Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

6th International Conference
on Electrical Engineering
ICEENG 2008

FPGA implementation of the portable automatic testing system for digital circuits

By

Mohamed H. El-Mahlawy* Ahmed Abd El-Wahab* Al-Emam S. Ragab**

Abstract:

The authors in paper [1] are presented the new automatic test equipment (ATE) for digital integrated circuits based on the signature analysis. This work introduced a complete framework for the testing of the printed circuit boards (PCBs) of the digital integrated circuits. In this paper, the Field Programmable Gate Array (FPGA) implementation of this new ATE is presented. The timing simulation and then design download are presented on the Spartan Xilinx chip (X2S400EFT256-7). In this paper, the concept of the portable ATE is presented that reduces the complexity of the traditional ATE. This compacted testing system approach is designed to apply the test pattern to the circuit under test (CUT) and to compact the response of the CUT by signature analyzer. The timing controller and the parallel port of the personal computer (PC) generate all required signals to control all steps of the test cycle for proper operation.

Keywords:

Automatic Test Equipment, and Testing of electronic circuits

* Egyptian Armed Forces
** Masr International University

1. Introduction:

With the advent of the complex integrated circuits, the data stream bits at the available test points of the printed circuit boards become very complex. Testing digital circuits for correct operation after manufacturing is an important problem. The problem is how to apply the required test patterns to the CUT and to analyze the response to locate the faulty components so that the circuit board returned to service. The ATE achieves this objective. It produces input test patterns and check binary outputs for correctness. A number of ATE systems are available, the variations are extensive, but most of them fall into two main categories [1-6]: Functional tester, and In-circuit tester. An in-circuit tester examines each component on the board individually. The in-circuit tester requires a special kind of test fixture to provide electrical connections to all component pins [7]. A functional tester is important to the final user of the circuit board. It verifies that the board performs the functions it was designed for. Only board inputs and outputs need to be tested. The required test fixture is the edge connector that it is inexpensive and simple to construct.

Pseudorandom testing is widely used for testing digital integrated circuits. Simple hardware circuits can generate pseudorandom test patterns, for example, an autonomous LFSR (ALFSR) [8]. The main concern is to choose the right test pattern generator to provide high fault coverage, while keeping the test set length short. For a given pseudorandom test set, fault coverage is generally obtained by fault simulation [9]. If the fault coverage is found unacceptable, the length of the test set is increased. Signature analysis is a compaction technique that detects errors in data stream bits caused by hardware faults. The *signature analyzer* (SA) compacts the output response for each output of the CUT into a shorter sequence. A smaller number of bits reduce the number of bits that must be stored and compared. Figure (1) illustrates as an example of the schematic of a 4-stage simple signature analyzer. This circuit resembles an LFSR but has an additional PROBE input that is connected to an output of the CUT. After all input test patterns are applied, the shift register will contain a *signature*. A signature can be computed for known correct output values and then compared to the measured signature [1]. When the good signatures and measured signatures are differed, a fault has been detected. The SA can detect almost all faults in the output of a CUT while requiring the storage of far fewer bits.

The aliasing probability of the signature analyzer is estimated. For an input data sequence of length *m*, if all possible errors in a sequence are equally likely, then the aliasing probability of an *n*-stage signature analyzer approaches 2^{-n} for $m >> n$ [10]. The error will appear in the data stream only if the test pattern generator (TPG) generates the test pattern that detects the fault.

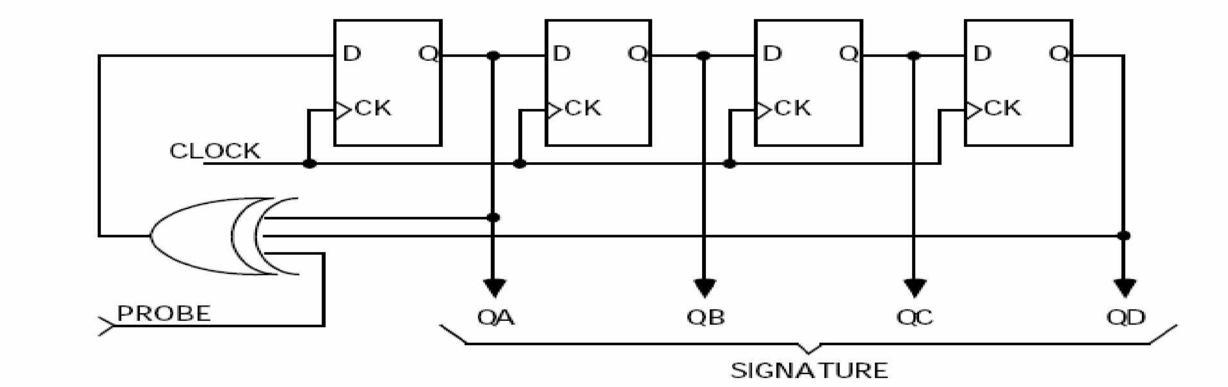


Figure (1): *4-stage simple signature analyzer.*

The objective of the design architecture in [1] is to make the generated signature repeatable and stable [1]. The controlling processing of the test cycle is required to achieve this objective. The TPG stimulates all nodes in the CUT. The stimulated node makes the SA compact all stream data bits into a *signature*. By supplying known input test patterns to a CUT, unique signature can be generated at various nodes in the circuit. Measuring an incorrect signature during troubleshooting will accurately indicate an incorrect waveform for that node as long as the error appears in the data stream and clocks into the SA.

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 FPGA gives users high performance, abundant logic resources, and a rich feature set, all at low price. This means that you can design, program, and make changes to your system whenever you wish. 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. In this paper, the FPGA implementation using the Spartan Xilinx chip (X2S400EFT256-7) of the presented testing architecture in [1] is presented. The concept of the portable ATE is presented to reduce the complexity of the traditional ATE.

