

**Military Technical College
Kobry El-Kobbah,
Cairo, Egypt**



**6th International Conference
on Electrical Engineering
ICEENG 2008**

FPGA implementation of the BIST intellectual property core for SRAM chips on the board

By

Mohamed H. El-Mahlawy* Mahmoud S. Hamed* M. H. Abd-El-Zeem* Isa Yossef *

Abstract:

During the last years, much time is spent on the development of Very Large Scale Integrated (VLSI) Circuits, which are circuits that contain millions of transistors within a small area. Most ICs are very complex, such that testing is very difficult, and much work must be done for developing a good test. Memory is considered an important element in the electronic system. Testing memories on the boards requires a special testing approach. Testing all these cells for high fault coverage is required. With the increasing complexity and density of memories, tests must be developed which require less application time and high fault coverage.

In this paper, an SRAM memory testing architecture using FPGA Spartan-3 System Board is presented. One of these tests, *Modified Algorithmic Test Sequence* (MATS), has been chosen to be implemented in our test architecture. This design is represented as an IP (Intellectual property) core that is able to perform the BIST (Built-In Self-Test) for memory on the board. This design is considered as a BIST testing tool for memory on the complex cards. This approach is considered to reduce the cost of the traditional ATE that consumes long time. So, our objective is to go step in the direction of the portable ATE that consumes less time. The hardware implementation occupies less hardware overhead. The FPGA implementation of this testable architecture demonstrates the efficiency of the testing approach.

Keywords:

Design for Test, and Memory Testing on electronic cards

* Egyptian Armed Forces

1. Introduction:

Nowadays, the application of integrated circuits in electronic systems has increased considerably. Much time is spent on the development of Very Large Scale Integrated (VLSI) Circuits, which are circuits that contain millions of transistors within a small area. They are manufactured in complicated processes, and consequently defects in the circuits are possible. Because it is important to know whether a manufactured chip is defective or not, it has to be tested to verify the correct working. Most ICs are very complex, such that testing is very difficult, and much work must be done for developing a good test. The researcher and the manufacturer are interested in where the fault appears in the circuit, and they need a test which localizes the fault.

Integrated circuits may be classified into two classes: combinational circuits and sequential circuits. The output of a combinational circuit depends only on the input signals, so a defect in it can be detected by a specific combination of input signal values. On the other hand, in sequential circuits, the output depends on the input signals and the internal state of the circuit and thus it contains a storage memory element, and detection of a defect is more complicated because it depends on both the input signals and the state of the storage memory element of the circuit. For the detection of a given defect, the circuit must be forced in a given state, and then a specific combination of input signals must be applied to the circuit. According to the 2001 International Technology Roadmap for Semiconductor (ITRS 2001), today's system on chip (SOC) are moving from logic dominant chips to memory dominant chips, since future applications requiring a lot of memory. The memory share on the chip is expected to be about 94% in 2014 [1].

Memory Testing requires a specific approach, because the memory can be in 2^n states, where n is the number of bits in the memory. Testing all these states is impossible, and much work has been done to develop a good test method for memories. With the increasing complexity and density of memories, tests must be developed which require less test time per bit and achieve a better fault coverage.

The traditional memory tests presented in [2] have time complexity $O(n^2)$, which made them uneconomical for larger memories. The *functional fault models* have been developed, which give an abstract functional description of the faults. Tests have been developed, based on these fault models, and a large group of these tests is known as a *March tests* [2]. The advantages of march tests lay in two facts. First, the fault coverage of the considered known models could be mathematically proven, although one could not have any idea about the correlation between the models and the defects in the real chips. Second, the test time for march tests were usually linear with the size of the memory, which make them acceptable from industrial point of view. In this paper, the march test called MATS [3-4] will be selected as an example for application of memory BIST architecture using Spartan-3 FPGA board. The Automatic Testing technique to

test the memory on board will be discussed. The authors in paper [4] have presented the FPGA implementation of the new automatic test equipment (ATE) for digital integrated circuits (combinational and sequential circuits) based on the signature analysis. In this paper, the FPGA implementation of the SRAM memory BIST architecture is presented with the selected March algorithm (MATS). The timing simulation and then design download are presented on the Spartan X3C200 Xilinx chip. The concept of the System-on-Chip (SoC) is presented that reduces the complexity of the traditional ATE. This testing architecture is designed to apply the test pattern to the memory under test (address bus, data bus, and control signals) and to compact its response by using multi-input signature analyzer (MISR). The timing controller generate all control signals to control all steps of the memory test cycle for proper operation. The signature comparison is achieved to detect the faulty memory.

This paper is organized as follows; section 2 talks, briefly, about the idea of this testing architecture. The implementation of this memory BIST architecture is presented in section 3. Finally, the experimental results and the conclusions will be discussed in section 4 and section 5, respectively.

2. Concept of the memory BIST architecture

Advances in the FPGA technology have led to the fabrication of chips that contain a very large number of logic gates, integrated on a single chip. Therefore, the design of the complete system on single chip is interesting to make the portable systems. The FPGA is a regular structure of logic cells (or modules) and interconnect. The integration of 74 series standard logic into a low-cost FPGA is a very attractive proposition. It enables to save printed circuit board (PCB) area and board layers thus reducing your total system cost. It is a superior alternative to mask-programmed Application Specific Integrated Circuits (ASICs). Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary. The FPGA also gives users high performance. Now, you can design, program, and make changes to your system whenever you wish. The Spartan-3 FPGA board enables a complete system functions in a low cost and space efficient manner [6].

The Digilent Spartan-3 development board, illustrated in Figure 1, is the main hardware component used in this project. This tool provides a powerful, self-contained development platform for designs targeting the new Spartan-3 FPGA from Xilinx company. The board has many components that allow the user to develop and evaluate a system centered on the Spartan-3 FPGA such as:

- A 200 K-gate Spartan-3 FPGA in a thin ball grid array package.
- A 2 Mbit PROM.
- Two independent 256Kx16 SRAM arrays (IS61LV25616AL 256K × 16).

- 4 character 7-segment display, 8 switches and 4 buttons.
- 50 MHz oscillator clock source.
- JTAG port.

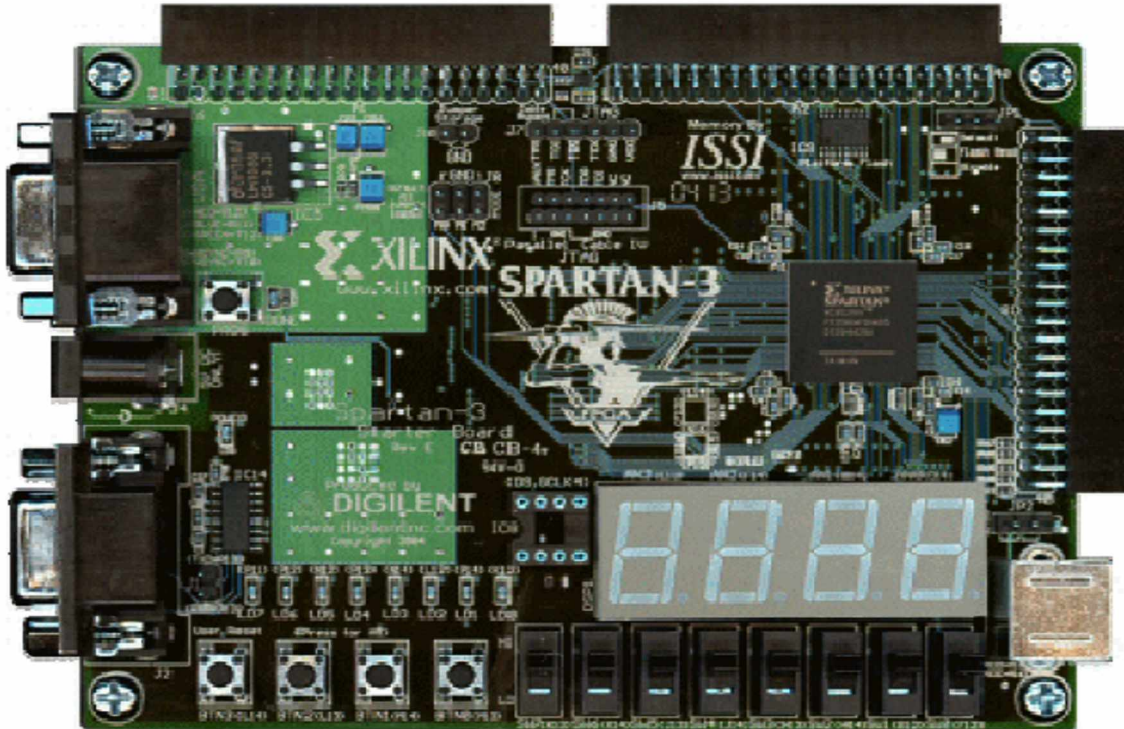
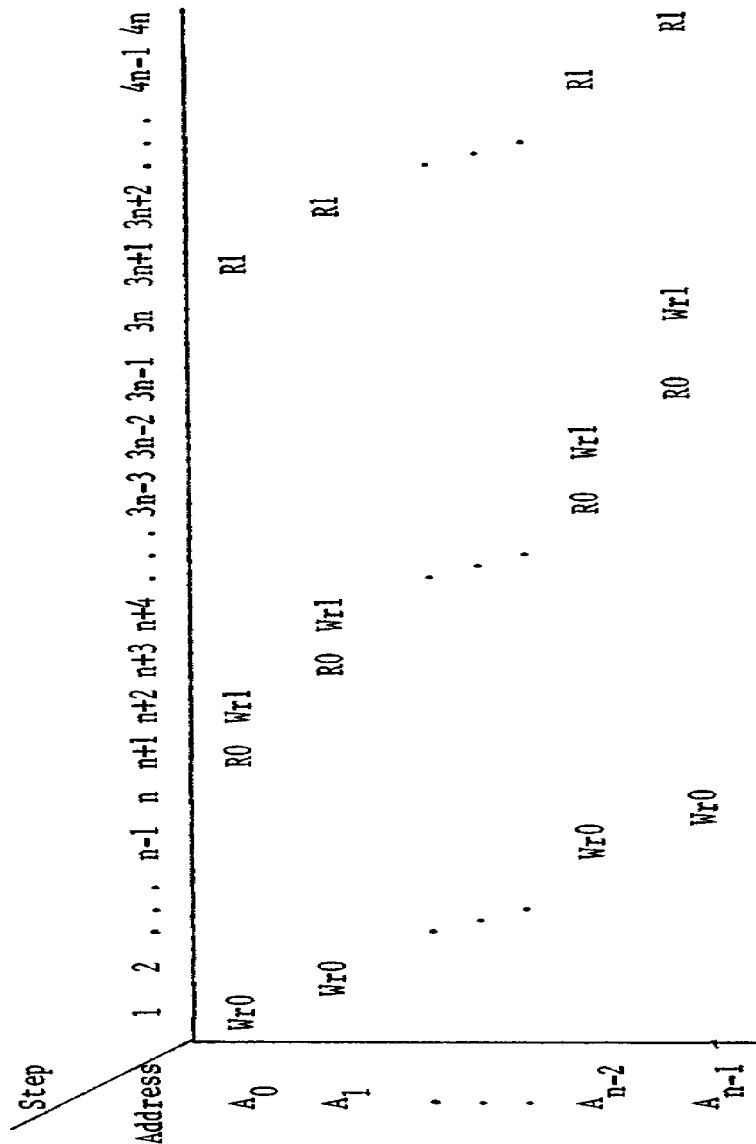


Figure 1: Digilent Spartan-3 Development Board.

