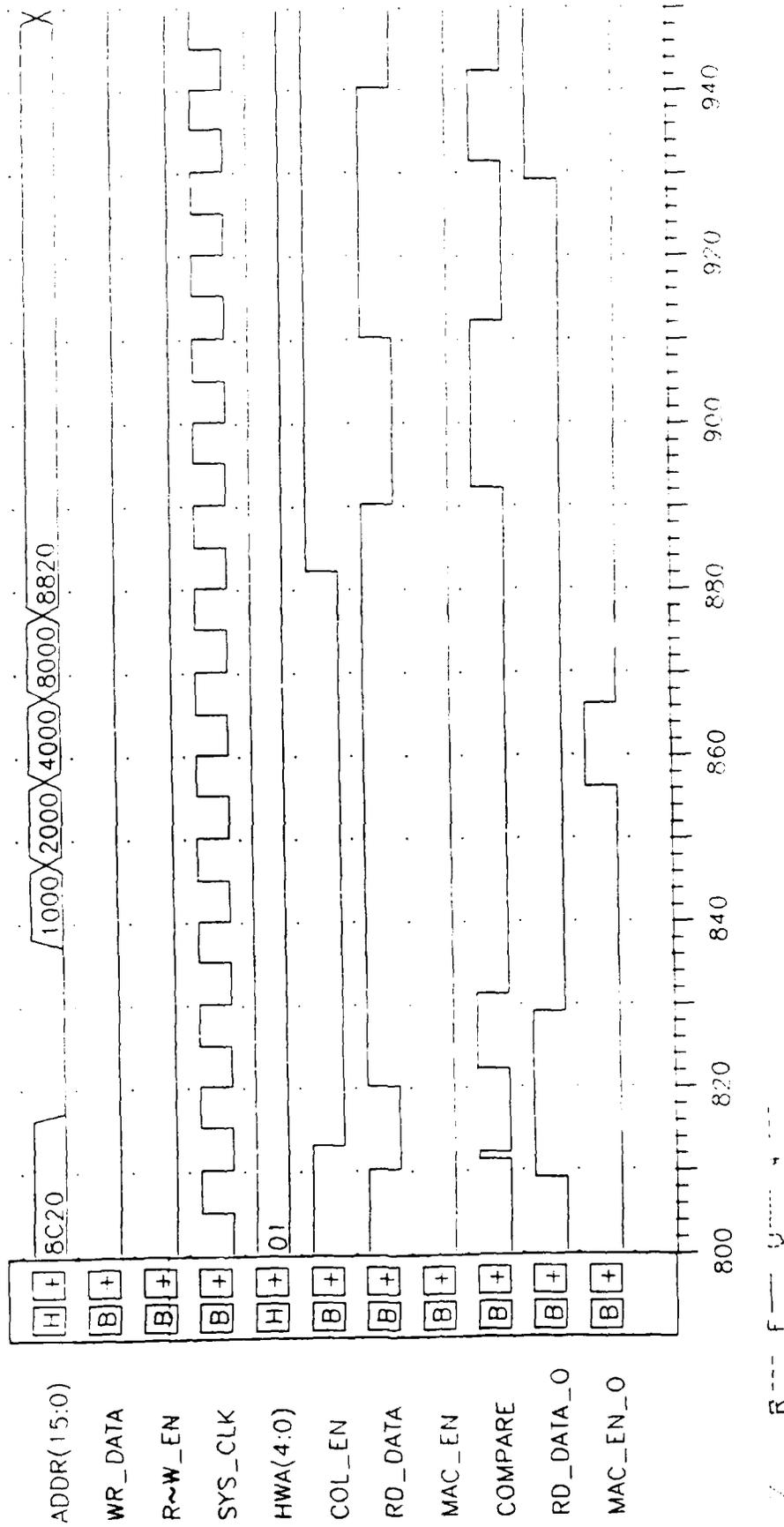
TEST_PROFF (350-500)