In this paper, section 2 presents the concept of ATE architecture in [1]. FPGA design steps of the potable ATE will be presented in section 3. The experimental results and the conclusions will be discussed in section 4 and section 5, respectively.

2. Concept of the ATE architecture presented in [1]

The paper, presented in [1], introduced a new ATE for fault detection and fault location to the net level on the printed circuit board (PCB), containing digital ICs. It performs the hot functional test for TTL and CMOS digital IC technologies. This system consists of two main integrated parts hardware, and software. The hardware part is equipped with all basic parts of the generalized ATE. The basic block diagram of the hardware part is shown in Figure (2). It is composed of the TPG to exercise the CUT as a stimulus, the SA to compact the response of the measured net as a test response compactor, the timing controller synchronized with either internal or external CLOCK to control each part of the system accurately, and a simple test fixture that provides the system with an electrical and mechanical interface with the CUT. In addition, it is equipped with the interface unit to enable the automatic signature transfer to the PC and generate the deterministic test patterns. The measurement technique based on SA gives a 99.999988 % worst-case probability of detecting an error.

This system has two main basic modes of operation: Pseudorandom testing mode (PRT mode), and Deterministic testing mode (DT mode). The TPG, based on pseudorandom testing (PRT), is called pseudorandom TPG (PRTPG) that guarantees at least 95 % fault coverage of stuck-at fault model [1]. The TPG based on deterministic testing (DT) is called deterministic TPG (DTPG). It has arbitrary test length according to the test patterns calculated from algorithmic methods. These algorithms support the detection of different fault models [11-12]. The software part with the interface unit is responsible to automatically transfer these calculated test patterns to the CUT. The output of both TPGs can drive different technologies such as TTL and CMOS digital ICs.

In the PRT mode, the switching between internal and external clock are avaiable using the SW_CK. The system has master CLEAR. The gate window is controlled by the control signal GATEOS. The PRTPG operates in two mode of operation; free-running mode and one-shot mode. In free-running mode, the gate is continuously opened and closed every 100 ms. The stability of the signature on the display ensures the proper operation. In one-shot mode, the gate is opened once and then it is closed. If you want to open the gate to calculate new signature, you need to press CLEAR. The switching between the free-running mode and one-shot mode is done using MODE_SEL. The RCLK, generated from the clock, is used to synchronize and control the sample rate of the probe input DATA and the PRTPG. So, the input DATA to the SA is processed every RCLK cycle within the gate interval. The output of the PRTPG is asserted in the rising edge of RCLK and the SA is asserted in the falling edge. This provides the test pattern, generated from the PRTPG, enough time to propagate through the ICs built in the electronic card before the acquisition of the DATA. The gate can be controlled accurately by timing controller card. RCLK is the gated clock inside the gate window

(99999 clocks). PRCLEAR is the signal that clears the PRTPG in the beginning of the gate. SIG_CLEAROS is the signal that clears the SA in the beginning of the gate. After 99999 clocks, the gate is closed. Then, the outputs of the PRTPG have constant binary values and the outputs of the SA has proper signature.

GATEOS controls the time interval of the gate. It is asserted at the falling edge of RCLK in starting of the gate and the closing of it. GATEOS1 is the half clock shift version of GATEOS. GATEOS1 controls the time to properly latch the signature after the testing processing finishes with half clock.