The march test can be defined as a sequence of march elements, where a march element is a sequence of memory operations performed sequentially on all memory cells. In a march element, the way from one cell to the next is specified by the address order, which can be increasing or decreasing. For some march elements, the address order can be chosen arbitrarily as increasing or decreasing. In a march element, it is possible to perform a write 0 operation (W0), write 1 (W1), read 0 (R0) and read 1 (R1) operation. The 0 and 1 after read operations represent the expected values of the read on the output. An example of a march element is \uparrow (R0; W1), where all memory cells are accessed in an increasing address order while performing R0 then W1 on each cell, before continuing to the next cell. By arranging, a number of march elements one after the other, a march test is constructed. Because of their simplicity and linearity with the memory size, all of them are in $O(n)$.

MATS tests detect any combination of stuck-at faults in a RAM, irrespective of the design of the decoder combinational circuit [6]. MATS tests rearrange the test sequence as shown in Figure 2. It was proved that MATS covers all stuck-at faults (SAF) in RAMs, independent of the decoder design [6].



test patterns for the address bus of the memory under test (MUT). The *memory_io* cell is used to handle the data to or from the data bus of the MUT. The *misr* cell is designed to compact the stream test pattern response from the data bus of the MUT into the corresponding compacted signature that displays on four-digit seven-segment display, through the *signature_display* cell.

Before the explanation of these cells in details, we need to focus on the functionality of the *Programmable Timing Generator* cell. The schematic diagram and the timing diagram of this cell are illustrated in Figure 4 and Figure 5, respectively. It produces eight signals: "Clock_DIV" signal refer to CLK_1M in Figure 3, "Shift_CLK_DIV" signal, "Neg_Shift_CLK_DIV" signal, "CARRY1", "CARRY2", "CARRY4", and "CARRY5" signals, and N_CLK_RW signal. This cell divides the clock CLK of the board (50 MHz) into another clock CLK_1M which used to drive the *signature-display* cell and the *timing_control* cell simultaneously. The main clock CLK is fed to a programmable divider which divides it by 3, and produces CARRY1 signal as shown in Figure 5. CARRY1 is fed to another programmable divider which divides it by 3 to produce CARRY2 signal. CARRY2 is used to drive a third programmable divider, which it divider by 4, to produce CARRY4 and CARRY5 signals using a 4-bit serial-in/parallel-out shift register. Two D-type Flip-Flops are used, one of them is used to divide CARRY2 signal by 2 and give the output clock CLK_DIV (CLK_1M), and the other is used to produce Shif_CLK_DIV, which is fed to *timing_control* cell, and Neg_Shif_CLK_DIV signals as shown in Figure 5. A third D-type Flip-Flop is used to divide CLK_DIV (CLK_1M) signal by 2 and produce CLK_RW and N_CLK_RW signals. Figure 5 illustrates all signals generated from this cell after the timing simulation.

3.1 Explanation of the design of the timing control cell

The *timing_control* cell is responsible for the timing synchronization of the proper testing operation. The schematic diagram of this module is illustrated in Figure 6. This cell has four input signals. CLK signal in this cell is fed from clock CLK_1M as shown in Figure 3 of the main cell. START signal used to start the test process. CLEAR signal used to initialize this module. Shif_CLK signal is the clock CLK_1M shifted by the duration of the CARRY2. It is used to control the write (WR) and read (OE) operations of the MUT.

This cell is composed of three parts. The first part is the *TESTING GATE CLOCK GENERATION*. This part is responsible for generating the testing gate clock (CLOCK1 and CLOCK1_sig). CLOCK1 is the gated clock inside the testing gate interval (CLOCK 3OS). It is composed of the five 4-bit binary counters, as shown in Figure 6. The assertion of the five 4-bit binary counters at the rising edge of the clock. CLOCK1 feeds the *mtpg* cell as will be shown in section 3.2. CLOCK1_sig feeds the *misr* cell as will be shown in section 3.5.

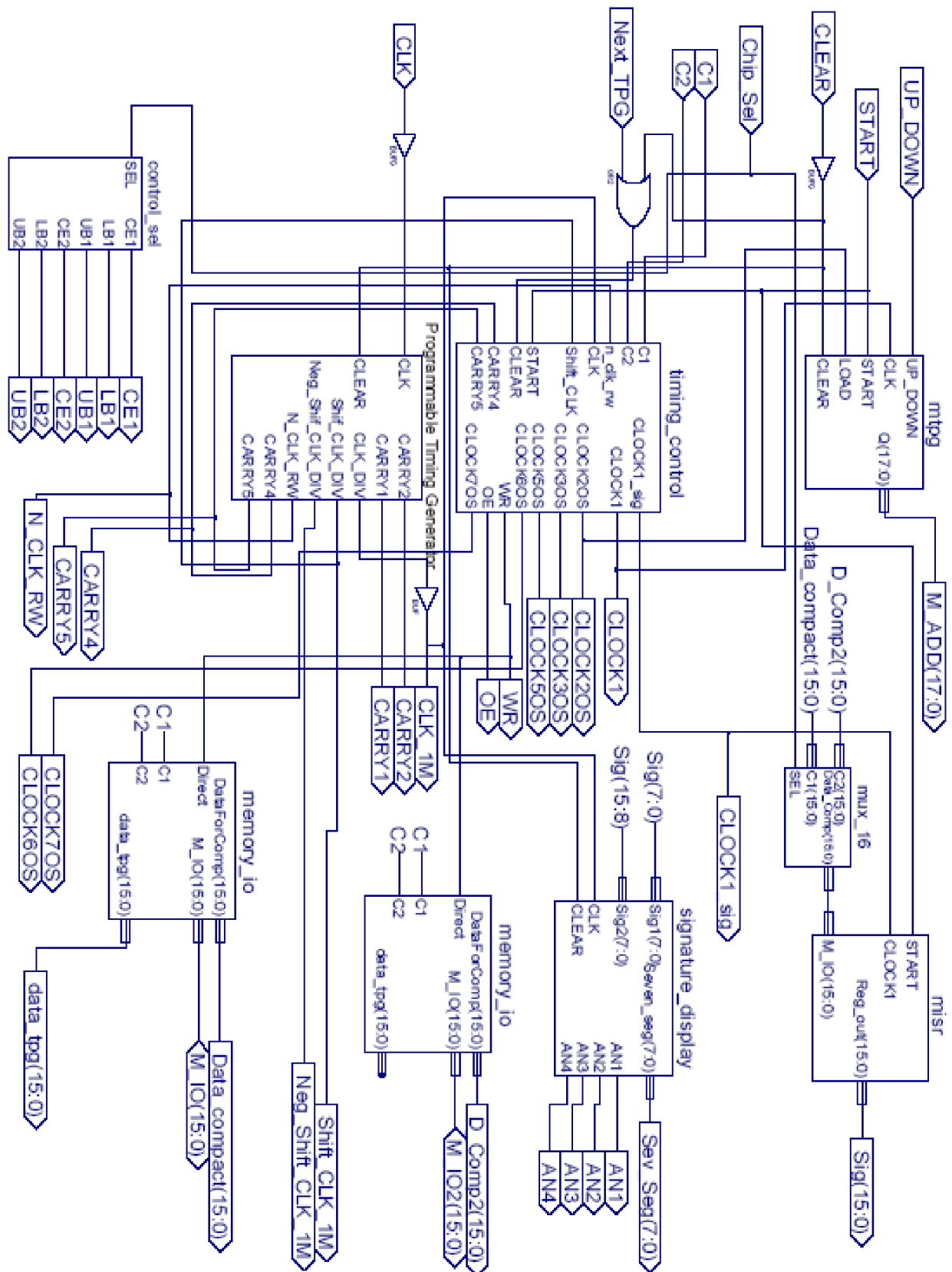


Figure 3: Block diagram of the FPGA implementation of the memory BIST architecture.

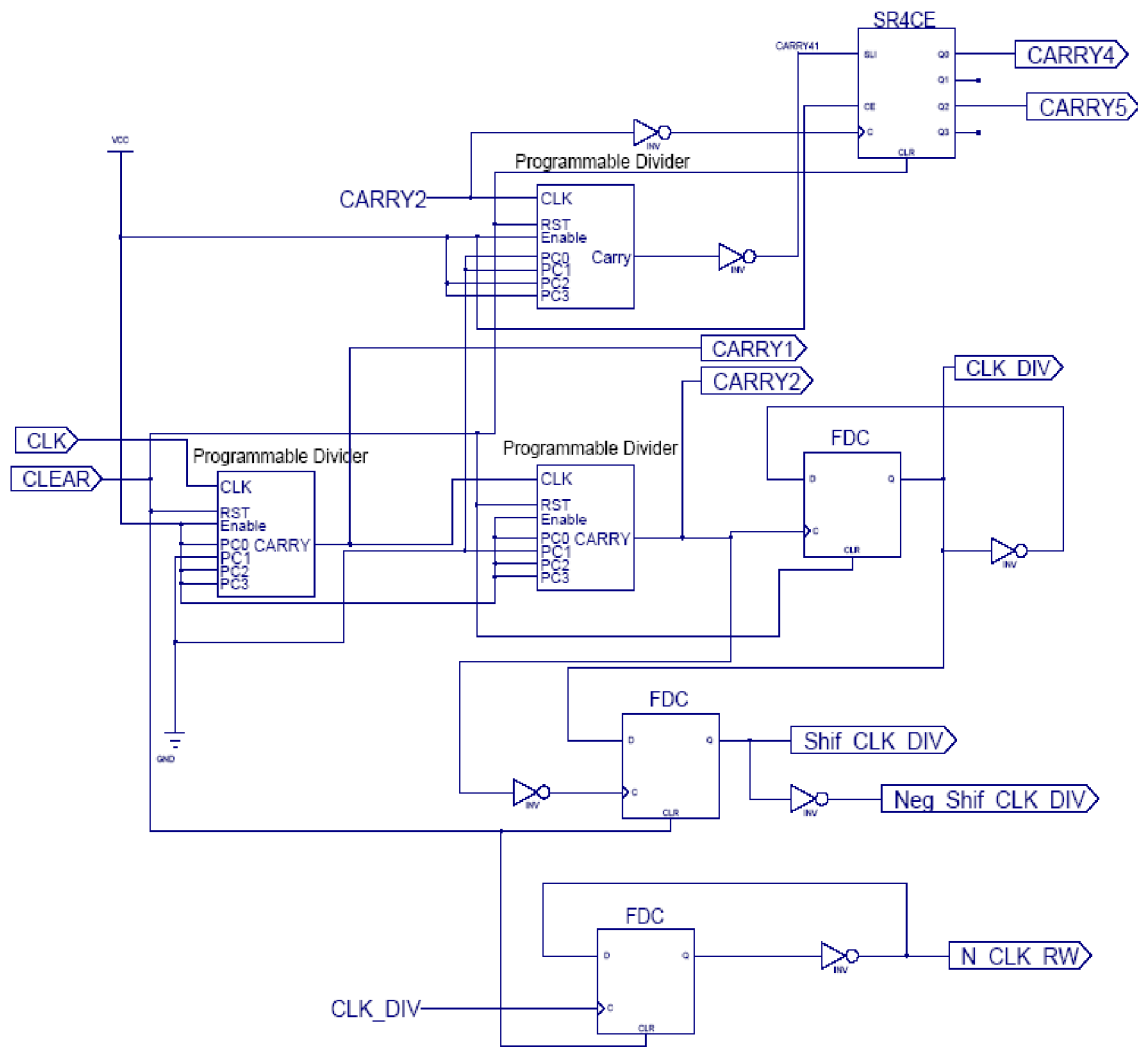


Figure 4: Schematic diagram of the programmable_timing cell.