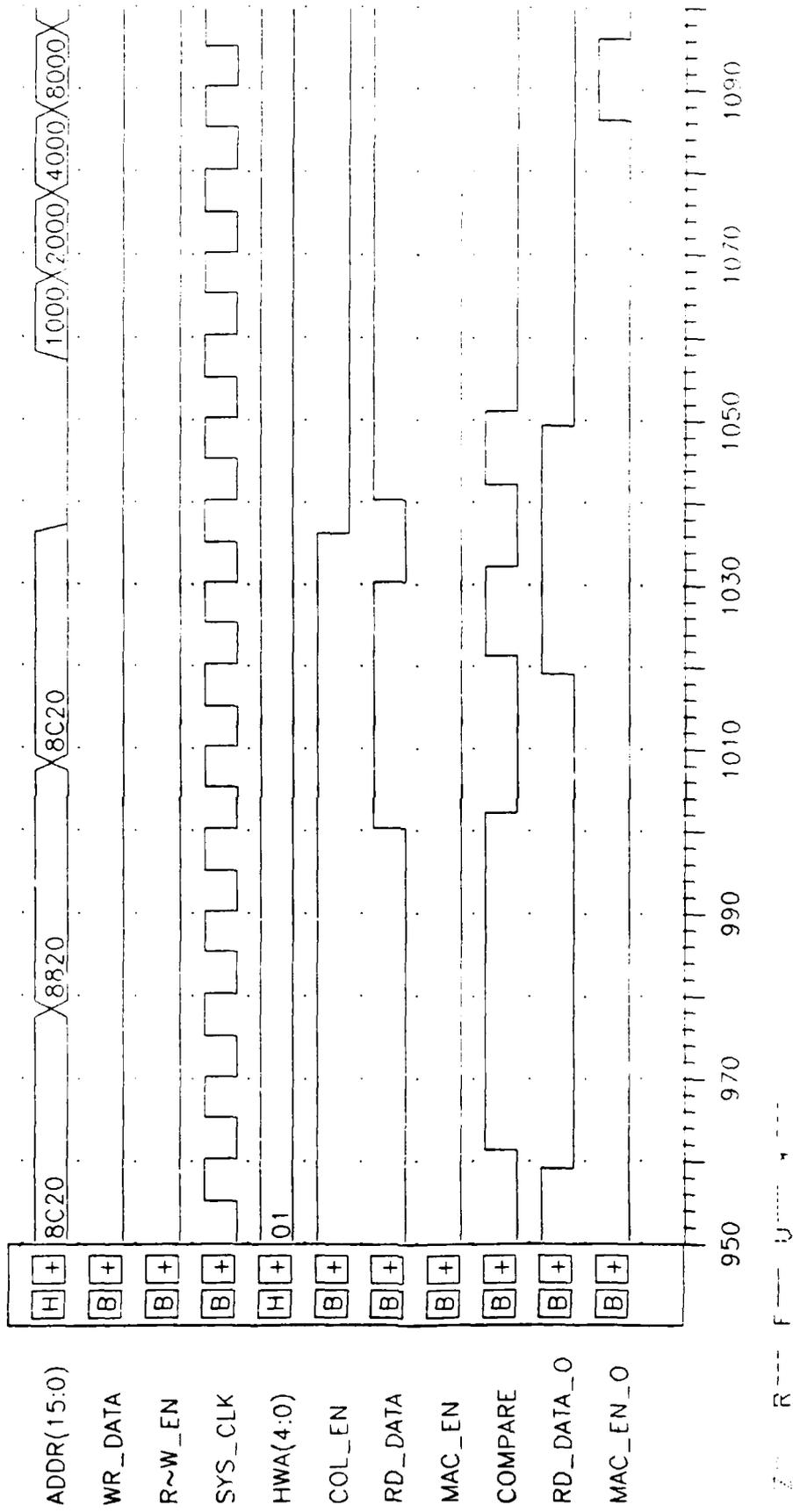
TEST_PROFF (500-650)



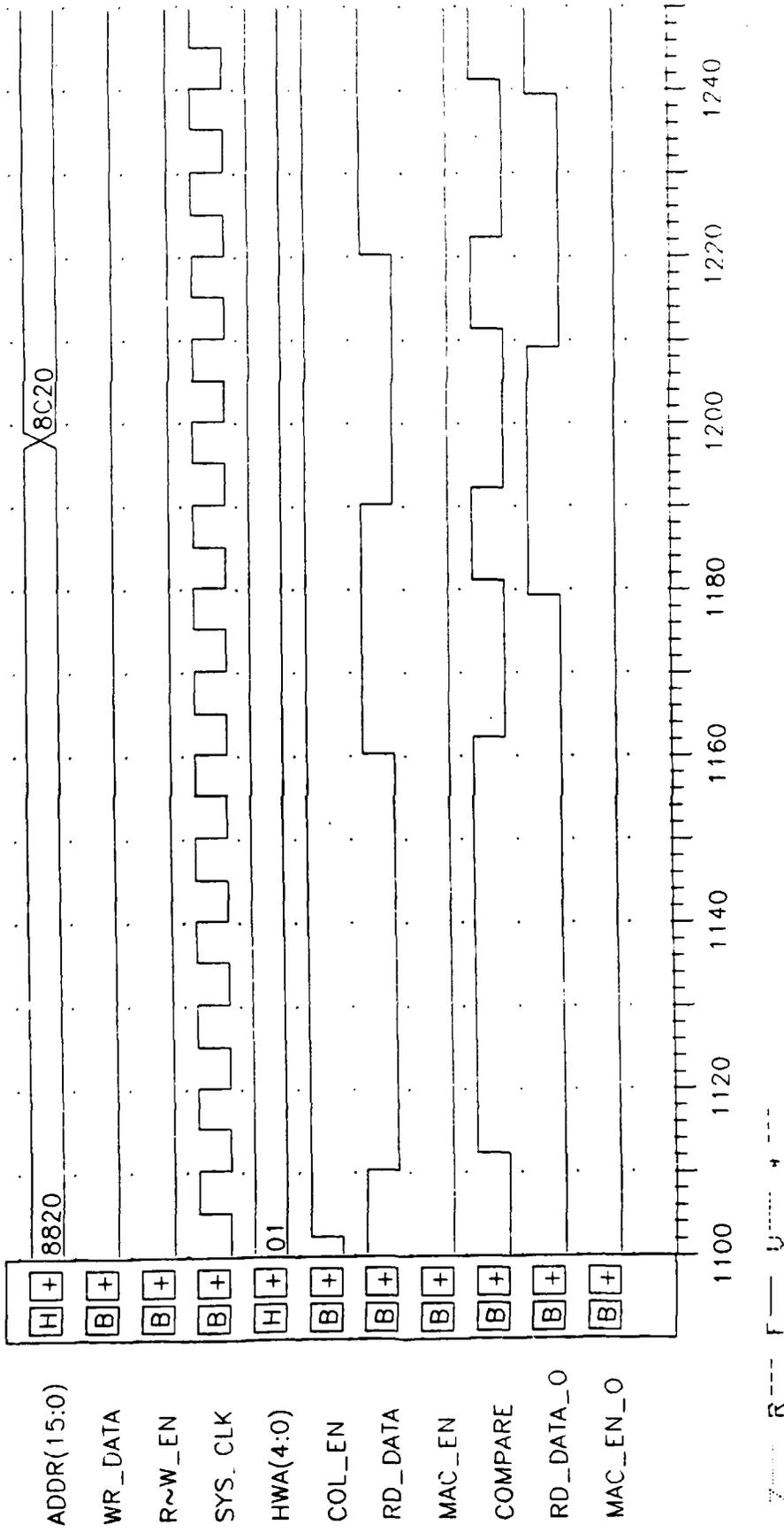
TEST_PROFF (650-800)