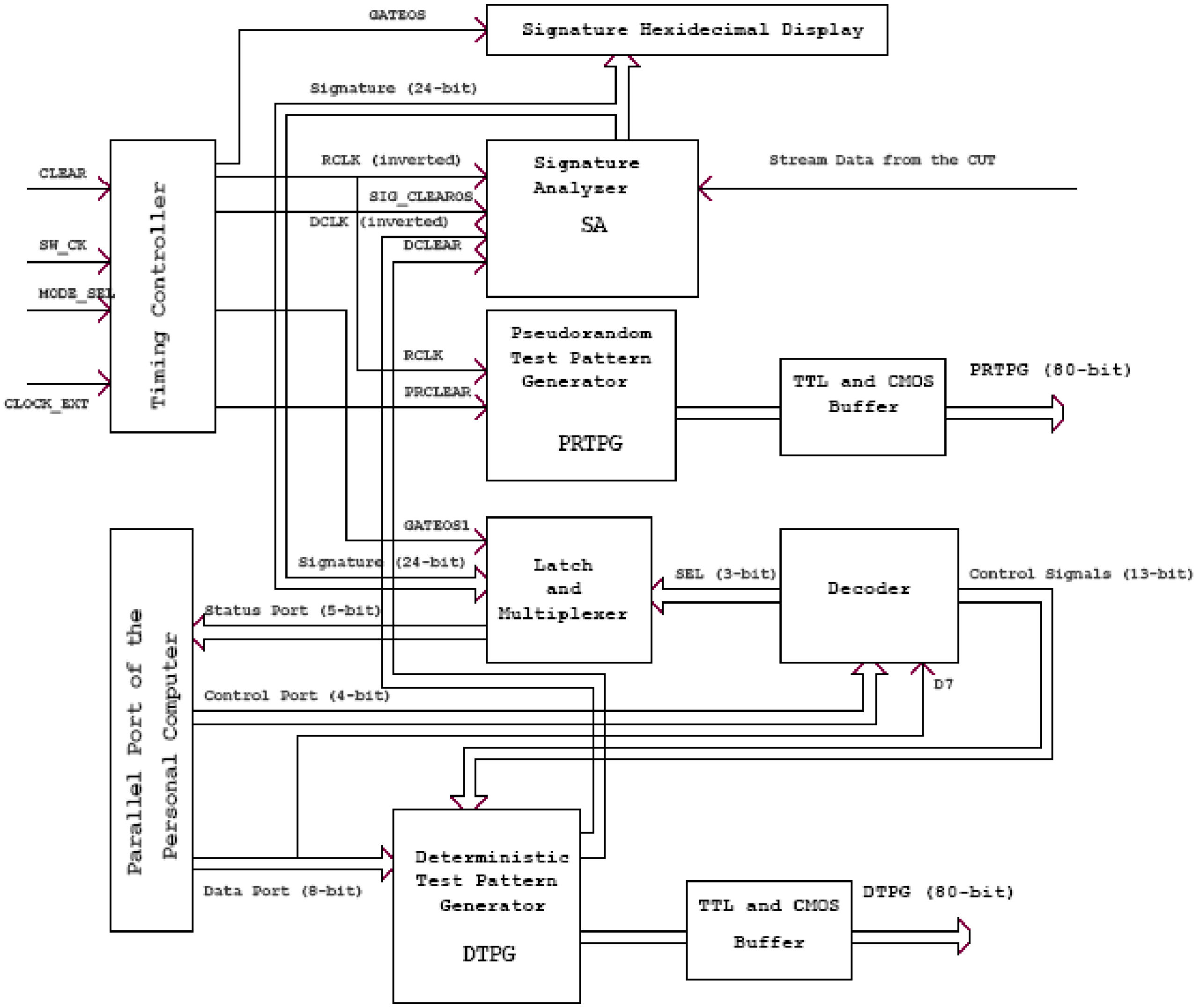


Figure (2): *Block Diagram of the basic architecture of the hardware part.*

In the DT mode, the deterministic test patterns are generated from the personal computer through the parallel port (Fig. 2). In this case, the number of the clocks inside the gate is variable depends on the required test patterns to deterministically exercise the CUT according to the calculation of the algorithmic methods [11-12] (path sensitization, D algorithm, PODEM, and FAN). These algorithms support the detection of the stuck-at fault. Other algorithms support the detection of different faults such as bridging faults and stuck-open faults [11-12]. The software part is responsible to retrieve these test patterns and generates all required control signals to automatically transfer these patterns to the CUT through the DTPG. The DCLK is used to synchronize and control the sample rate of the probe input DATA and the DTPG. DCLEAR is the signal that clears the SA in the beginning of the gate. After the gate is closed, the proper signature is displayed. The generated signature from either pseudorandom testing or deterministic testing is automatically transferred to the PC.

Some CUT inputs of the electronic cards needs fixed logic state and the other can be tested using the random binary signals. The hybrid between pseudorandom testing and deterministic testing mode is divided the CUT inputs into two sets. One set of CUT inputs that needs fixed logic state is connected to single deterministic test pattern generated from DTPG and the other set is connected to the PRTPG. The test patterns applied to CUT inputs is the concatenation of the PRTPG and DTPG.

The interface unit of this system is composite of the latch, a multiplexer module and a decoder module interfaced through the parallel port of the PC. From Fig. 2, the decoder is accepted the four input signals from control port. The generated output signals from the decoder are the encoded signals that control the proper operation of the DTPG module and the multiplexer. The control port and the decoder generate all required control signals. Due to the limited number of the input signals of the status port, the signature is multiplexed. The multiplexer is controlled to automatically transfer the signature to the PC after it is latched. The signature is stored in a file for fault detection and fault location processing.

The system is designed to make the overall timing control, generate test pattern to hot exercising the CUT, receive the response from the CUT, compact the response from a golden reference CUT in programming mode or from the malfunction CUT in the testing mode, and store the reference signature to a database file for that CUT. The software is designed to characterize the CUT. It automatically makes the comparison between the good signature and the measured one, carries out test analysis to determine the source faulty node(s). This system represents an interactive testing system providing the troubleshooter with an easy access to an efficient testing system for locating the source faulty node(s) [1].

3. FPGA design steps of the potable ATE

The FPGA-based development board provides an inexpensive and expandable platform on which to design and implement digital circuits of all kinds. This board, illustrated in

Figure (3), includes the Spartan Xilinx chip (X2S400EFT256-7),182 user I/Os routed to six standard 40-pin expansion connectors, a socket for a JTAG-programmable 18V04 configuration Flash ROM, a 50 MHz oscillator, and a socket for a second oscillator. The board can be easily adapted with several existing expansion boards, so more complex designs can easily be achieved.

This section presents the design and implementation of the hardware part of the portable ATE, based on FPGA technology. Figure (4) illustrates the block diagram of this portable ATE on the Spartan Xilinx chip (X2S400EFT256-7) board. This system is mainly designed to work in PRT mode and DT mode. The *timing_controller* cell is designed to control the testing process in the PRT mode. The *prtpg* cell is designed to generate pseudorandom test patterns for CUT in the PRT mode. The *signature_analyzer* cell is designed to compact the stream test pattern response from CUT test points into the corresponding signatures. The *dtpg* cell is designed to generate deterministic test patterns for CUT in the DT mode. This cell is controlled by *decoder_4_to_16* cell. The programmable control register in the *decoder_4_to_16* cell supports new software instructions in this design to extent the testing capabilities. Before the explanation of these cells, we need to focus on the functionality of the *DividerBy50* cell. The schematic diagram and the timing diagram of this cell are illustrated in Fig. 5 and Fig. 6, respectively. This cell divides the clock CLK_INB of the development board (50 MHz) into another clock Clock_Osc of the testing operation (1 MHz) in the PRT mode.