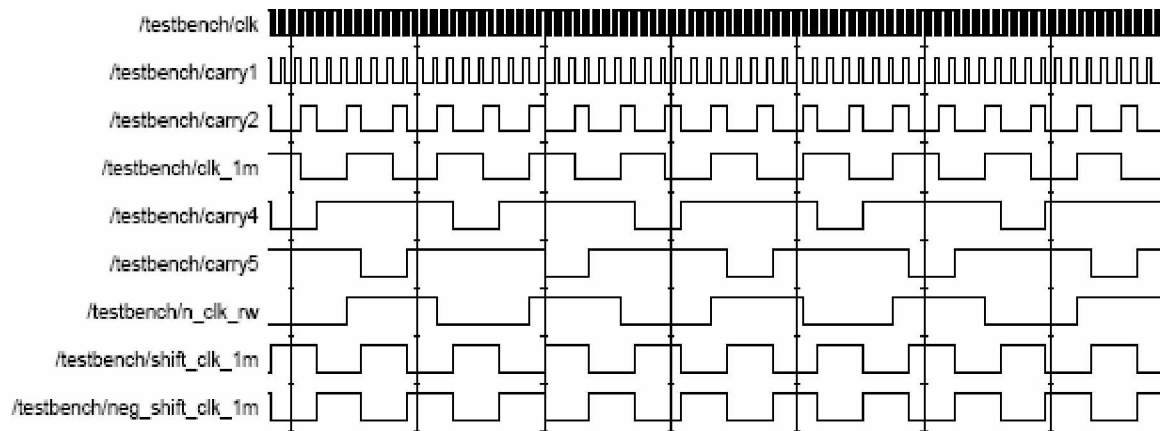


Figure 5: Timing diagram of the Programmable Timing Generator cell.

The second part is the *TESTING SIGNAL SYNCHRONIZER*. This part is responsible for generating the output control signals; CLOCK2OS, CLOCK3OS, and CLOCK5OS. It is composed of a 4-bit BCD counter, seven Flip-Flops and other random logic gates as shown in Figure 6. CLOCK2OS signal is used to clear the *mtpg* cell as will be shown in section 3.2 at the beginning of each testing gate. CLOCK3OS signal is the testing gate interval and CLOCK5OS signal is the half clock shift version of CLOCK3OS as shown in Figure 7 to Figure 12.

The third part is responsible for generating the memory control signals (WR, and OE). It is composed of two 4-to-1 multiplexers and other random logic gates as shown in Figure 6. The control signals C1 and C2 are used to control the selection of the two multiplexers according to table 1. They are used to generate OE and WR signals as shown in Figure 7 to Figure 12.

Table 1

C1	C2	WR	OE	Mode
0	0	Shift_CLK	1	Write mode
1	0	1	Shift_CLK	Read mode
0	1	CARRY4	CARRY5	Read-Write mode

3.2 Explanation of the design of the *mtpg* cell

This cell is responsible for accessing the memory address bus. It is composed of five 4-bit binary counters, which are connected in cascaded manner, and other random logic gates as shown in Figure 13. It has five inputs. CLK signal in this cell is fed from clock CLOCK1 of *timing_control* cell as shown in Figure 3. START signal is used to start the test process. CLEAR signal is used to initialize the module. LOAD signal in this cell is fed from signal CLOCK2OS of *timing_control* cell. UP_DOWN signal is used to select the way of the accessing process (increment or decrement). The outputs of this counter Q(17:0) are connected to the memory address bus connections in Spartan-3 FPGA card as shown in table 2.

Table 2 External SRAM address bus connections to Spartan-3 FPGA

Address bit	FPGA pin	Address bit	FPGA pin	Address bit	FPGA pin
Q(17)	L3	Q(9)	E4	Q(1)	N3
Q(16)	K5	Q(8)	E3	Q(0)	L5
Q(15)	K3	Q(7)	F4		
Q(14)	J3	Q(6)	F3		
Q(13)	J4	Q(5)	G4		
Q(12)	H4	Q(4)	L4		
Q(11)	H3	Q(3)	M3		
Q(10)	G5	Q(2)	M4		

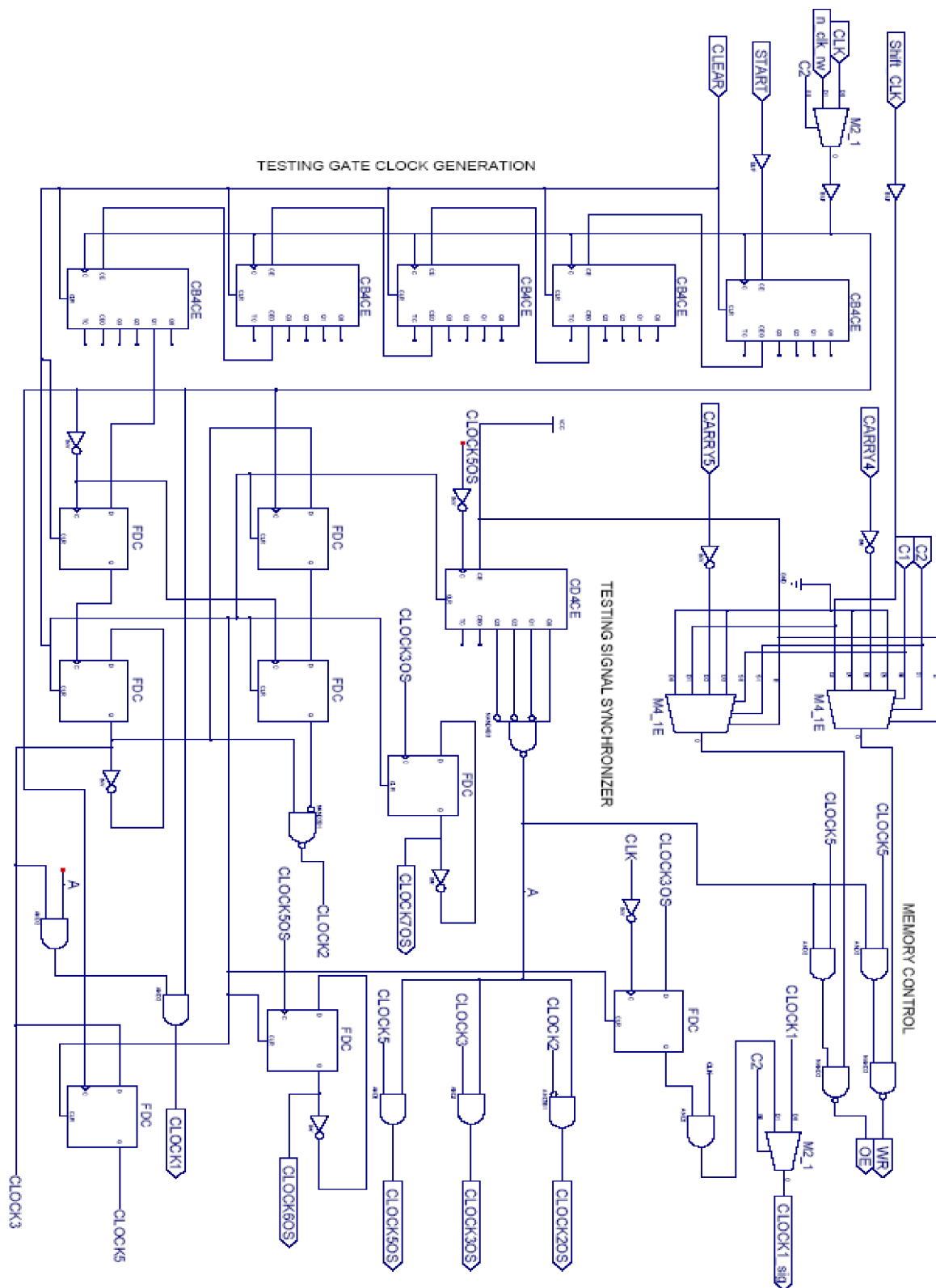


Figure 6: Schematic diagram of the timing_control cell.

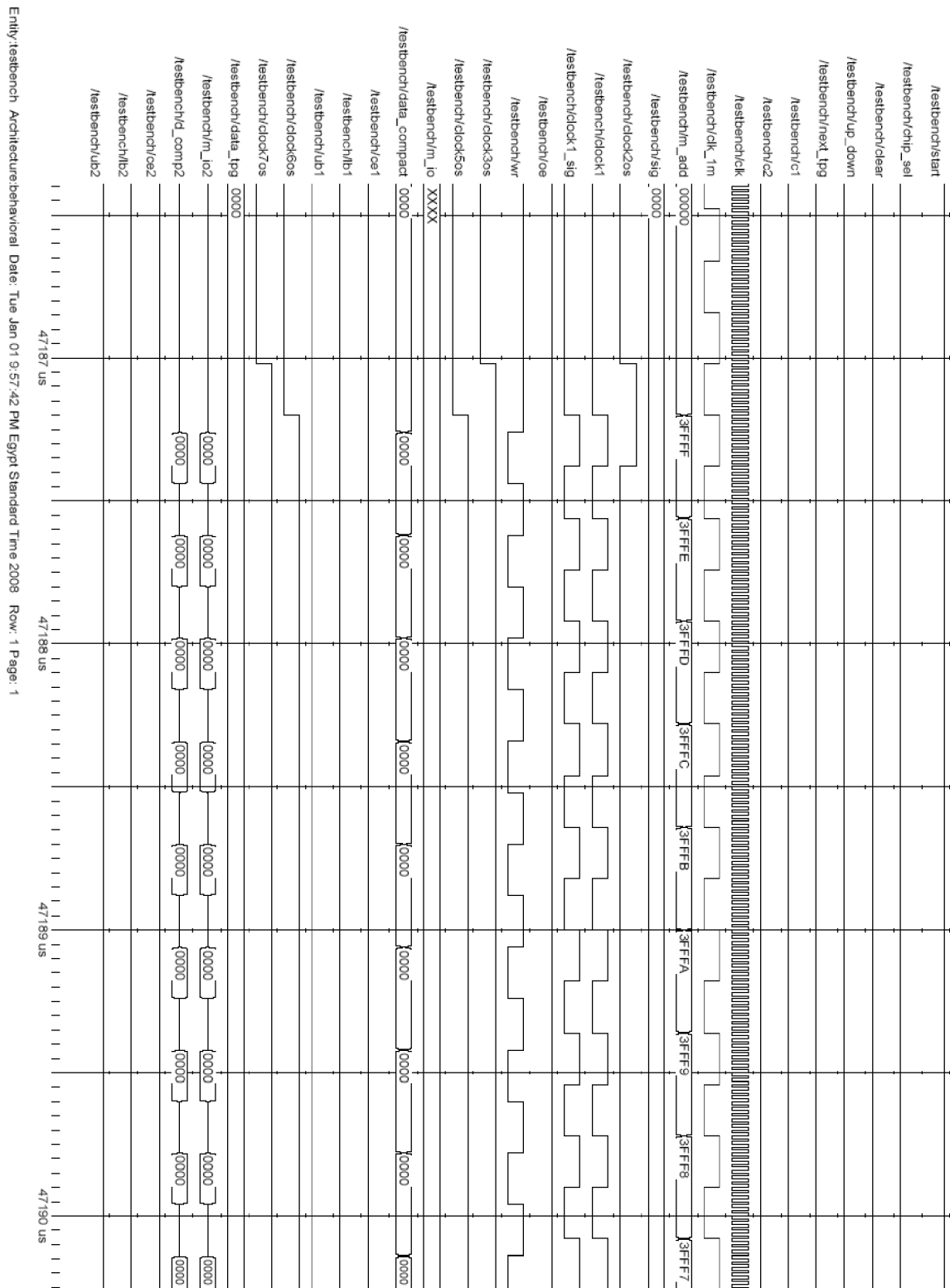


Figure 7: Timing diagram of the starting testing gate in the write operation.