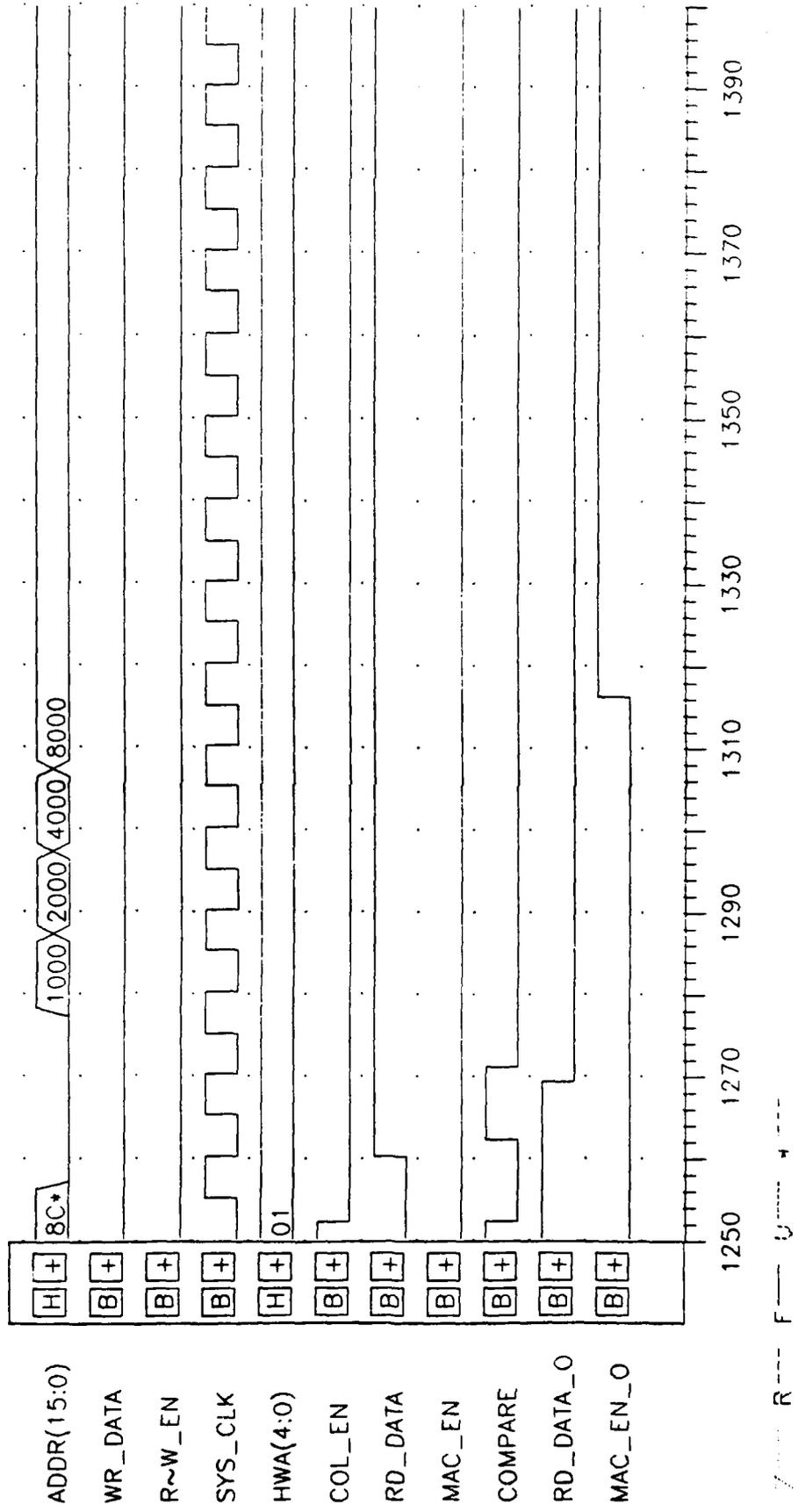
TEST_PROFF (800-950)