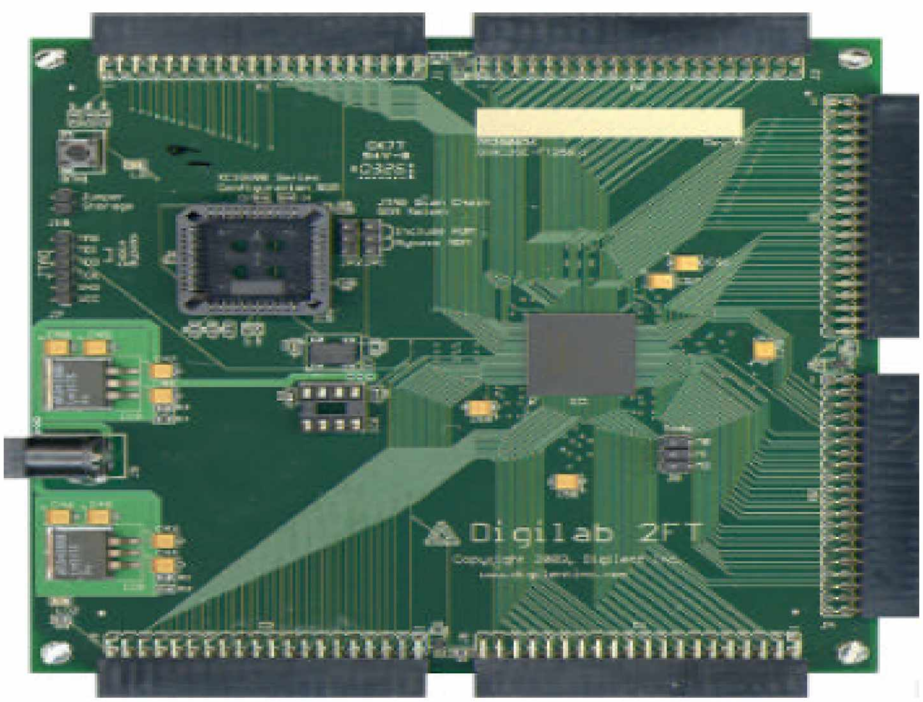


Figure (3): *The Spartan Xilinx chip (X2S400EFT256-7) board for the hardware part of the portable ATE.*

In Figure (5), there are two programmable dividers (*divideby5* cell). Each one is programmable to divide the clock by 5. The divided signals CARRY1 and CARRY2 are illustrated in Fig. 6. The last D-type Flip-Flop is used to divide by 2. It generates symmetric clock CLK_DIV (1 MHz) illustrated in Fig. 6 with clock Clock_Osc. From Figure (6), the assertion of signal CARRY1 at the falling edge of the clock CLK_INB. Also, the assertion of signal CARRY2 at the falling edge of the clock CARRY1. However, the assertion of the programmable dividers (*divideby5* cell) at the rising edge of the input clock of the cell. This situation removes any glitches and Fig. 6 illustrates the glitch-free of all signals generated from this cell (*DividerBy50* cell) after the timing simulation.