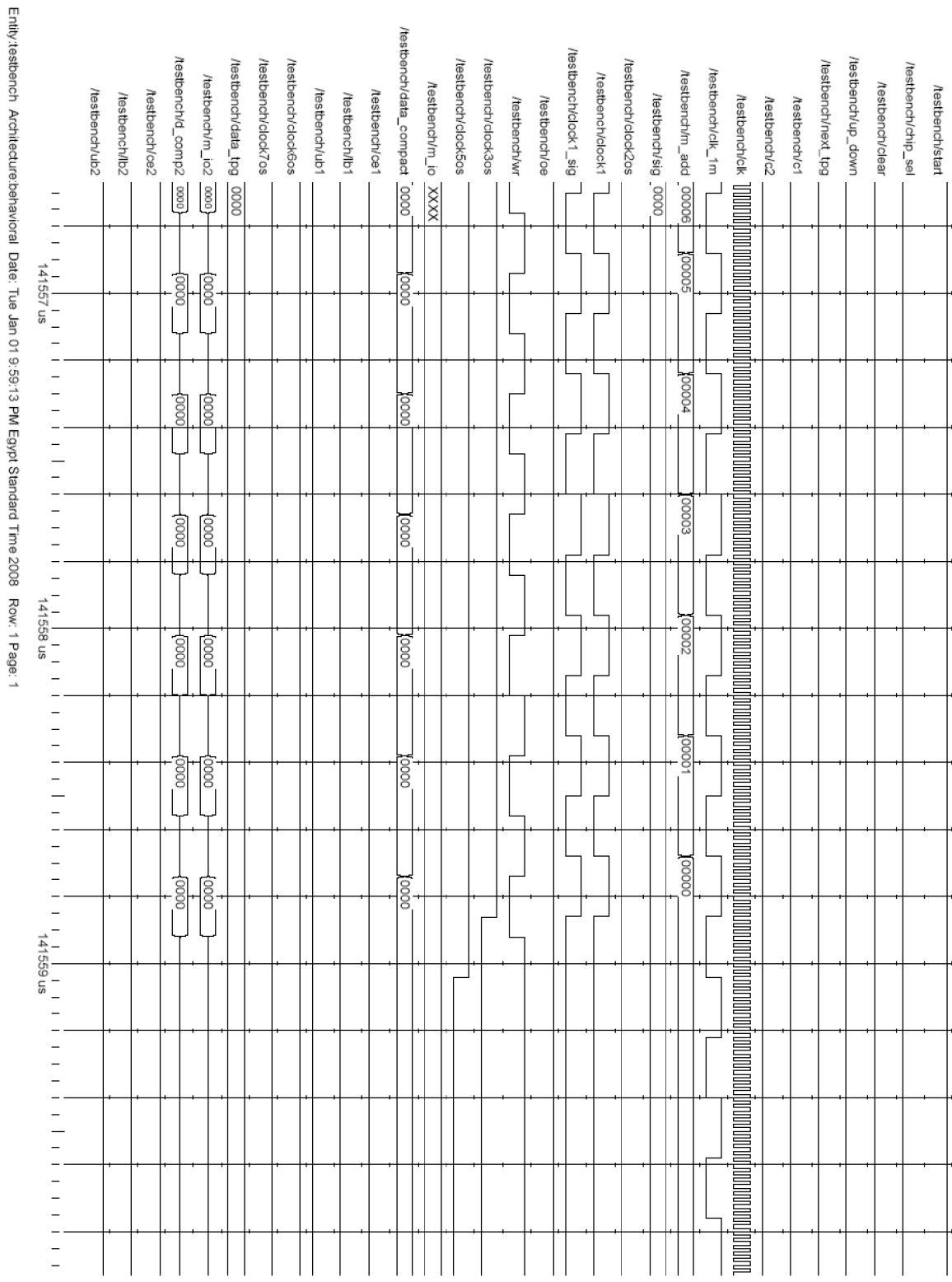


Figure 8: Timing diagram of the ending testing gate in the write operation.

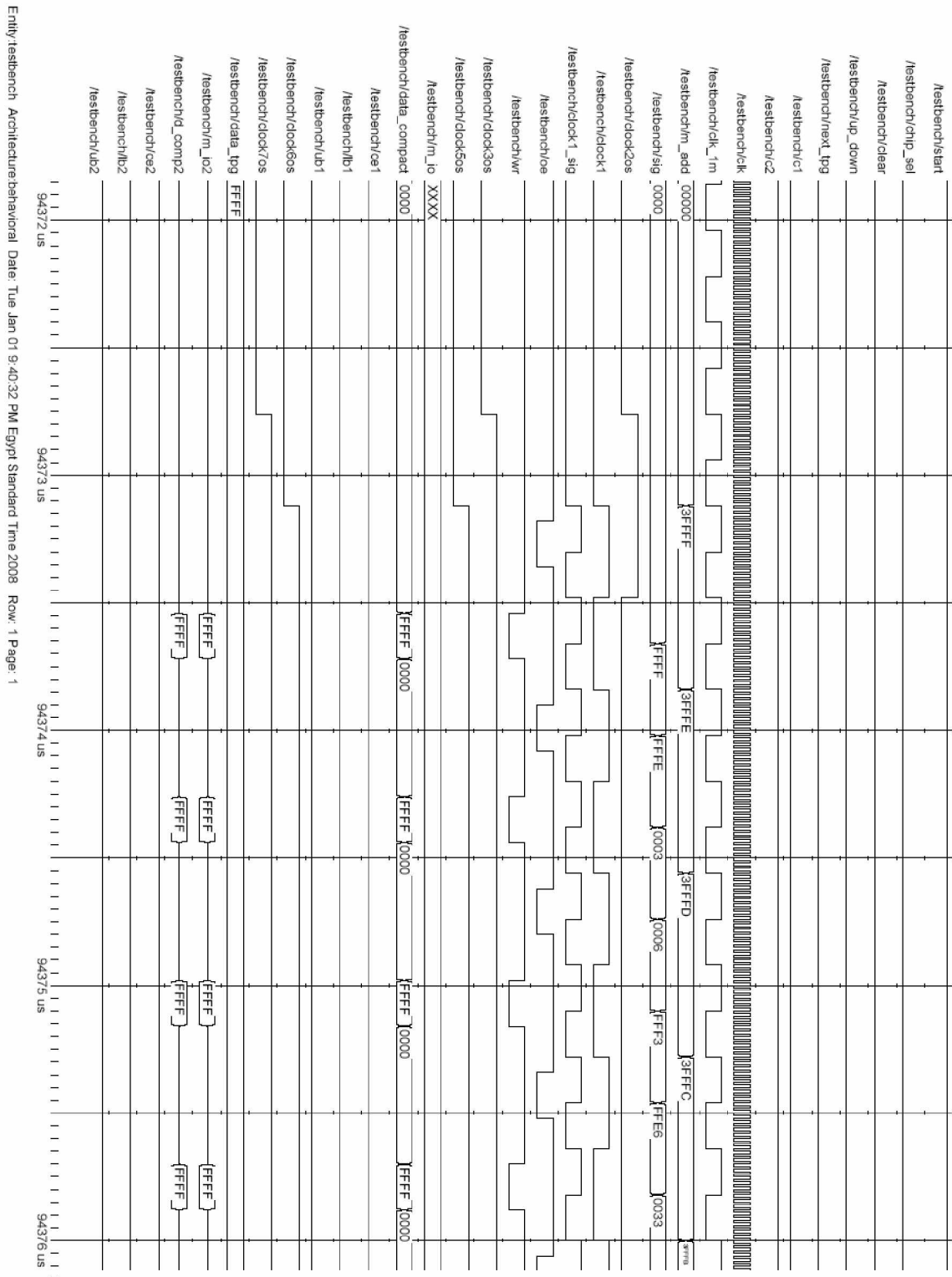


Figure 9: Timing diagram of the starting testing gate in the read & Write operation.

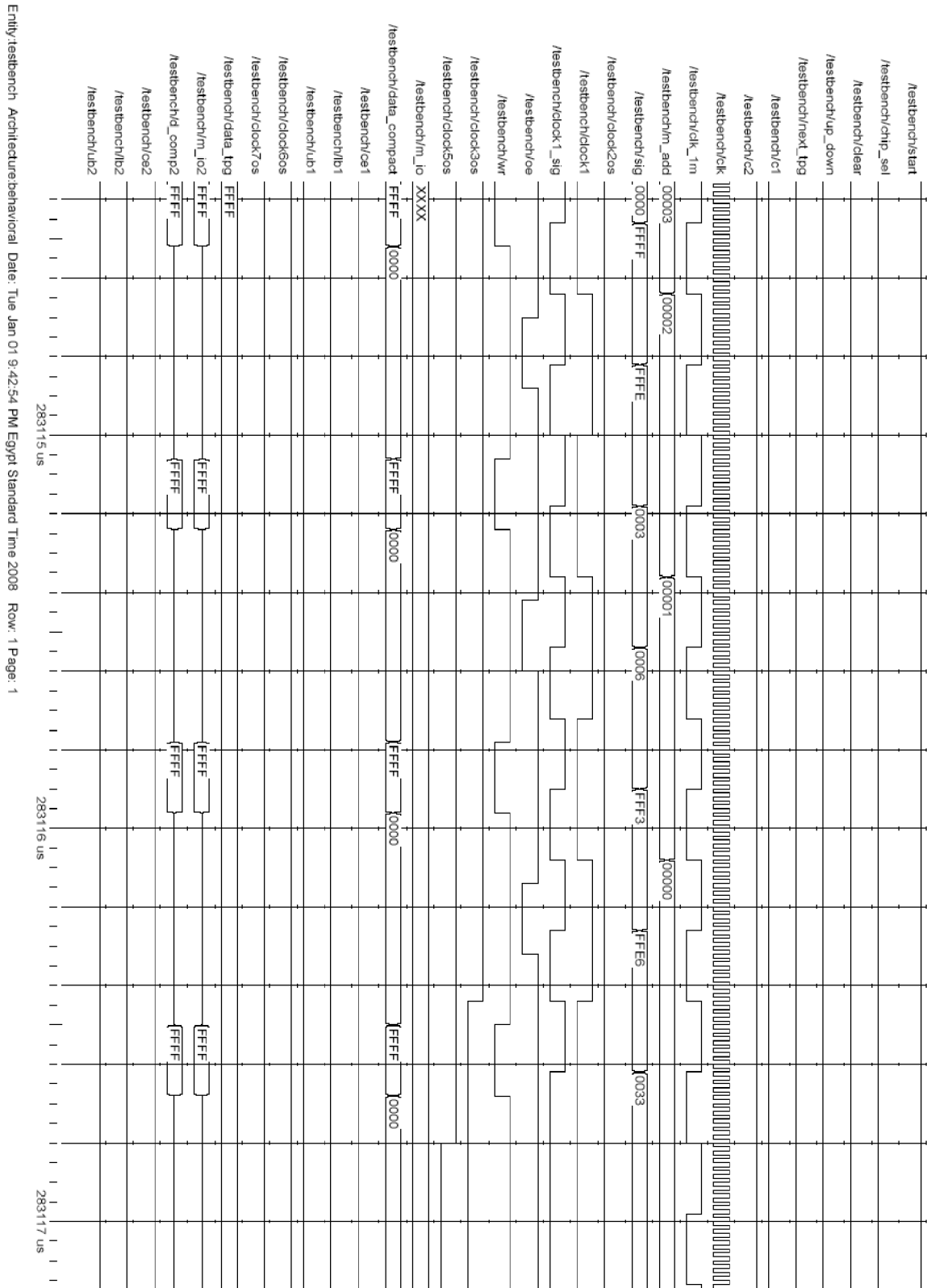


Figure 10: Timing diagram of the ending testing gate in the read & write operation.

