

3.6 Digital Blocks and RAM block

In the subsections of this section, we will examine in detail all the digital blocks and RAM block in the VQ system. There are one RAM block and 3 functional digital blocks named ‘Control block’, ‘Interface block’ and ‘Encoder block’. First, we start analyzing the digital blocks. System clock frequency is 2MHz. The determination of the clock frequency will be explained later in Chapter 4.

3.6.1 Control Block

The block named ‘Control Block’ is responsible of refresh procedure of the VQ system. Refresh procedure can be summarized as follows:

1. Waiting until the refresh time which is determined by Interface block (The second digital block)
2. Submit the control signal sequentially to RAM block containing template-vectors.
3. Activate sequentially DCELL row for writing template voltage.
4. Submit the control signal to the switch of the IN_AMP blocks switching its input from input pads to DACs’ outputs during refresh period.

During the design, the following criteria are considered

1. Refresh one DCELL row at each clock cycle.
2. Properly activating and deactivating rows for avoiding wrong template writing
3. Avoiding reading and writing conflict to Dual Port RAM. (This is resolved by simply using different clock edge on different input port of RAM block.)

Control Block is designed using VERILOG hardware description language. Let us now briefly examine the behavioral VERILOG code of the block, which can be found also in Appendix E.

Terminals' names and functions:

reset : This input terminal is used to initialize D type flip-flops in the block. System reset is active when signal is Logic 0.

clk : This is an input terminal bringing system clock signal.

Akt_Dac : This is an output terminal. It is Logic 1 when system refreshes the DCELL matrix. Otherwise it is Logic 0.

Ram_cs : This is an output terminal. It brings the control signal CS to the RAM block for reading the template-vectors in refresh period.

ref_time : This is an input data bus containing 5 bits. It contains the information about the time period between refresh period (refresh time).

Ram_adres: This is an output bus containing 6 bits. It is connected to RAM block's address bus.

Row_akt : This is an output bus containing 64 bits. Each bit of the bus is connected to the respective DCELL row write activation signal V_{ϕ} of the matrix.

1. Operation Principles

Terminal Akt_Dac is connected to the all IN_AMP blocks for controlling input signals switching of these blocks between DACs' outputs and system input vector pads. After the block is initialized by reset signal, it starts to count until an internal counter is equal to the refresh time that is derived from the data in ref_time bus. In order to decrease number of bits required for programming the refresh time, a simple DECODER function added into the block, which decode ref_time data. Thus we can vary refresh time from 1 to $(2^{32}-1)$ clock cycles. This configuration is preferable also for rapidly changing written template-book on DCELL matrix when needed. When internal counter arrives to the refresh time, block enters to the refresh period. Thus first, Signal Akt_Dac becomes Logic 1 meaning inputs of IN_AMP blocks are switched to DACs' outputs. Then, RAM block start to receive Ram_cs signal at every rising edge of the clock signal. Control block changes the activated DCELL row and the address bus of the RAM

block sequentially every falling edge of the clock signal. Refresh operation of each DCELL row takes only 1 clock cycle. Thus all of entire matrix is refreshed within the 64 clock cycles. After refreshing phase, DCELL matrix restarts to evaluate input vectors. Thus control signal Akt_Dac becomes Logic 0 and internal counter restart to count until refresh time.

The Behavioral VERILOG model is synthesized using SYNOPSIS®. During the synthesize process, small silicon area was our primary objective. AMS 0.6 μm standard cell library HRDLIB is used. And finally, synthesized schematic contains 491 standard cells. We can not show the schematic view of the Control block due to its large cell count. Yet, VERILOG NETLIST of the block can be found in Appendix E. VERILOG Simulation result of Control block can be seen in Figures 3.77a and 3.77b