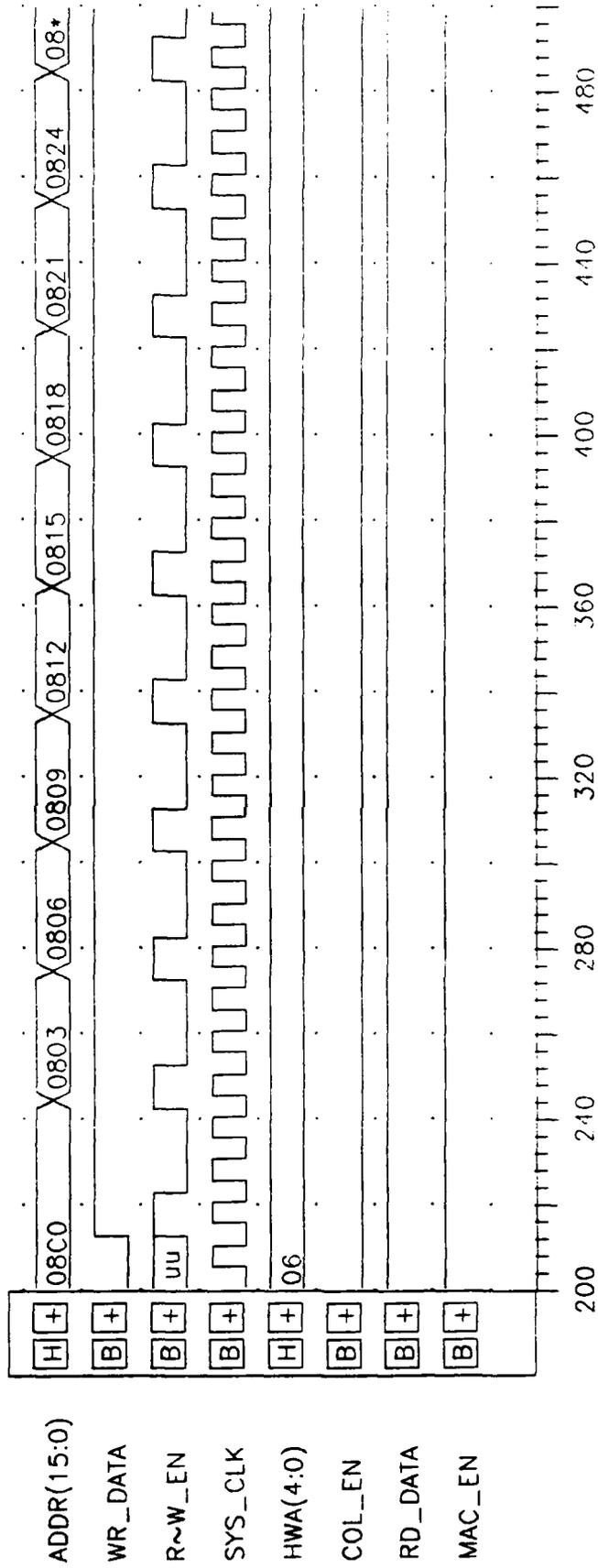
TEST_PROFF (950-1100)



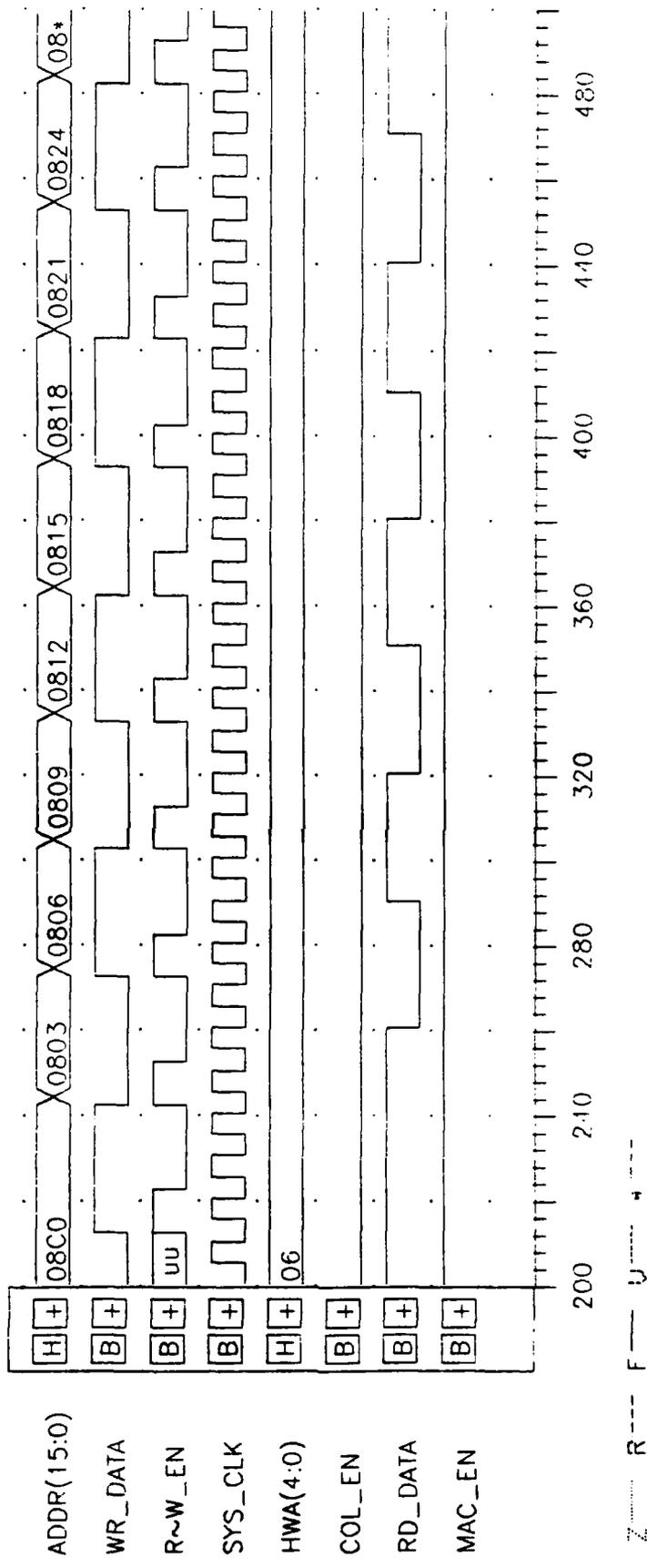
TEST_PROFF (1100-1250)