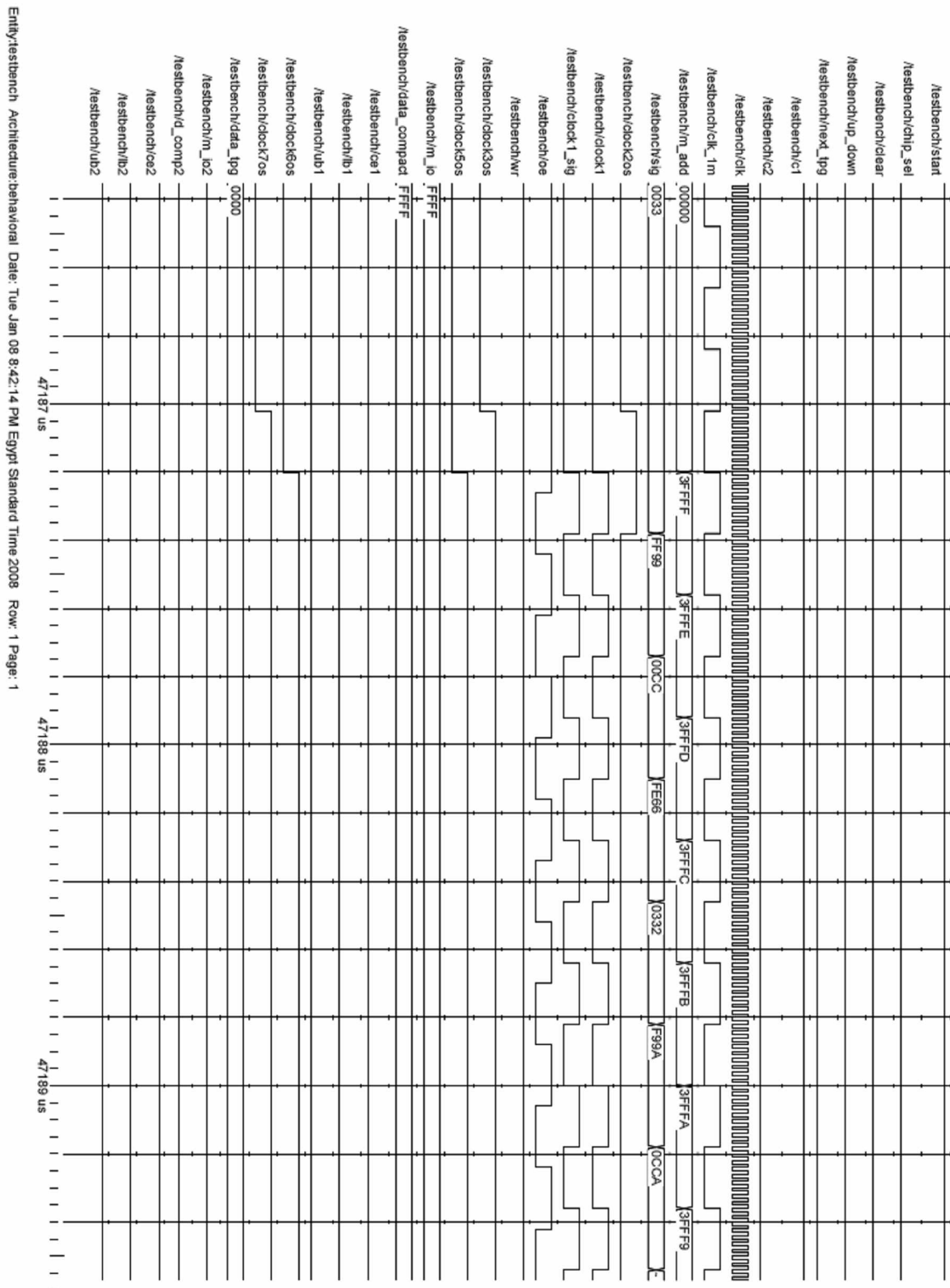


Figure 11: Timing diagram of the starting testing gate in the read operation.

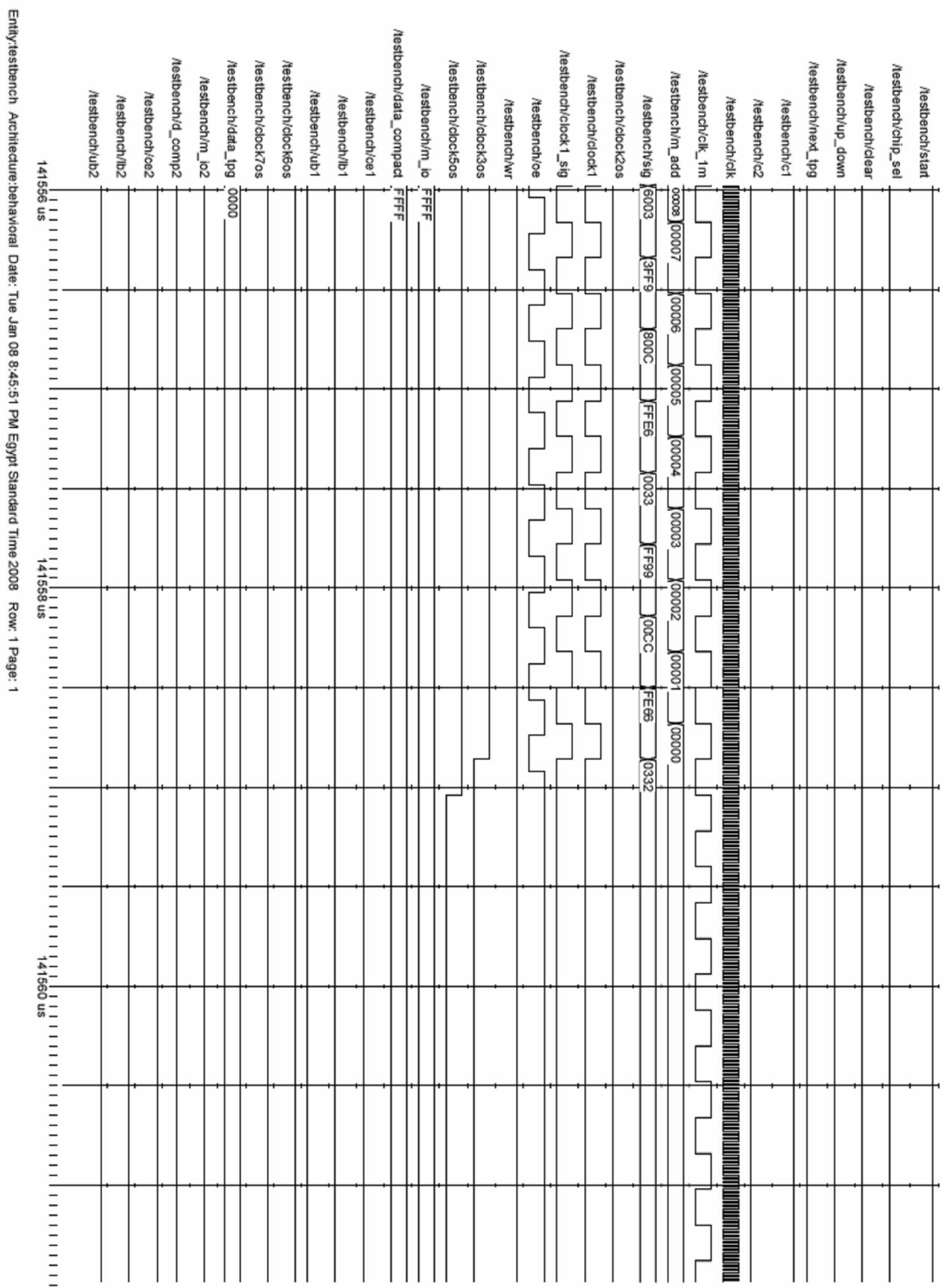


Figure 12: Timing diagram of the ending testing gate in the read operation.

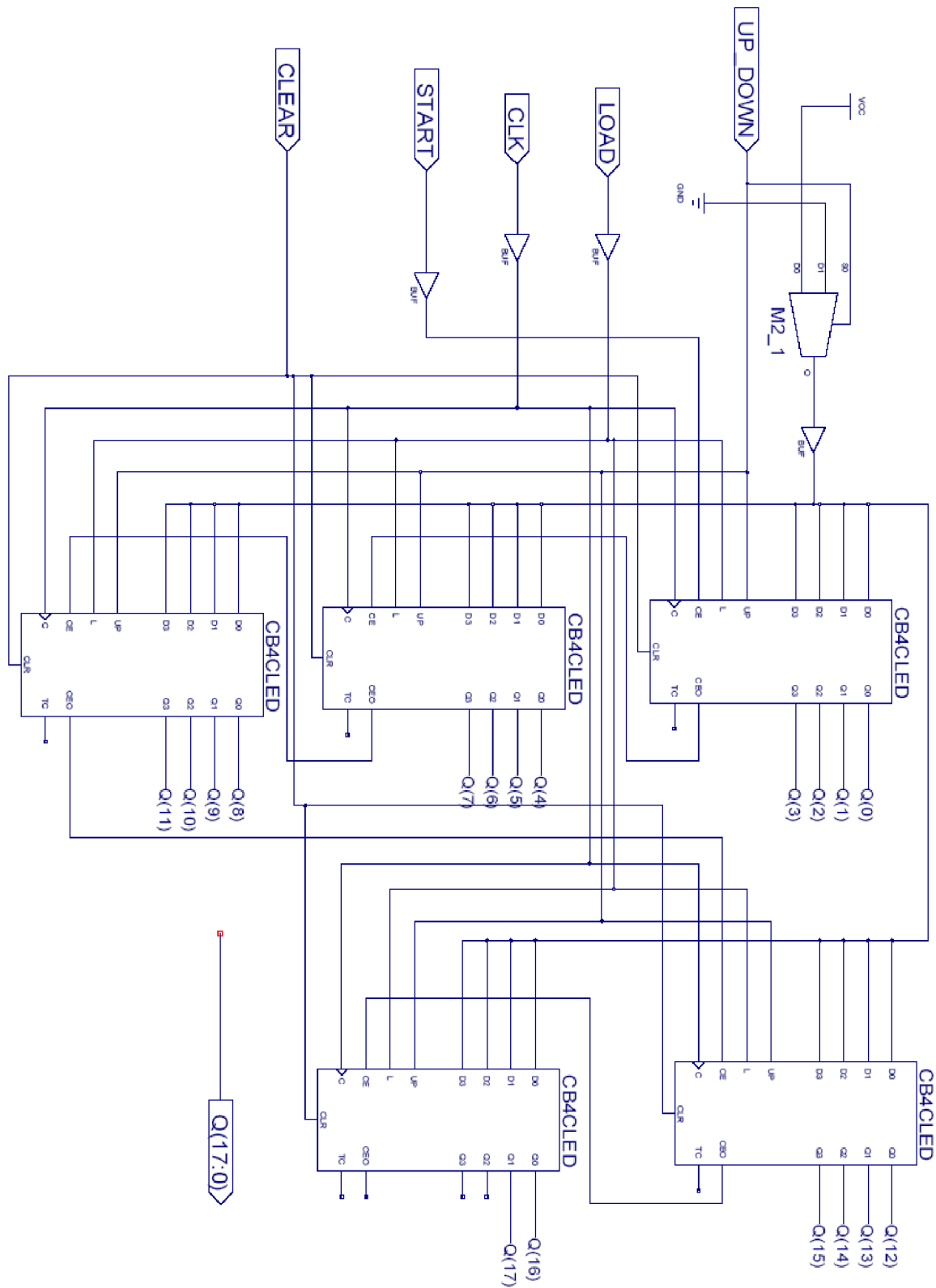


Figure 13: Schematic diagram of the mtpg cell.