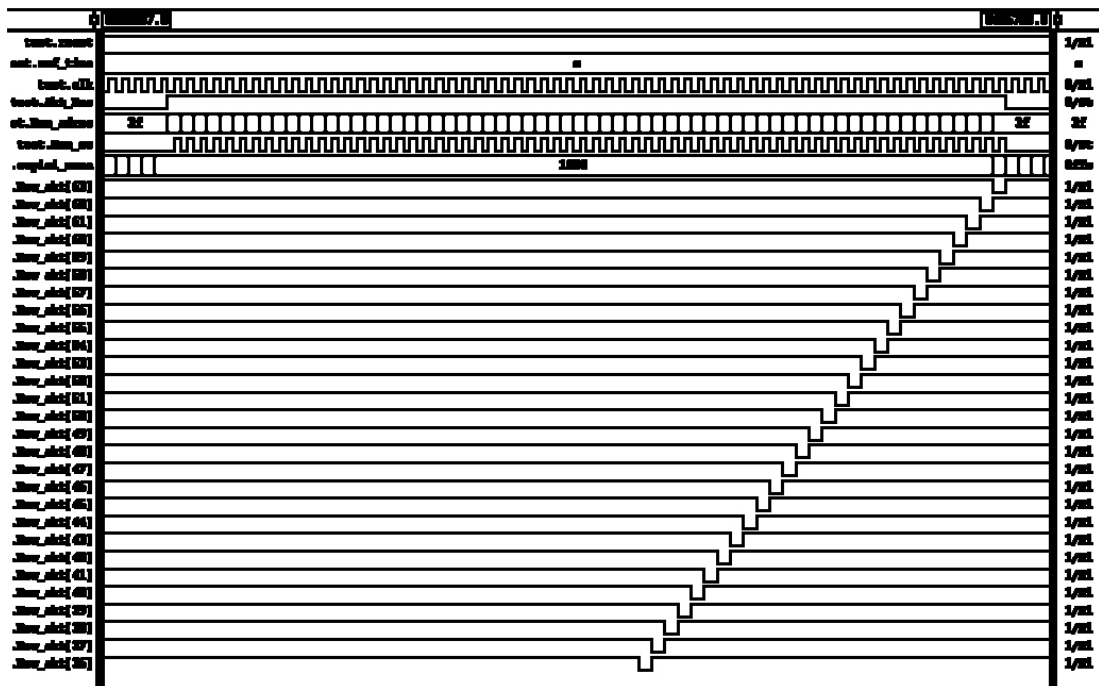


Figure 3.77a :Verilog simulation result 1 of Control block

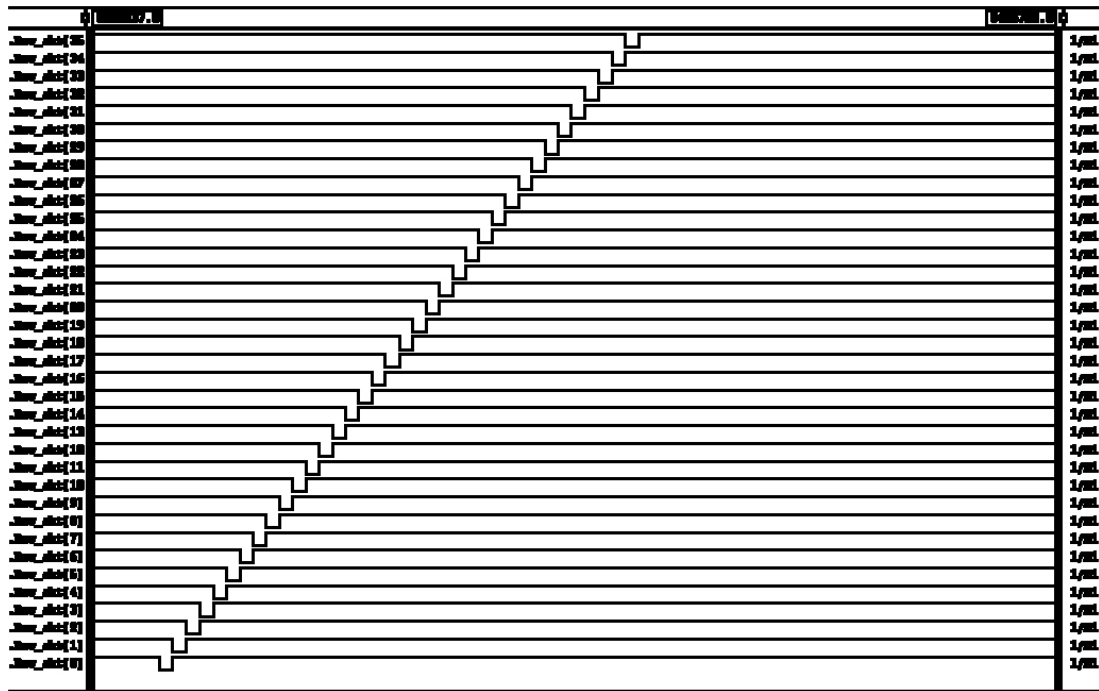


Figure 3.77b : Verilog simulation result 2 of Control block

In Figure 3.77a, sayici_zama represents internal counter containing 32 bits. As we can note easily, internal counter count until the refresh time (1000) derived with decoding the data of ref_time equal to c. Then the signal Akt_Dac becomes Logic 1. Control Block starts to change the RAM block's address bus (Ram_adres), activates DCELL row (Row_akt bus's bits in Figures 3.77a and 3.77b) and read the RAM content (Ram_cs) sequentially until all DCELL rows are refreshed. After refresh period, signal Akt_Dac becomes Logic 0 and internal counter restart to count.

3.6.2 Interface Block

Second digital block named 'Interface block' is responsible of communication with outside of Vector quantizer system and determination of refresh time. These procedures are:

1. Reading (Writing) digitized template-book contents from (to) Dual Port RAM without read or write conflict.
2. Sending refresh time knowledge to Control block.

In order to facilitate communication with the block, asynchronous write and read procedures are preferred. Interface block is designed using VERILOG hardware description language. Before analyzing the Interface block, we must explain memory

organization of the system. Our template-book contains 64 template-vectors, each template-vector contains 16 vector elements and each vector element has 6 bits of resolution. Thus our memory requirement is a RAM having 64 word of 96 bits length. AMS provided us a dual port RAM of size 64word x 8bits (dpram64x8). Thus, we have used 12 dpram64x8 blocks to constitute the required system memory. Let us now briefly examine the behavioral VERILOG code of the block, which can be found also in Appendix E.

1. Terminals' names and functions

Reset	:This input terminal is used to initialize D type flip-flops in the block. System reset is active when signal is Logic 0.
clk	:This is an input terminal bringing system clock signal.
prg1_do...prg12_do	: These are input buses of size 8 bits. They are connected to the data_out bus A of respective dpram64x8.