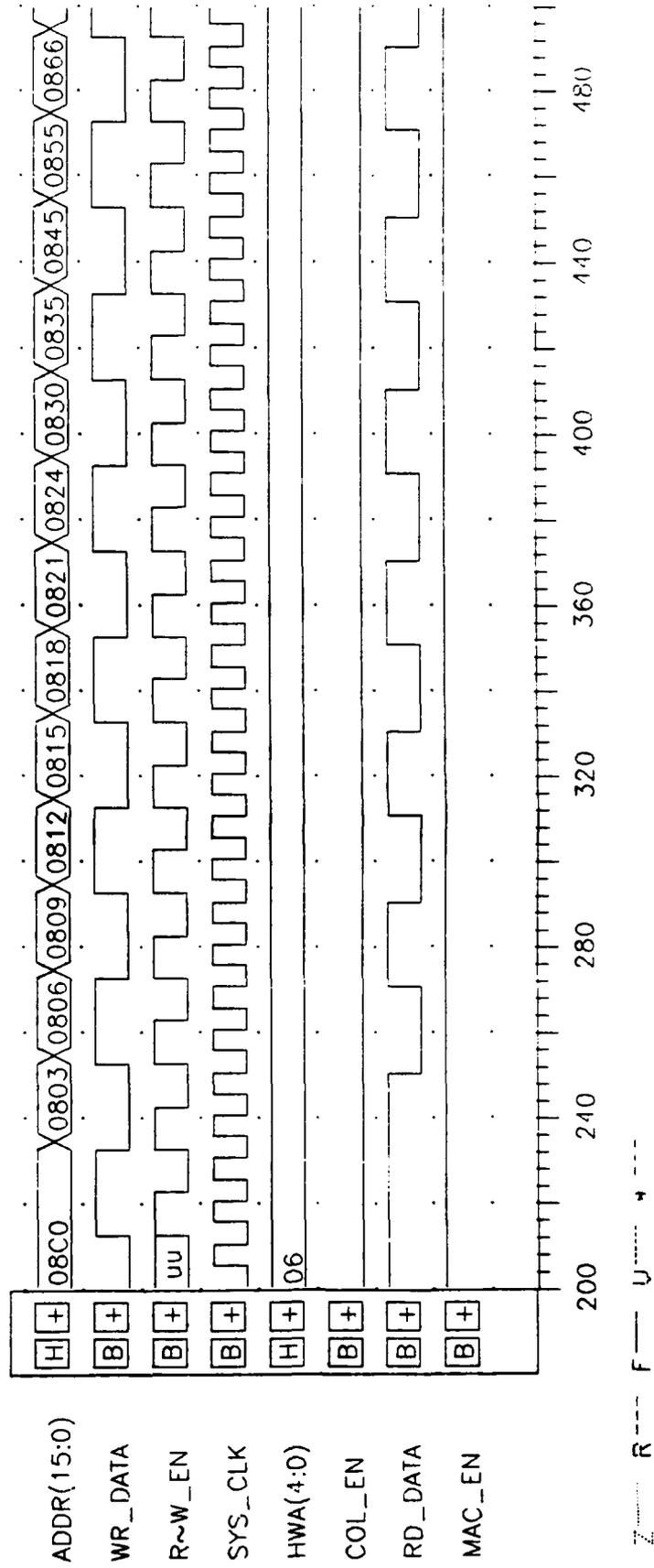
TEST_PROFF (1250-1400)