3.3 Explanation of the design of the control sel cell

Because of Spartan-3 board has two SRAM chips; we used this cell to select on which one of these memories the test will be applied. Both SRAM devices share common write-enable (WE), output-enable (OE), and address bus signals Q(17:0). However, each device has a separate chip select enable (CE), and byte selection control UB, and LB to select the high or low byte in the 16-bit data bus. This cell consists of six 2-to-1 multiplexers and gives two groups of outputs, each group consist of three outputs: (CE1, LB1, UB1) or (CE2, LB2, UB2). These outputs are connected to the memories through the FPGA pin connections as indicated in table 3. It has one input signal, SEL, to select memory under test. So, if SEL is High, the first memory device is chosen. While, if SEL is Low, the second memory device is chosen. Figure 14 illustrates its schematic diagram.

Table 3 External SRAM control connections to Spartan-3 FPGA

Control signal	FPGA pin
CE1	P7
LB1	T4
UB1	P6
CE2	N5
LB2	R4
UB2	P5

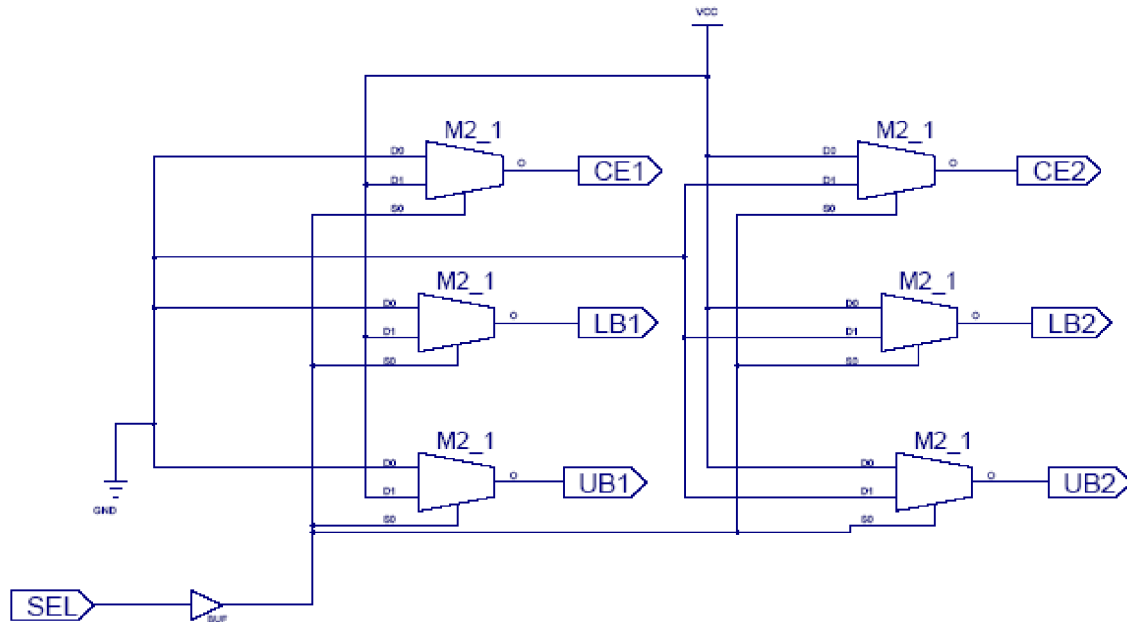


Figure 14: Schematic diagram of control_sel cell.

3.4 Explanation of the design of the memory_io cell

The main purpose of this cell is to control the data flow from/to the data bus of the memory in the write/read memory operation, respectively. In the memory read operation, the data from the memory is acquired and applied to the *misr* cell (see section 3.6) through *mux_16* cell (see section 3.5). In the write memory operation, the specific test pattern to the memory is applied. The signal *Direct* in this cell controls the data flow from/to the memory. The data bus of the memory is fed through 16-bi-directional input/output buffers. If the signal *Direct* is low, the *data_tpg(15:0)* signals are written to the MUT (write operation) and are directed also to the *misr* cell through the *DataForComp(15:0)* signals. If the signal *Direct* is high, the *data_tpg(15:0)* signals will be floated and the data from the MUT will be read (read operation) and directed to the *misr* cell through the *DataForComp(15:0)* signals. Figure 15 illustrates the schematic diagram of *memory_io* cell, while table 4 shows the truth table of the bi-directional buffer.

Table 4 shows the truth table of the bi-directional buffer

Inputs		Bidirectional	Outputs
T	I	IO	O
1	X	Z	X
0	1	1	1
0	0	0	0

The *data_tpg(15:0)* signals are controlled by signals C1 and C2 (see table (5.1)). The following VHDL code of *Det_TPG* cell is:

```

entity Det_TPG is
    Port ( C : in std_logic_vector(1 downto 0);
          Data_tpgMATS : out std_logic_vector(15 downto 0));
end Det_TPG;

architecture Behavioral of Det_TPG is
begin
    process (C)
    begin
        case c is
            when "00" => Data_tpgMATS <= "0000000000000000";
            when "01" => Data_tpgMATS <= "0000000000000000";
            when "10" => Data_tpgMATS <= "1111111111111111";
            when others => NULL;
        end case;
    end process;
end Det_TPG;

```



3.5 Explanation of the design of the misr cell

The main purpose of this cell is to compact the data outputs from *memory_io* cell and then send it to the *signature_display* cell (see in section 3.6). The general block diagram of the *MISR* as compacted test response is illustrated in Figure 16.

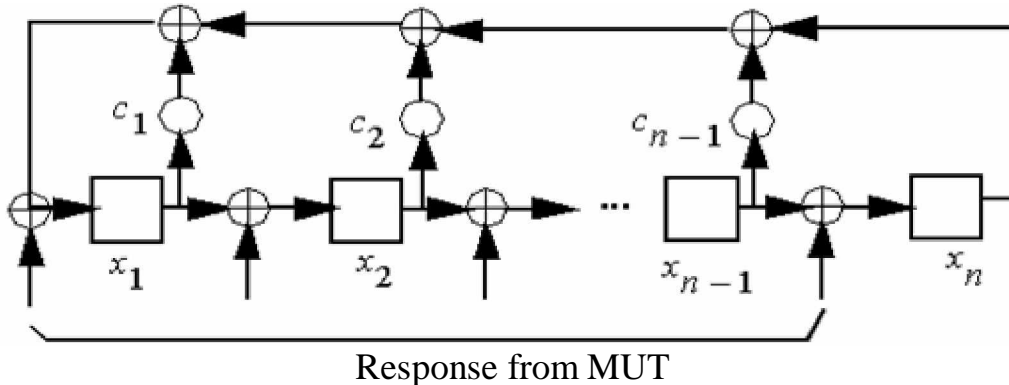


Figure 16: General block diagram of the *misr* cell.

It has two control inputs, CLOCK and START. The data input of this cell is a single 16-bit data $M_IO(15:0)$ and the schematic diagram of this cell is illustrated in Figure 17. The input CLOCK, which is fed from clock CLOCK1_sig in the *timing_control* cell and responsible for triggering the cell at its falling edge. START signal which used to initialize the cell when it goes low. The memory data-bus $M_IO(15:0)$ is compacted. The input DATA to the *misr* cell is processed every CLOCK cycle within the testing gate interval.

Because we have to test one memory device at a time, the *mux_16* cell is placed to select the data from the specified memory device. It takes its input from one of the *memory_io* cells and sends it to the *misr* cell as shown in Fig. 3. The control signal that controls the selection of the memory device is the signal Chip_Sel.

3.6 Explanation of the design of the signature-display cell

The main purpose of this cell is to display the generated signature from *misr* cell on the Spartan-3 seven segment display. It consists of three major parts, a 2-bit binary counter, a 2-to-4 decoder and the *m-sig-display* sub-cell. The 2-bit binary counter controls the 2-to-4 decoder and *m-sig-display* sub-cell. The clock applied to this counter is divider by 1000 as illustrated in Figure 18. The 2-to-4 decoder select specific digit of the display. The *m-sig-display* sub-cell transfers the signature to the seven segment format for proper display. Figure 18 and Figure 19 illustrate the schematic diagram of the *signature-display* cell and the schematic diagram of the *m-sig-display* sub-cell respectively. Figure 20 illustrates the VHDL code of the seven segment conversion.

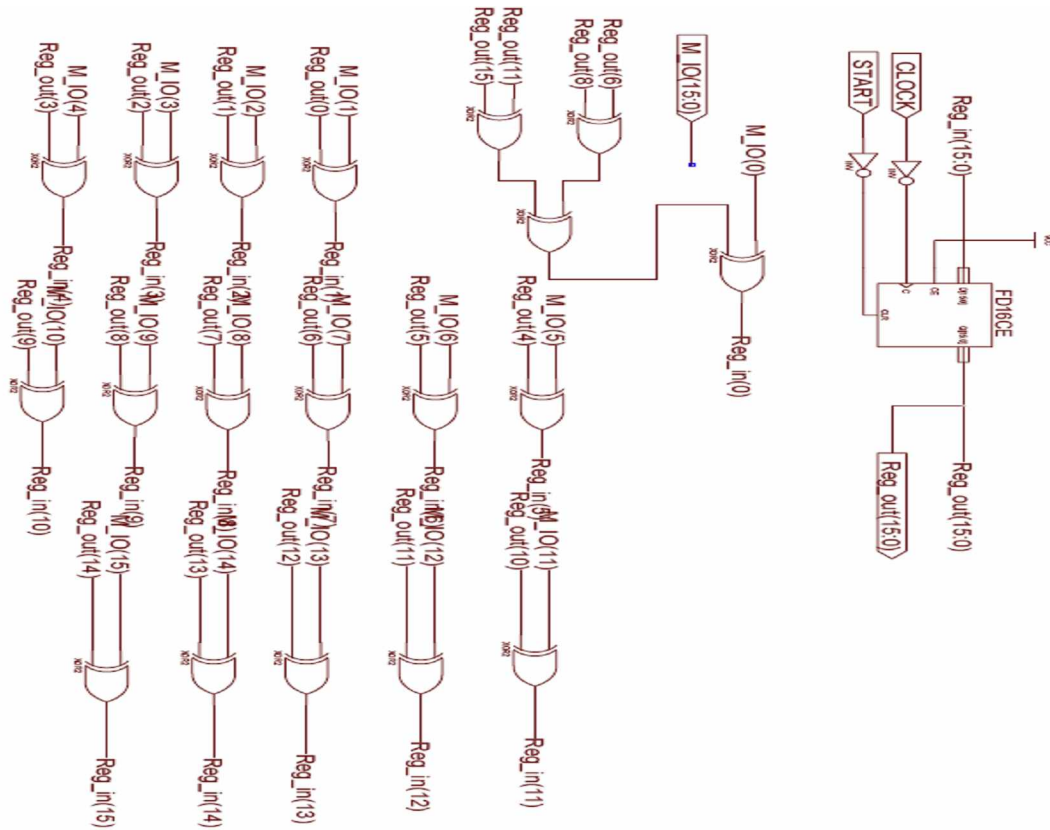


Figure 17: Schematic diagram of the misr cell.