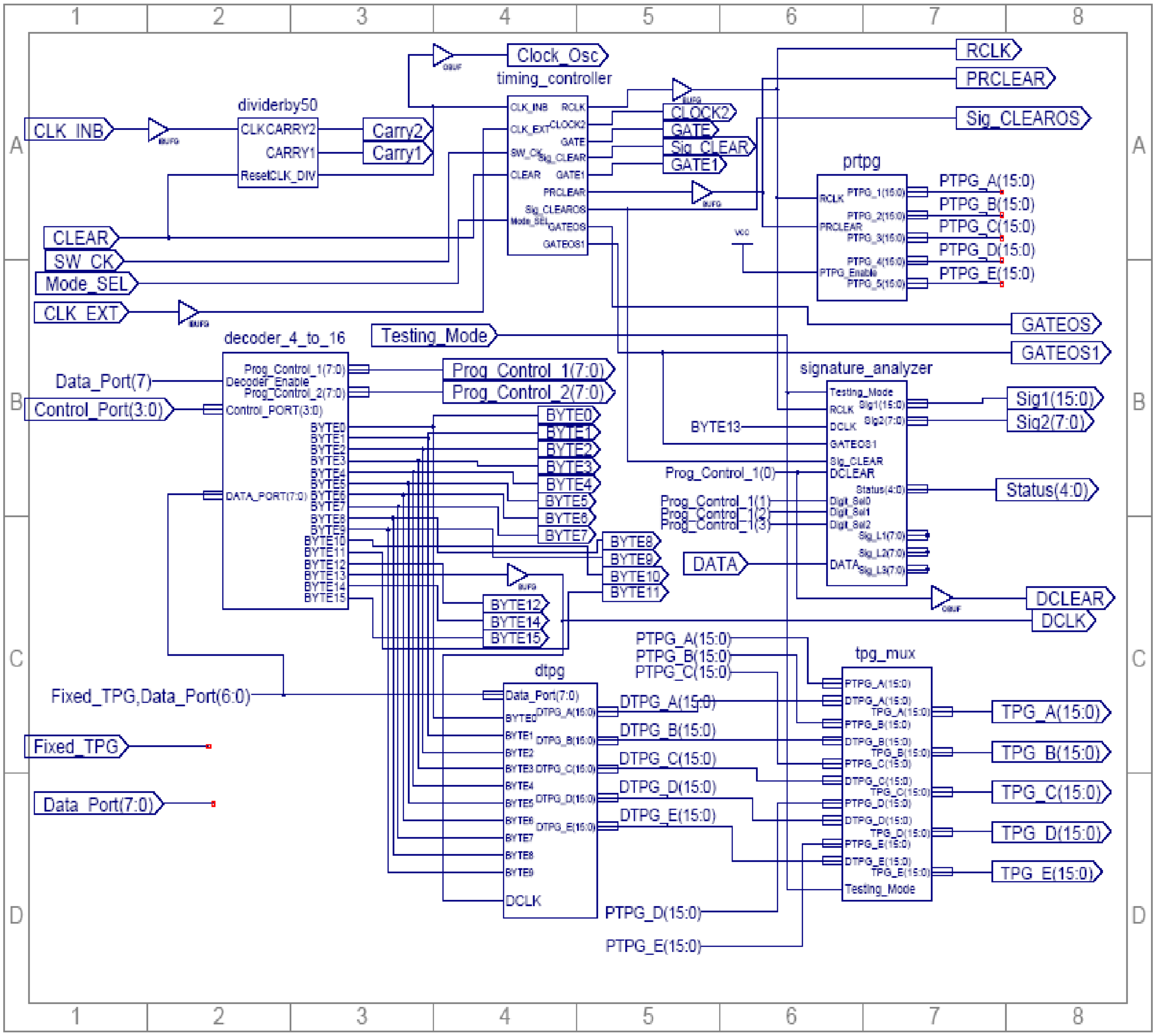


Figure (4): *Block diagram of the FPGA implementation of the hardware part of the portable ATE.*

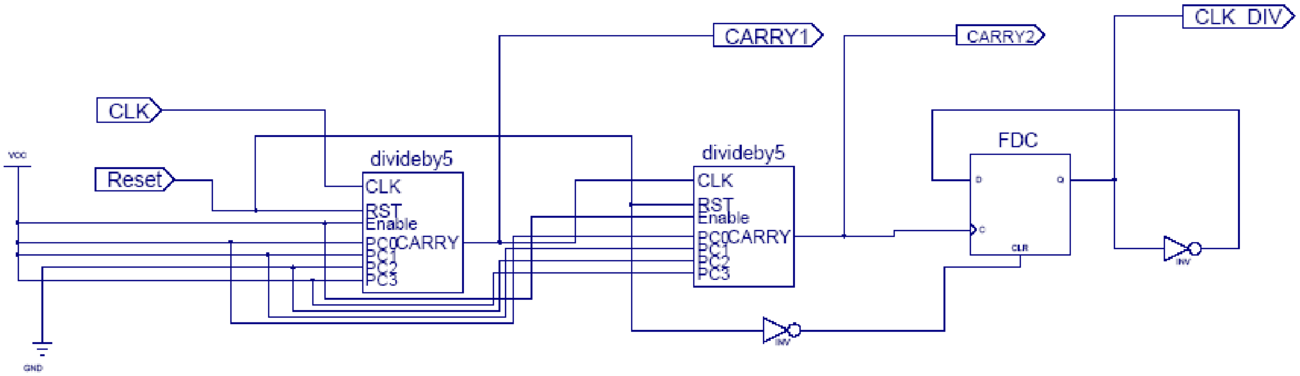


Figure (5): *Schematic diagram of the DividerBy50 cell.*

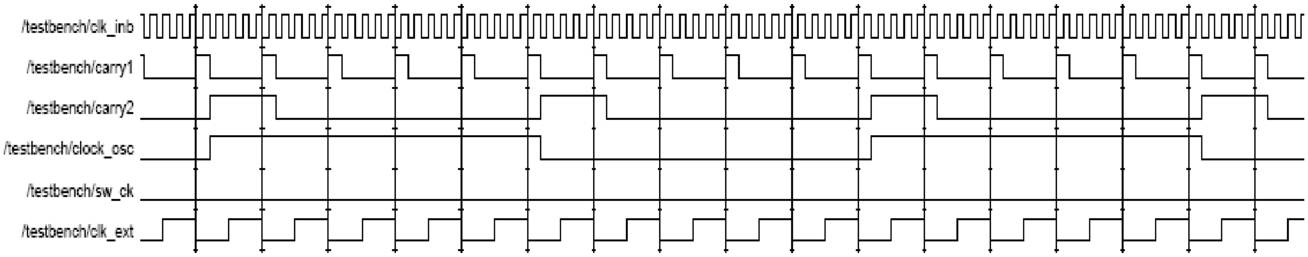


Figure (6): *Timing diagram of the DividerBy50 cell.*

3.1 Explanation of the design of the timing_controller cell

The timing_controller cell is responsible for the timing synchronization of the proper testing operation in the PRT mode. The schematic diagram of this module is illustrated in Figure (7). This cell has five input signals; SW_CK, MODE_SEL, CLEAR, and clock CLK_EXT and clock CLK_INB. SW_CK is the signal that switches between the internal and external synchronized clock through the MUX. When SW_CK sets LOW, the internal clock (1 MHz) is the synchronized clock CLK_INB. When SW_CK sets HIGH, the external clock is the synchronized clock CLK_EXT.

This cell is composed of the three parts. The first part is the *TESTING GATE GENERATION*. This part is responsible to generate the testing gate. It is composed of the five 4-bit BCD counters and four Flip-Flops as shown in Figure (7). From Fig. 7, the assertion of the five 4-bit BCD counters at the rising edge of the clock, and the assertion of four Flip-Flops at the falling edge of the clock. This situation removes any glitches.

The second part is the *TESTING SIGNAL SYNCHRONIZER*. This part is responsible to generate the output control signals; RCLK, CLOCK2, Sig_CLEAR, GATE, and GATE1. It is composed of six Filp-Flops and other random logic gates as shown in Figure (7).