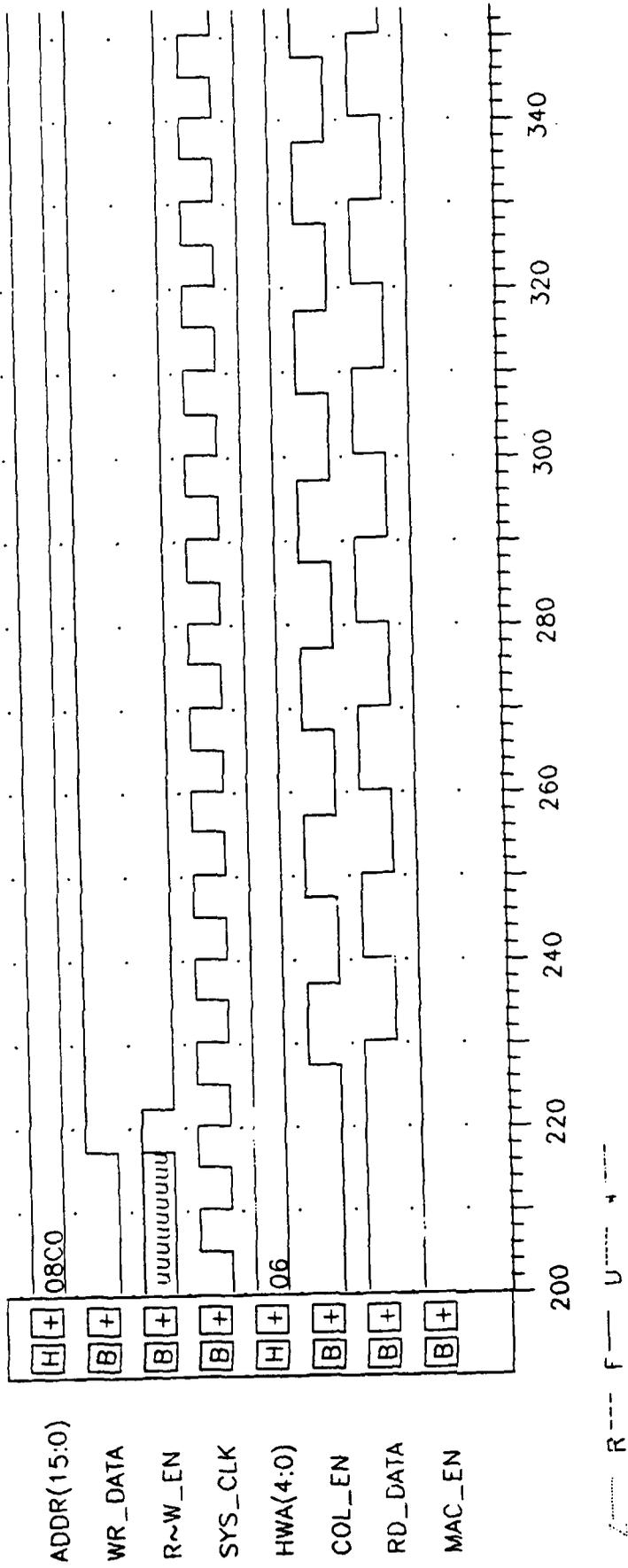
TEST_RWEN.1



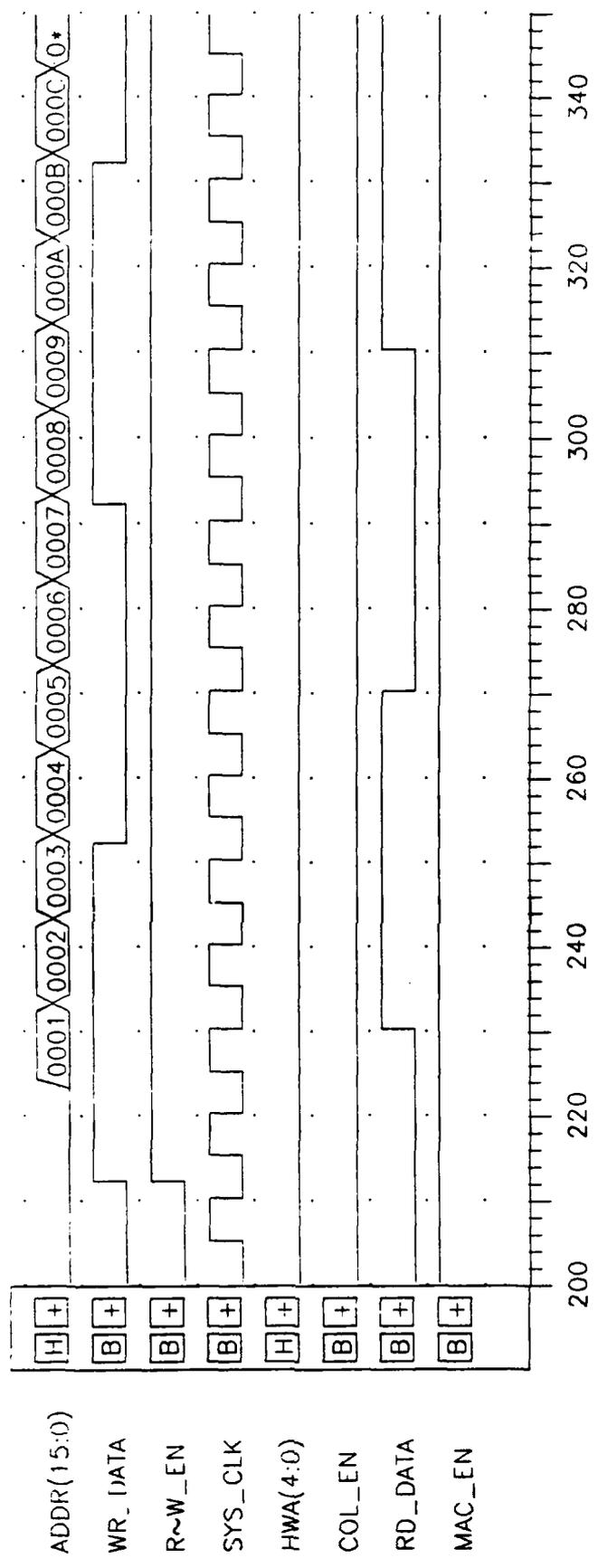
TEST_RWEN.2