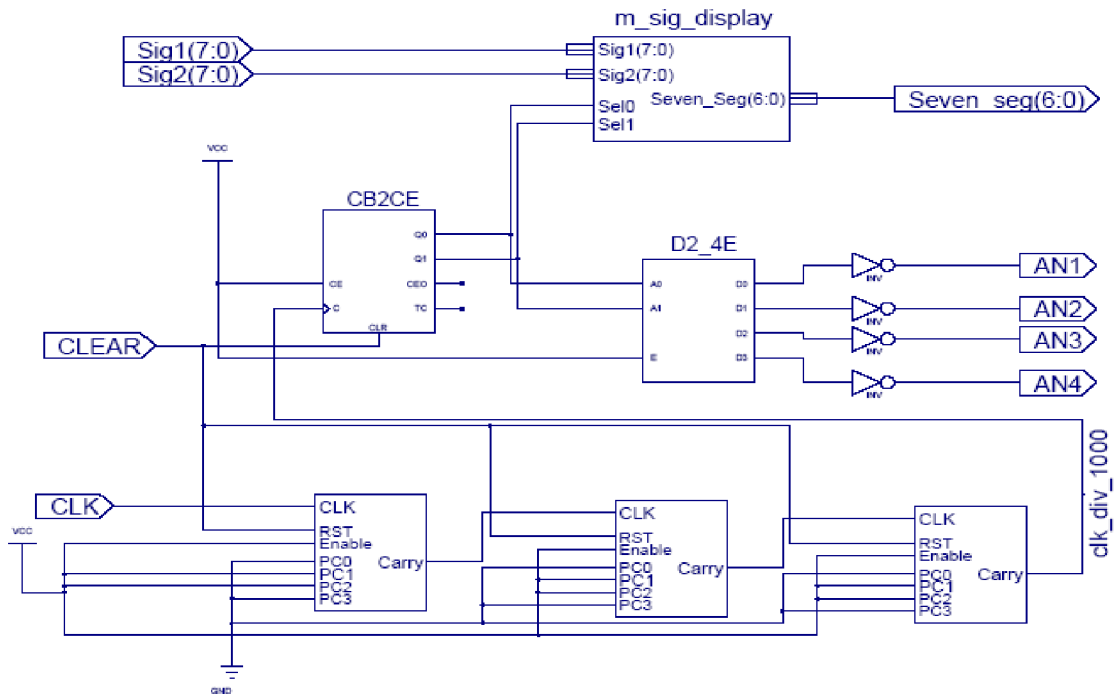


Figure 18: schematic diagram of the signature-display cell.

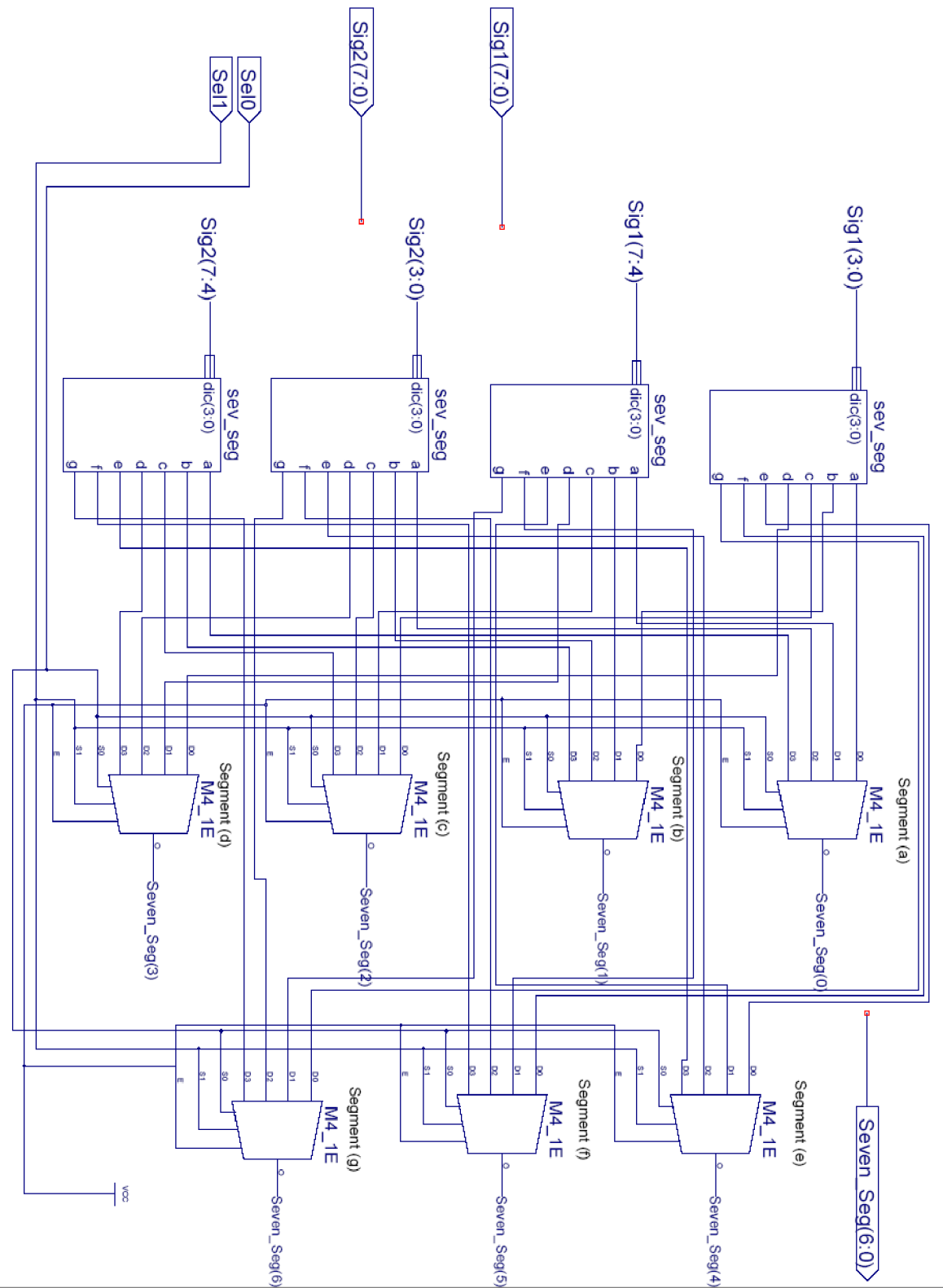


Figure 19: schematic diagram of the m-sig-display sub-cell.

entity sev_seg is

```

port(dic : in std_logic_vector(3 downto 0);
     a,b,c,d,e,f,g : out std_logic );
end sev_seg;

architecture Behavioral of sev_seg is
begin
process(dic)
begin
    case dic is
        when "0000" => a<='0';b<='0';c<='0';d<='0';e<='0';f<='0';g<='1';
        when "0001" => a<='1';b<='0';c<='0';d<='1';e<='1';f<='1';g<='1';
        when "0010" => a<='0';b<='0';c<='1';d<='0';e<='0';f<='1';g<='0';
        when "0011" => a<='0';b<='0';c<='0';d<='0';e<='1';f<='1';g<='0';
        when "0100" => a<='1';b<='0';c<='0';d<='1';e<='1';f<='0';g<='0';
        when "0101" => a<='0';b<='1';c<='0';d<='0';e<='1';f<='0';g<='0';
        when "0110" => a<='0';b<='1';c<='0';d<='0';e<='0';f<='0';g<='0';
        when "0111" => a<='0';b<='0';c<='0';d<='1';e<='1';f<='1';g<='1';
        when "1000" => a<='0';b<='0';c<='0';d<='0';e<='0';f<='0';g<='0';
        when "1001" => a<='0';b<='0';c<='0';d<='0';e<='1';f<='0';g<='0';
        when "1010" => a<='0';b<='0';c<='0';d<='1';e<='0';f<='0';g<='0';
        when "1011" => a<='1';b<='1';c<='0';d<='0';e<='0';f<='0';g<='0';
        when "1100" => a<='0';b<='1';c<='1';d<='0';e<='0';f<='0';g<='1';
        when "1101" => a<='1';b<='0';c<='0';d<='0';e<='0';f<='1';g<='0';
        when "1110" => a<='0';b<='1';c<='1';d<='0';e<='0';f<='0';g<='0';
        when "1111" => a<='0';b<='1';c<='1';d<='1';e<='0';f<='0';g<='0';
        when others => null;
    end case;
end process;
end Behavioral;

```

Figure 20: VHDL code of the sev_seg cell.

4. Experiments and results

In this section, the evaluation of the design performance on the FPGA chip as a real time application system is presented. After applying the MATS test (W0, R0 W1, R1) we get the following signatures after the simulation of the architecture and the same signatures are obtained after downloading of the architecture on the Spartan-3 board, that validates the design:

- For W0, the signature was 0000.
- For R0 W1, the signature was 0033.

- For W0, the SA was 0332.

All the cells for this memory BIST architecture are connected altogether and implemented on FPGA chip Xilinx (X3S200FT256-4). The timing simulation of the complete design is presented to verify proper operation. Figure 21 to Figure 23 illustrate the overall timing diagrams of the three operation modes of the memory testing architecture. The report generated due to this implementation is given as follows.

Design Summary

Logic Utilization:

Number of Slice Flip Flops:	102 out of 3,840	2%
Number of 4 input LUTs:	178 out of 3,840	4%

Logic Distribution:

Number of occupied Slices:	128 out of 1,920	6%
Number of Slices containing only related logic:	128 out of 128	100%
Number of Slices containing unrelated logic:	0 out of 128	0%
Total Number of 4 input LUTs:	178 out of 3,840	4%
Number of bonded IOBs:	157 out of 173	90%
IOB Latches:	48	
Number of GCLKs:	2 out of 8	25%

Total equivalent gate count for design: 2,253

Timing Summary:

Speed Grade: -4

Minimum period: 9.745ns (Maximum Frequency: 102.617MHz)