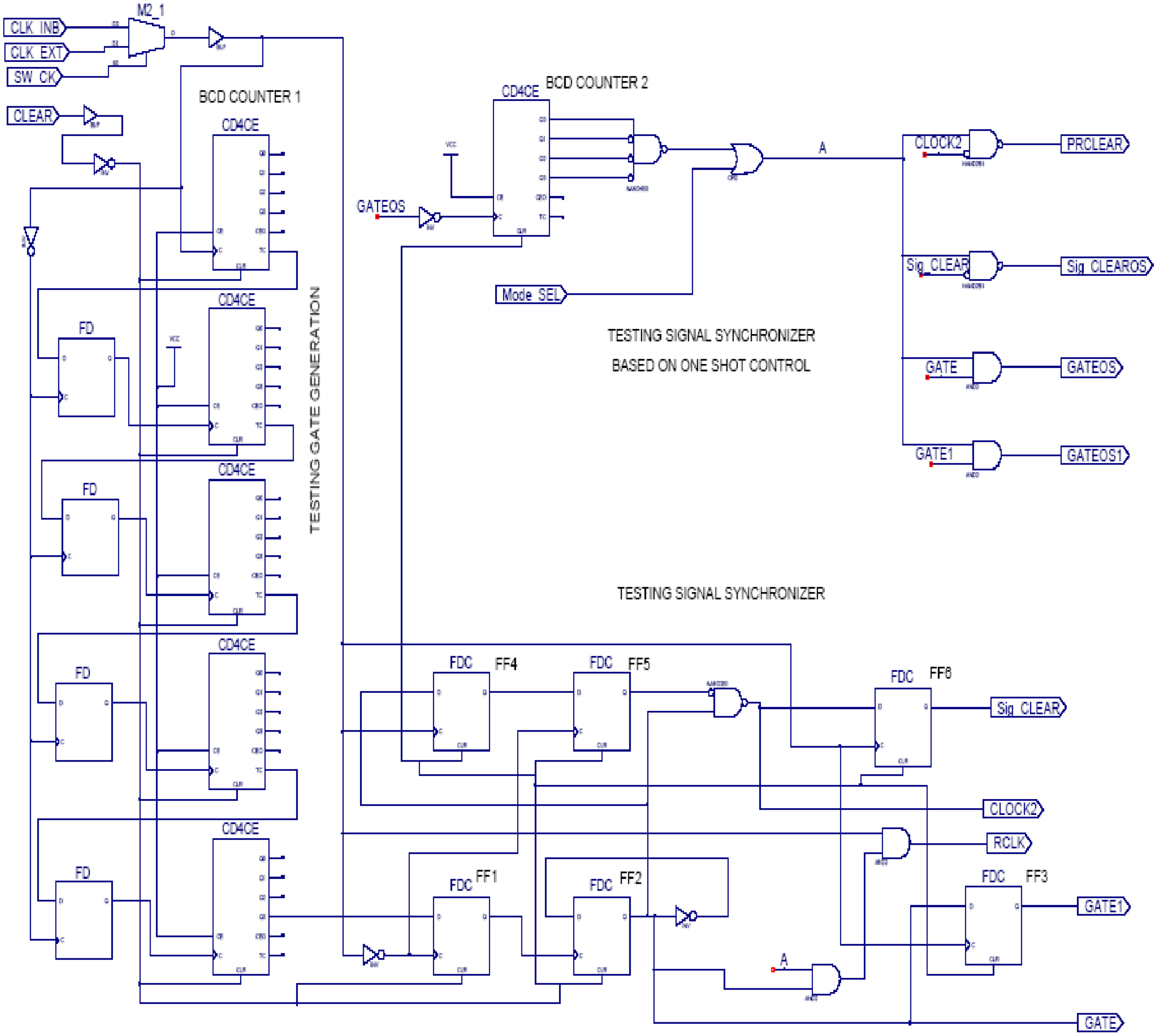


Figure (7): *Schematic diagram of the timing_controller cell.*

The third part is the *TESTING SIGNAL SYNCHRONIZER BASED ON ONE SHOT CONTROL*. This part is responsible to generate the output control signals; PRCLEAR, Sig_CLEAROS, GATEOS, and GATEOS1. It is composed of a 4-bit BCD counter and other random logic gates as shown in Figure (7).

The *TESTING GATE GENERATION* part with the Flip-Flop FF1, and Flip-Flop FF2 generate the control signal GATE, which is asserted at the falling edge of the clock in starting of the testing gate and the closing of it. GATE1 is the half clock shift version of GATE through Flip-Flop FF3. Figure (8) and Figure (9) illustrate the timing of these control signals.

The *TESTING GATE GENERATION* part with the Flip-Flop FF4 and Flip-Flop FF5 generate the control signal CLOCK2 that clears the PRTPG at the beginning of the testing gate. It is asserted at the falling edge of the clock. Sig_CLEAR is the half clock shift version of CLOCK2 through Flip-Flop FF6. It clears the *signature_analyzer* cell at the beginning of the gate. It is asserted at the rising edge of the clock. Figure (8) and Figure (9) illustrate the timing of these control signals. From Figure (7), RCLK is the gated clock inside the testing gate. It is used to synchronize and control the sample rate of the probe input DATA and the PRTPG. Therefore, the input DATA to the *signature_analyzer* cell is processed every RCLK cycle within the gate interval. After 100000 clocks, the gate is closed. Then, the outputs of the PRTPG has constant binary values as shown in Fig. 10 and the outputs of the *signature_analyzer* cell has proper hexadecimal signature “299BD5” (the input DATA is HIGH). The output of the PRTPG is asserted in the rising edge of RCLK and the output signature from the *signature_analyzer* cell is asserted in the falling edge. This provides the test pattern, generated from the PRTPG, enough time to propagate through the integrated circuits built in the electronic card (CUT) before the acquisition of the DATA.