uP_cs	:This is an input terminal. Microprocessor activates the Interface block with this signal. Block is activated when signal is Logic 0.
uP_write	: This is an input terminal. Microprocessor uses this terminal for writing the template-book. Block starts the writing operation when signal is Logic 0.
uP_read	: This is an input terminal. Microprocessor uses this terminal for reading the template-book. Block starts the reading operation when signal is Logic 0.
uP_dataHz	:This is an output terminal. Its value is Logic 1 during the writing operation and Logic 0 during the reading operation.
uP_adres	:This is an input bus containing 10 bits. It is connected to the address bus of microprocessor.
uP_dataout	:This is an output bus containing 8 bits. It is connected to the microprocessor data bus.

uP_datain	: This is an input bus containing 8 bits. It is connected to the microprocessor data bus too.
write_out	:This is an output terminal. It is connected to the WRA terminals of each dpram64x8. Thus, it controls the writing operations to the RAM blocks. Write_out is active when it is equal to Logic 1.
read_out	:This is an output terminal. It is connected to the RDA terminals of each dpram64x8 block. It is obtained by simply inverting write_out signal. Read_out is active when it is equal to Logic 1.
write_data	:This is an output bus containing 8 bits. It is connected to the DIA bus of each RAM block. Interface block sends template – vector data that will be written through this data bus. It is obtained by sampling uP_datain bus.
uP_adresd	:This is an output bus containing 6 bits. It is connected to the ADA bus of each RAM block. Interface block sends template –vector address through this address bus. It is obtained by sampling least significant 6 bits of uP_adres bus.
CSOUT	:This is an output bus containing 12 bits. Each bit of the bus is connected to the CSA terminal of respective RAM block. Thus, Interface block controls the activated RAM block through this bus.
ref_time	:This is an output bus containing 5 bits. It is connected to the Control block. It contains the information about the refresh time.