Minimum input arrival time before clock: 10.689ns

Maximum output required time after clock: 13.779ns

Maximum combinational path delay: 13.898ns

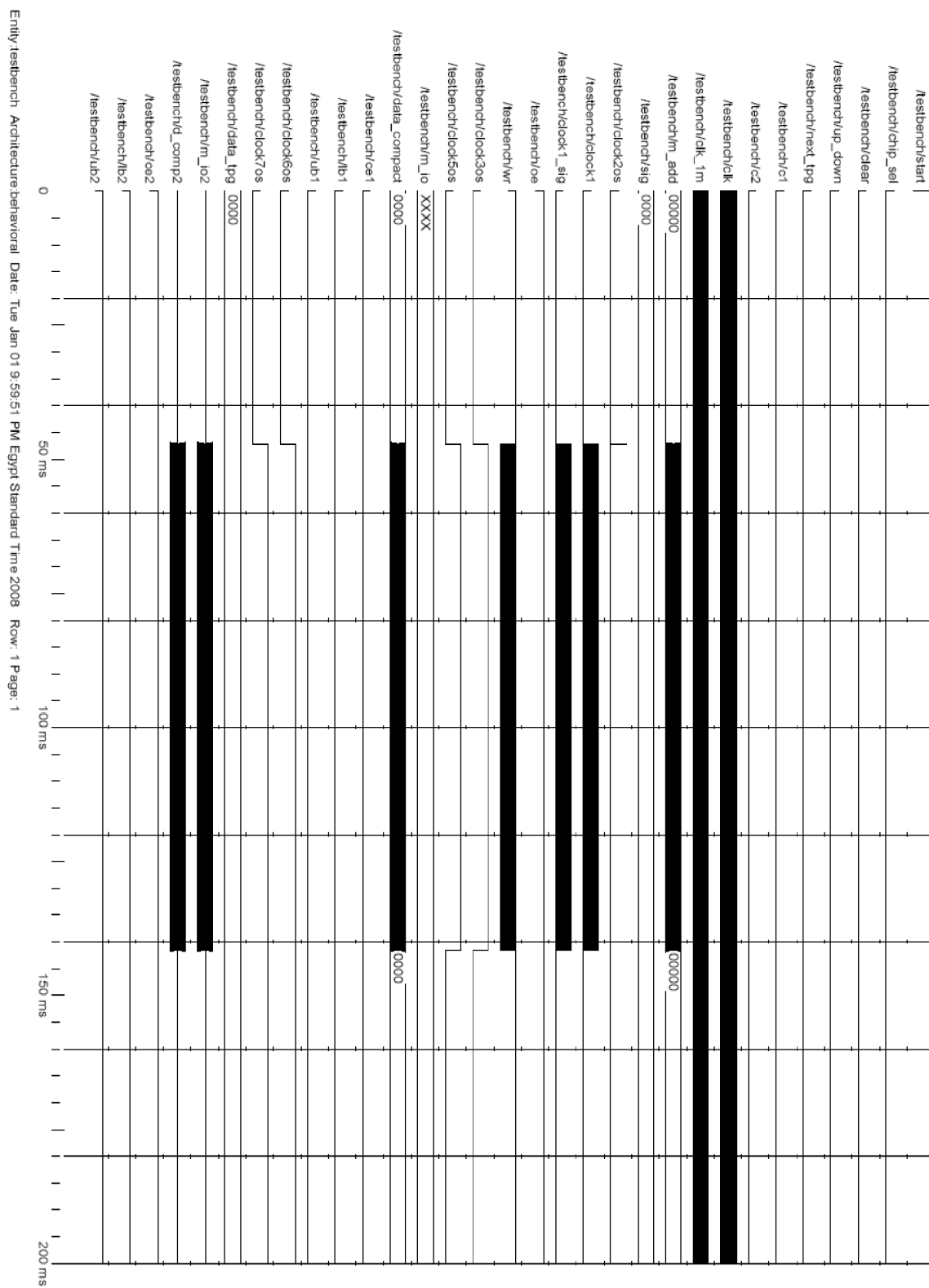


Figure 21: Timing diagram of the testing mode W0.

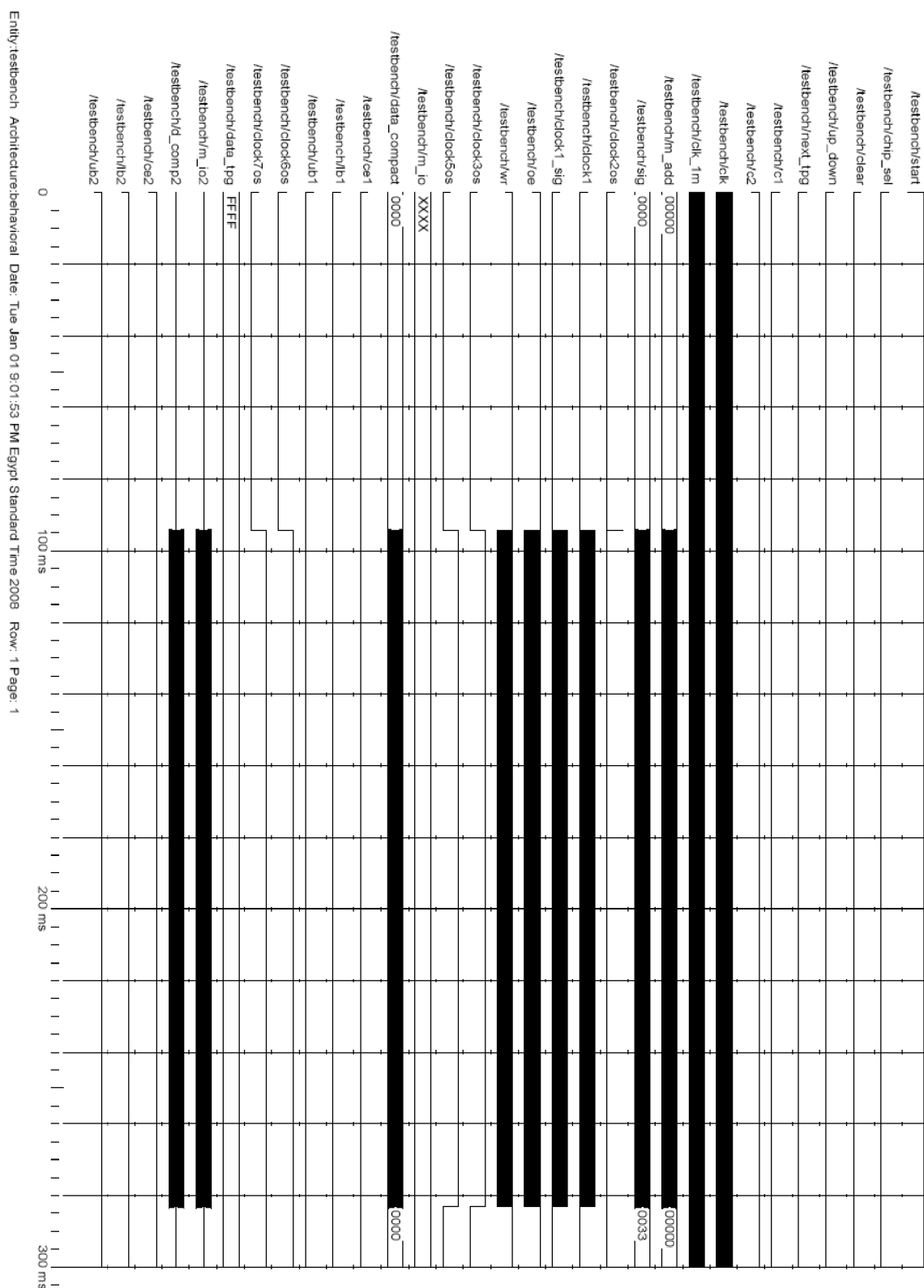


Figure 22: Timing diagram of the testing mode ROW1.

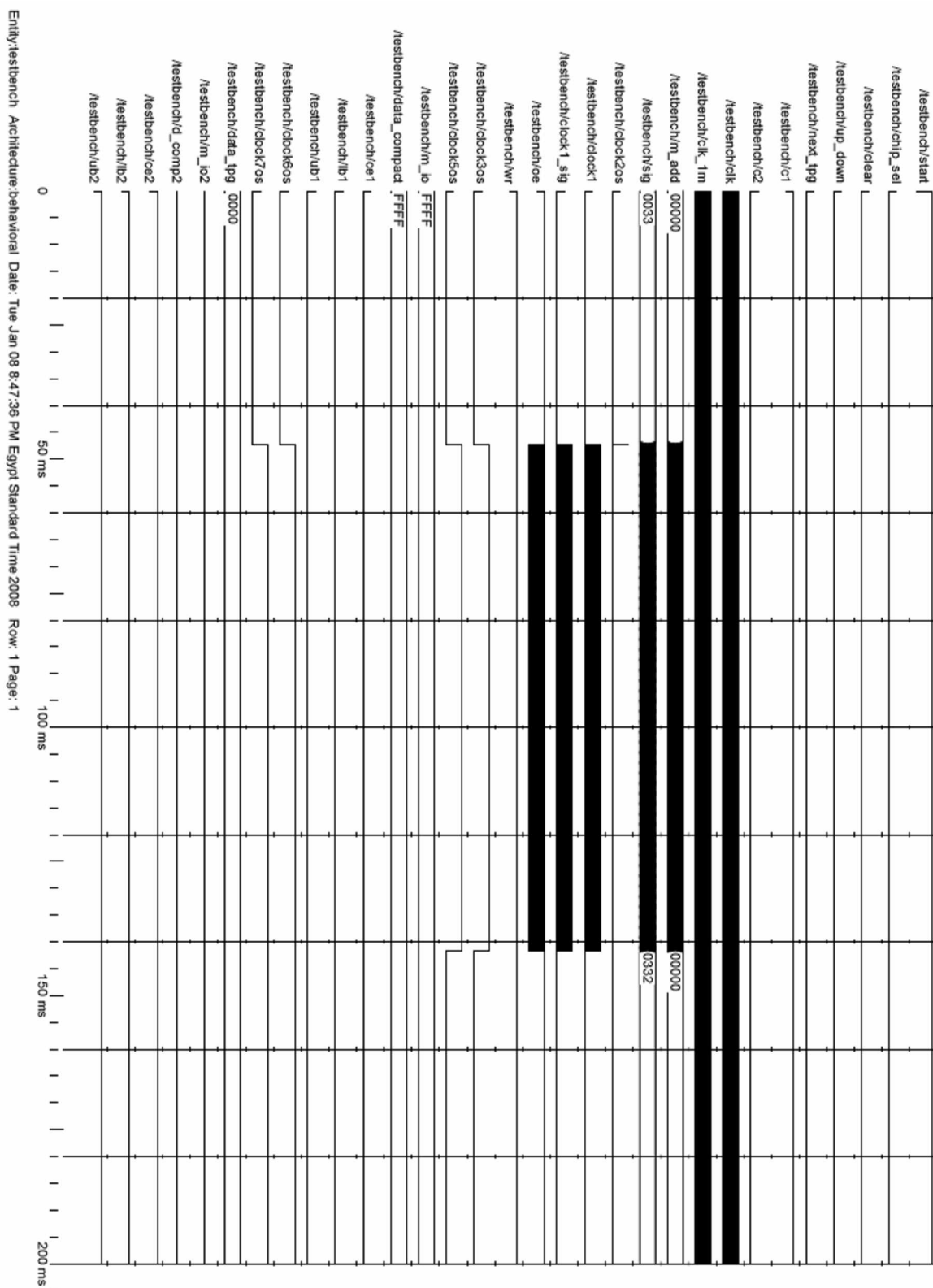


Figure 23: Timing diagram of the testing mode R1.

5. Conclusions:

In this paper, the memory BIST approach for testing memory ICs is implemented using FPGA technology. The schematic diagrams presented here were used to design the portable testable design. This design is represented as an IP (Intellectual property) core that is able to perform the BIST (Built-In Self-Test) for memory on the board.

The hardware experiment results were compared with the simulation results to verify the design performance. The measured signatures in the implemented design are compared with the simulated signatures. The signatures are identically achieved. This design is considered as a BIST testing tool for memory on the complex cards. This approach is considered to reduce the cost of the traditional ATE that consumes long time. So, our objective is to go step in the direction of the portable ATE that consumes less time.

Now, it is easy to test the SRAM memories which are surface mounted on the Spartan-3 board without removing them from the board. The hardware implementation occupies about 2,253 GE on Spartan-3 device, which means that the hardware overhead is 1% of the chip area for the test circuitries.

References:

- [1] Said Hamdioui, *Memory Fault Modeling Trends: A Case Study*, JOURNAL OF ELECTRONIC TESTING: Theory and Applications 20, 245–255, 2004.
- [2] Parag K. Lala, *Digital circuit testing and testability*, Academic Press, 1997.
- [3] R. Dekker et al., *A Realistic Fault Model and Test Algorithms for Static Random Access Memories*, in proc. of IEEE Trans. on Computers. Vol. C9, no. 6, pp. 567–572, 1990.
- [4] I. Schanstra and A.J. van de Goor, *Industrial Evaluation of Stress Combinations for March Tests Applied to SRAMs*, in proc. of IEEE Int. Test Conference, pp. 983–992, 1999.
- [5] Mohamed H. El-Mahlawy, And Ahamed Seddik, *Design and Implementation of New Automatic Testing System For Digital Circuits Based on The Signature Analysis*, 12th International Conference on Aerospace Sciences & Aviation Technology, Military Technical College, Egypt, 29-31 May 2007.
- [6] Xilinx Spartan 3 FPGA Introduction Webpage, Xilinx, http://www.xilinx.com/xlnx/xil_prodcat_landingpage.jsp?title=Spartan-3
- [7] Magdy S. Abadir and Hassan K. Reghbati, *Functional Testing of Semiconductor Random Access Memories*, Computing Surveys. Vol. 15, No. 3, 1983.