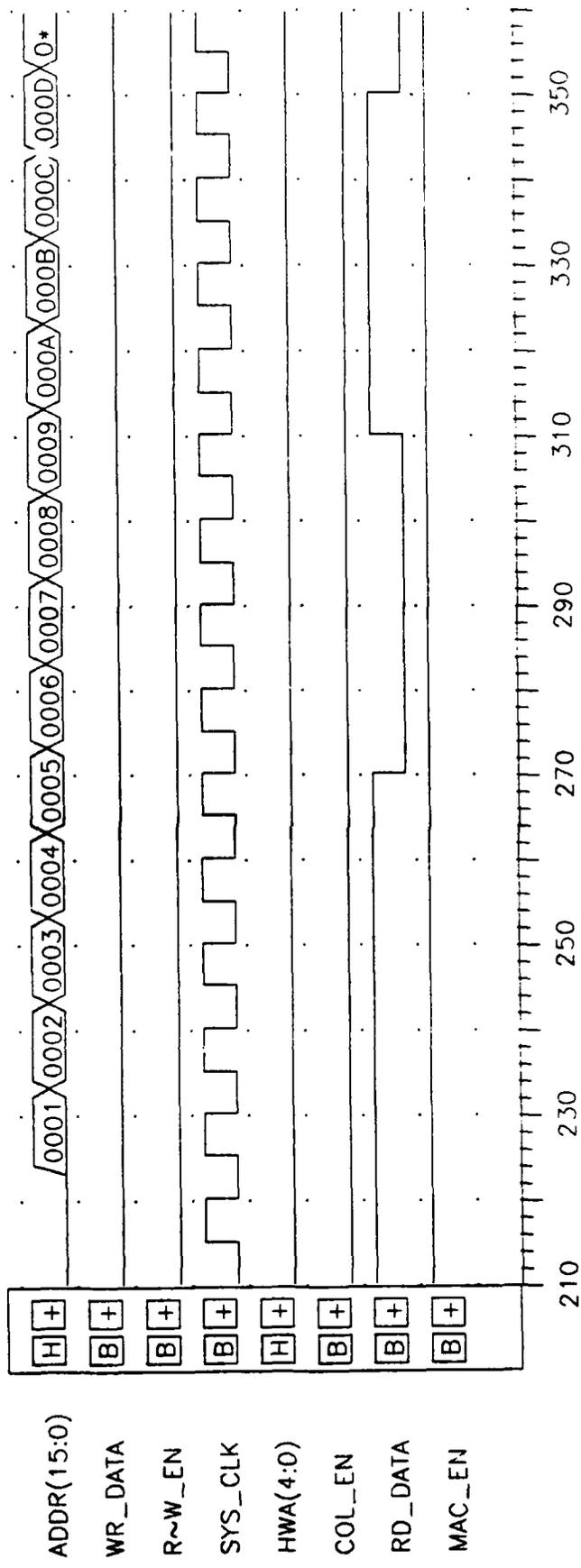
TEST_RWEN.3



TEST_OUTMUX

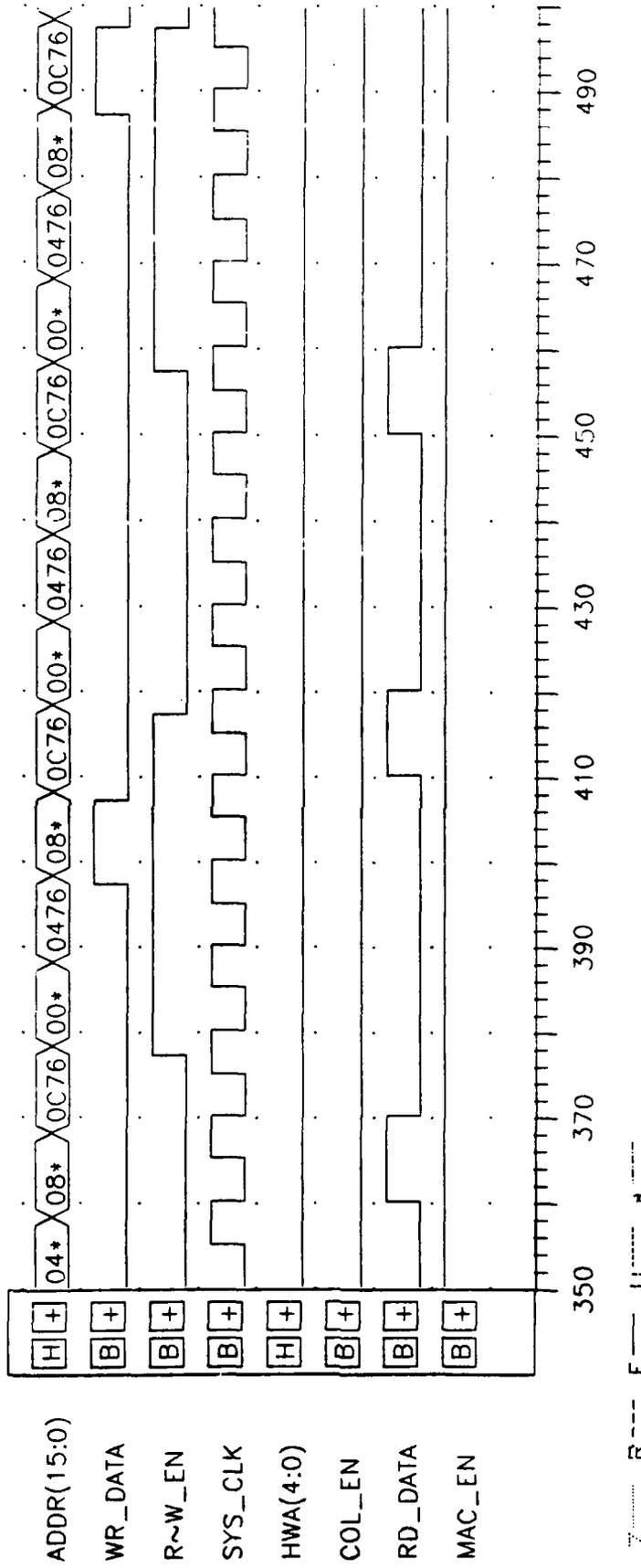


TEST_WRITE