Least significant 6 bits of uP_adres bus are used to address directly RAM blocks (uP_adresd). Thus, to select the correct RAM block among 12 dpram64x8 or ref_time register, we use the most significant 4 bits of address bus. Address organization of the Interface block can be seen in Table 3.5.

Table 3.5 : Address organization of the Interface block

	RAM 1	RAM 2	RAM 3	RAM 4	RAM 5
uP_adres[adres9:6]	0000	0001	0010	0011	0100

	RAM 6	RAM 7	RAM 8	RAM 9	RAM 10
uP_adres[adres9:6]	0101	0110	0111	1000	1001

	RAM 11	RAM 12	Ref Time
uP_adres[adres9:6]	1010	1011	1100

Control Block uses ports B of RAM blocks. A write conflict may occur if the Control Block wants to read the template-vector that the Interface Block wants to change (write) at the same time. To avoid this conflict, I changed the active clock edge of the Interface block. As the Control block reads with the rising edge of the system clock, Interface block operates at the falling edge of the clock.

Microprocessor data bus is bi-directional and signal uP_dataHz controls the bi-directional data pads. If the signal uP_cs is Logic 0 (active) and uP_write is Logic 1 (non-active), Interface block executes reading process. This is preferred to decrease read access time. To avoid the mistakes due to asynchronous nature of the data that comes from microprocessor, the data used in the writing operation are obtained by sampling the address and data buses of microprocessor. Read and write cycles of the Interface Block can be seen in Figures 3.78, 3.79.

To change the refresh time of the system, we must follow the write procedure. After addressing the ref_time register with respect to the Table 3.5 and sending write control signal, Interface Block writes the least significant 5 bits of microprocessor data bus into the register.

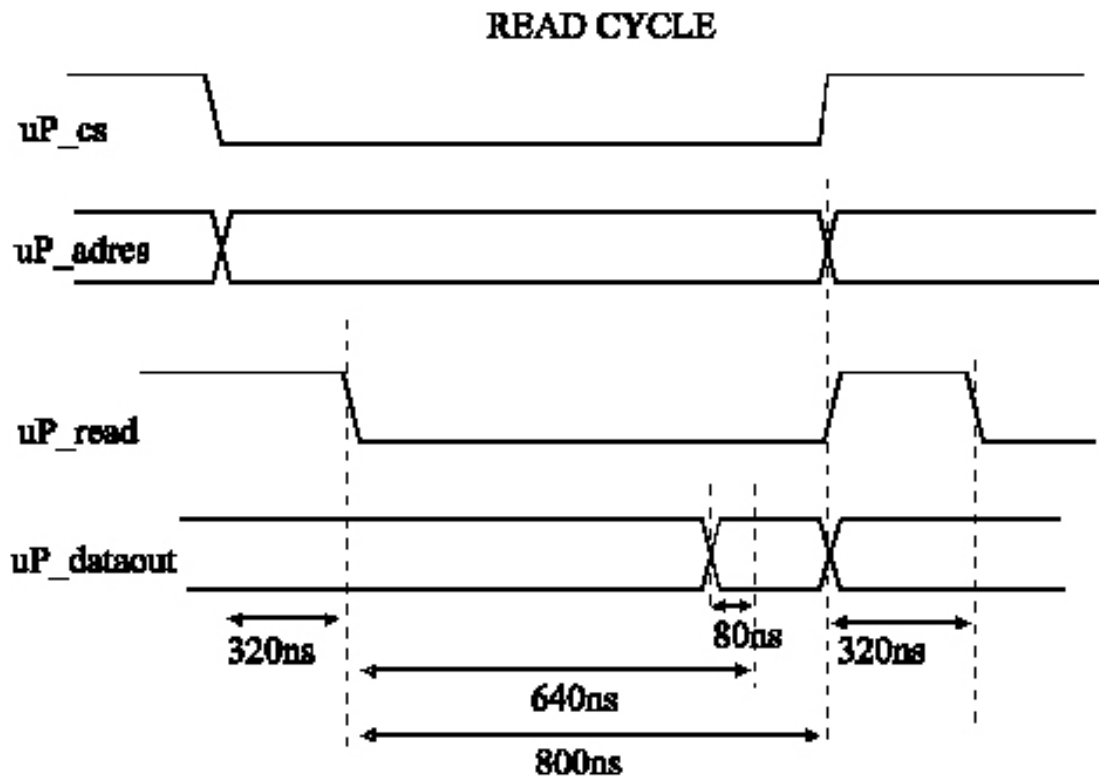


Figure 3.78 : Read cycle of the Interface block

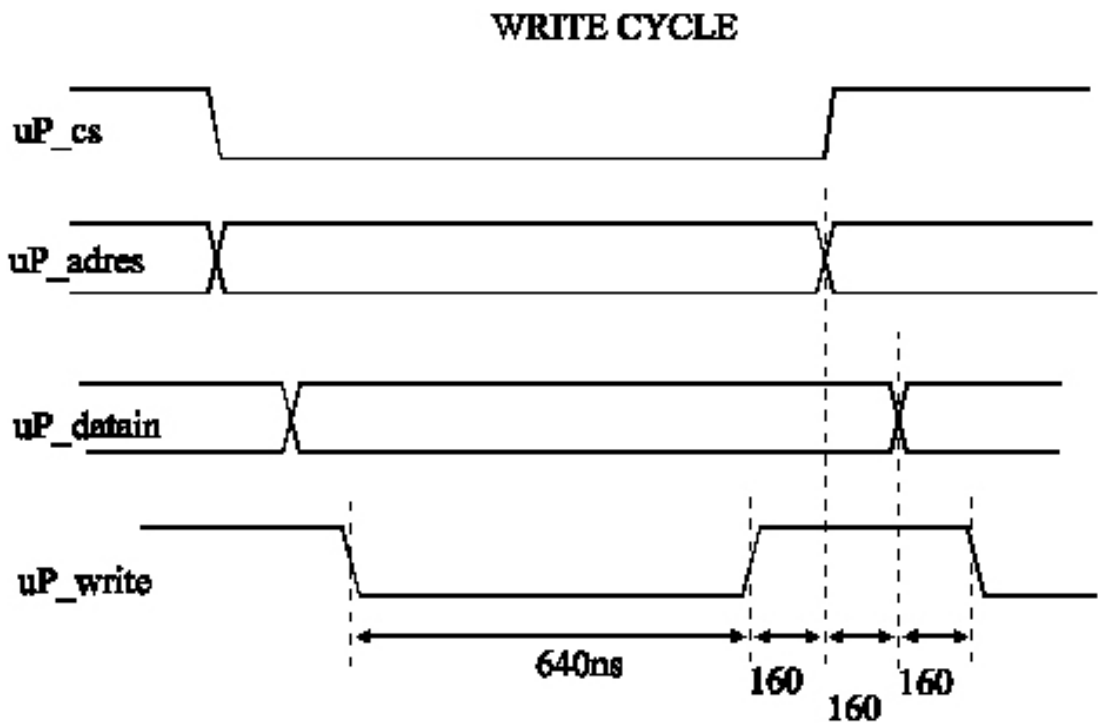


Figure 3.79 : Write cycle of the Interface block

In Figures 3.79, the proposed write cycle prevents data corruption during asynchronous write operation. Shorter cycle times are possible by simply increasing clock frequency of the Interface Block. Behavioral VERILOG code of the Interface Block is synthesized using SYNOPSIS®. Synthesized schematic of the block contains 124 standard cells of AMS 0.6µm standard cell library HRDLIB. The block Interface and the block Control consume a silicon area of 546.48e-3 mm² (828µm x 660µm). We can not show the schematic view of the Interface block due to its large standard cell count. Yet, VERILOG NETLIST of the block can be found in Appendix E. VERILOG Simulation results of Interface block can be seen in Figure 3.80.