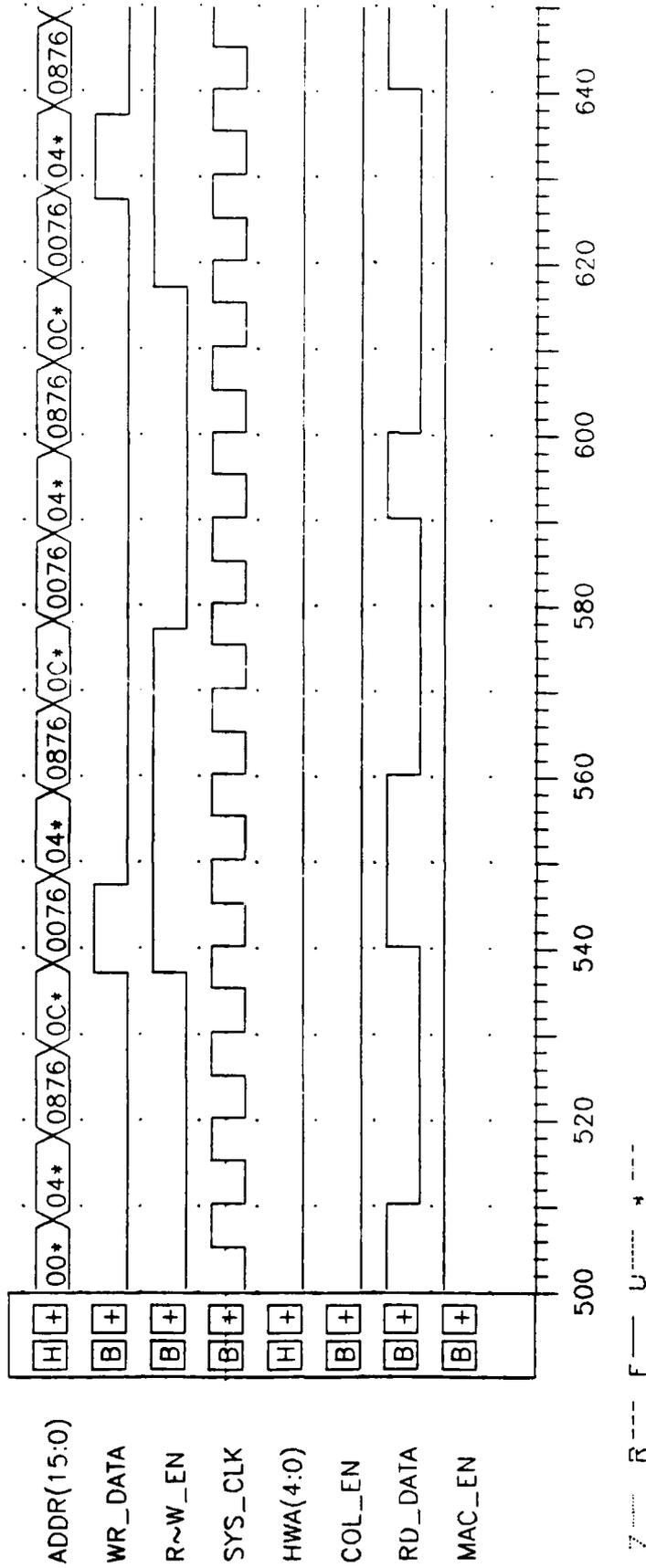


Z R F U

TEST_READ



TEST_QUAD - Walking Ones
(350 - 500)



TEST_QUAD - Walking Ones
(500 - 650)