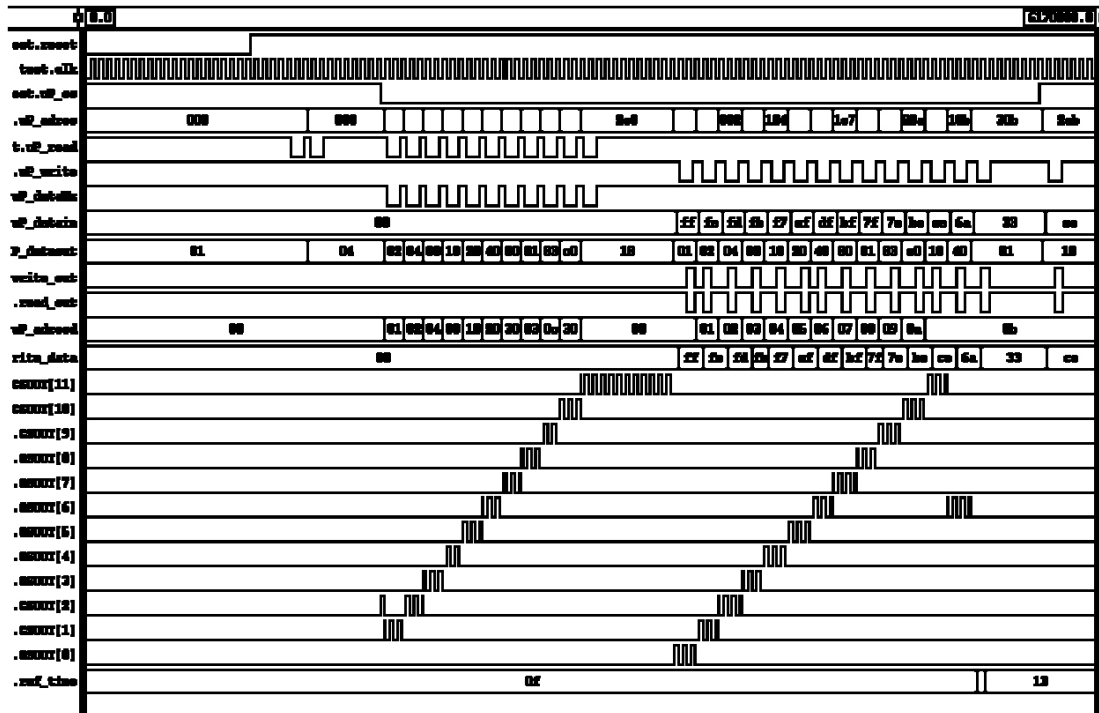


Figure 3.80 : VERILOG simulation result of Interface Block

With respect to the VERILOG simulation results in Figure 3.80, we can note the following:

1. If uP_cs is Logic 1, all bits of CSOUT bus are Logic 0. Thus, there is neither read operation nor write operation. (CSOUT)
2. If uP_cs is Logic 0 and uP_write is Logic 1, there is a read operation. (read_out and CSOUT)

3. During the write operation, first we change the address (thus addressed RAM block (CSOUT)), then we modify the data. Thus we can write the template-vectors' data without data corruption. (write_out, uP_adresd, write_data, CSOUT)
4. The data that will be written into the RAM block (write_data) is obtained by sampling uP_datain bus.
5. We can change easily the content of the ref_time register.
6. Read and write operations are asynchronous. (CSOUT)

3.6.3 Encoder Block

Third digital block named 'Output encoder' is responsible to encode WTA network output. Optimal encoding differs with respect to probability density function of the input vector source. Designed encoder block is a simple digital encoder whose output is the index of its input, which can be denoted by:

$$BO = \begin{cases} i & \text{if } BI = 2^i \\ BO & \text{otherwise} \end{cases} \quad 3.117$$

In (3.117), BO represents block output and BI represents block input. This is an optimal encoder if each template-vector has the same probability. The Behavioral VERILOG code of encoder block can be found in Appendix E. Block contains one input bus containing 64 bits and one output bus containing 6 bits. Block changes its output only when there is one and only one input bit that is Logic 0. And the block output varies with respect to the equation in (3.117). Behavioral VERILOG code of Output Encoder block is synthesized using SYNOPSYS®. Output Encoder block contains 146 standard cells of AMS 0.6µm standard cell library HRDLIB. The block Encoder consumes a silicon area of 105.21e-3mm² (167µm x 630µm). We can not show the synthesized schematic view of Encoder block due to its large cell count. Yet, VERILOG NETLIST of the block can be found in Appendix E

VERILOG Simulation result of Encoder Block can be seen in Figure 3.81.

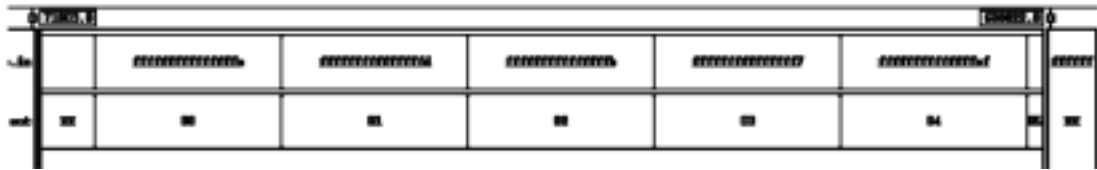


Figure 3.81 : VERILOG simulation result of Encoder Block

3.6.4 RAM block

In the VQ system, RAM block is responsible for digitally storing the template-book. We need this storage block for the refreshment need of the DCELL matrix. We have two digital blocks that will control the RAM block. The first one is the Control Block that wants to read the RAM block contents during the refresh period. And the second block is the Interface Block that wants to read (or write) the RAM block with respect to the microprocessor communicating with it. In order to meet our need, I used a Dual-Port RAM (DPRAM) block in the VQ system. I explained earlier the design methodology avoiding the write (or read) conflict while operating through different port of the RAM on the same address at the same time. The conflict may occur only if we want to write on a RAM's word from a port while we want to read (or write) the same word on the other port.

As mentioned previously, system storage requirement is 64x96 bits. In AMS macro block library, there is a dual-port ram (dpram64x8) of size 64x8 bits. Thus, I used 12 dpram64x8 blocks. Ports A of the DPRAM blocks are used by Interface block, whereas the Control block uses ports B. General information about the block can be found in Appendix E.

DPRAM blocks are connected as follows:

1. Signal CSA of each block is connected separately to the Interface block where they form CSOUT bus.
2. Signals CSB are connected to Ram_cs terminal of the Control block

Thus Interface Block may control each DPRAM block separately whereas Control Block control them together.

3. Signals RDA are connected to read_out terminal of the Interface Block.
4. Signals RDB are connected to logic level 1 because Control Block may only read the RAM blocks.
5. Signals WRA are connected to write_out terminal of the Interface Block.
6. Signals WRB are connected to logic level 0 because Control Block may never write to the RAM blocks.
7. Signals ENA and ENB are connected to logic level 0. Signals NRSTA and NRSTB are connected to logic level 1.
8. All of the address buses ADA are connected together to the bus uP_adresd of the Interface Block.
9. All of the address buses ADB are connected together to the bus Ram_adres of the Control Block.
10. Input data bus DIA of each DPRAM block is connected to the bus write_data of Interface Block.
11. Output data bus DOA of each DPRAM block is connected to the respective bus prog_dox of the Interface Block.
12. Output data bus DOB of each DPRAM block is connected to the input data bus of the respective digital to analog converter block.

The schematic view of connected DPRAM blocks can be seen in Figure 3.82.