The PRTPG operates in two mode of operation, free-running mode, and one-shot mode. The switching between the free-running mode and one-shot mode is done using MODE_SEL. When MODE_SEL sets HIGH, the free-running mode is used. In free-running mode, the gate is continuously opened and closed every 100 ms. When MODE_SEL sets LOW, the one-shot mode is used. The *TESTING SIGNAL SYNCHRONIZER BASED ON ONE SHOT CONTROL* part is responsible to open the testing gate once. If you want to open the testing gate again to generate new signature, you need to press CLEAR. CLEAR is used to initialize the hardware system.

The output control signals; PRCLEAR, Sig_CLEAROS, GATEOS, and GATEOS1 perform the same functions of output control signals; CLOCK2, Sig_CLEAR, GATE, and GATE1, respectively but under the control of MODE_SEL. GATEOS1 controls the time to properly latch the signature in the *signature_analyzer* cell after the testing operation finishes with half clock and GATEOS is connected to display the signature only when it is asserted low. These output control signals are considered the main signals that assure the proper operation of the portable ATE throughout the testing cycle.

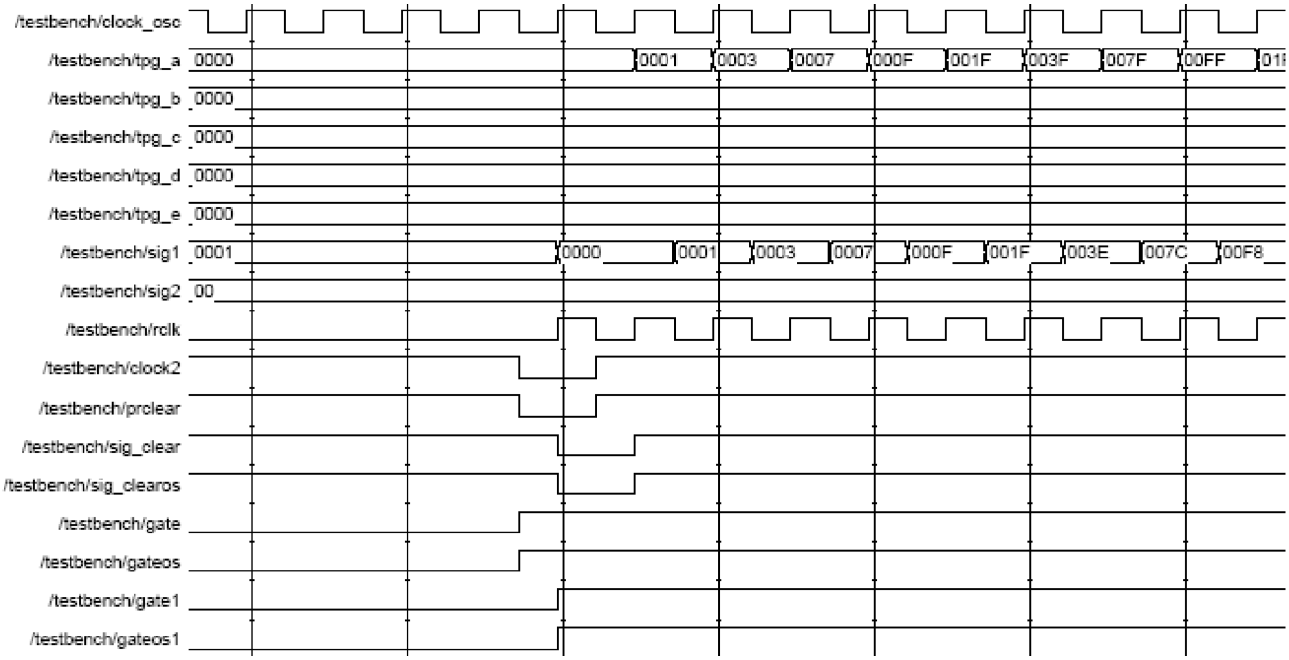


Figure (8): *Timing diagram of the starting testing gate in the PRT mode.*

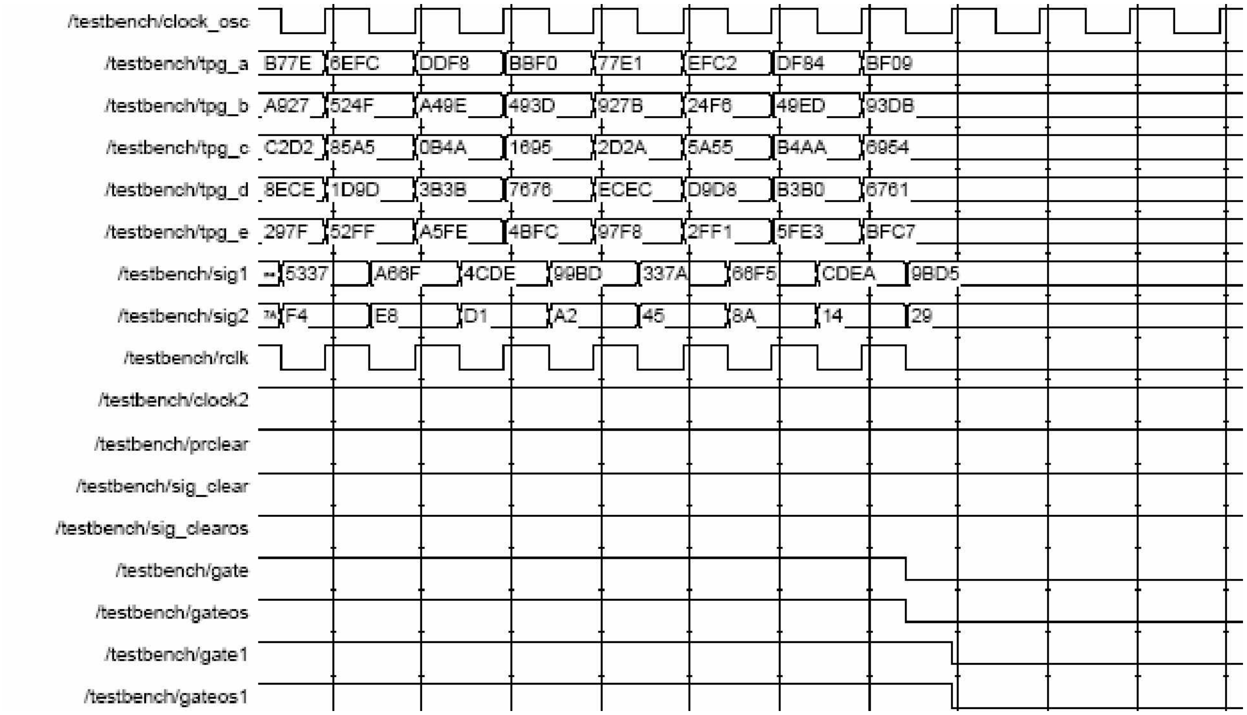


Figure (9): *Timing diagram of the ending testing gate in the PRT mode.*

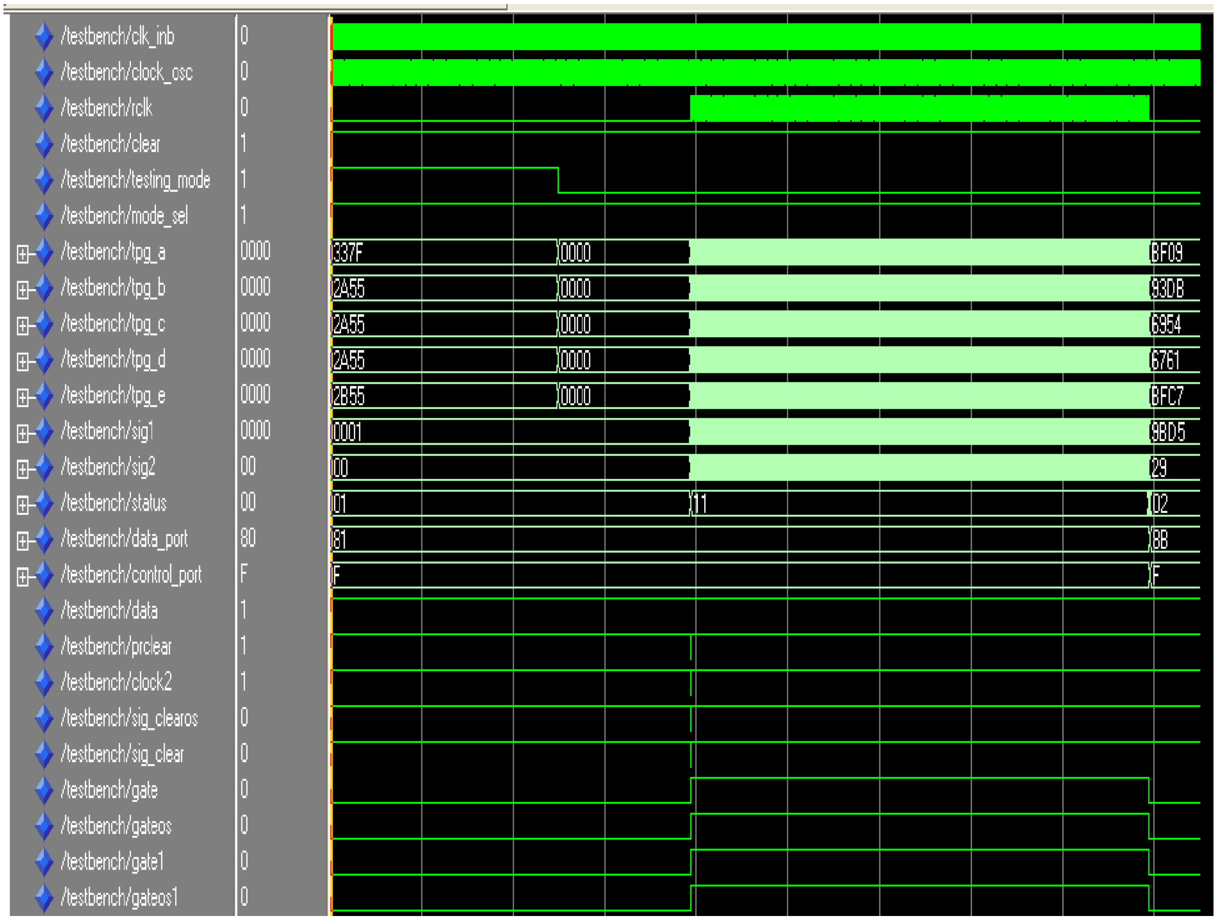

/testbench/clk_inb 0
/testbench/clock_osc 0
/testbench/rclk 0
/testbench/clear 1
/testbench/testing_mode 1
/testbench/mode_sel 1
/testbench/tpg_a 0000 337F 0000 BF09
/testbench/tpg_b 0000 2A55 0000 93DB
/testbench/tpg_c 0000 2A55 0000 6954
/testbench/tpg_d 0000 2A55 0000 6761
/testbench/tpg_e 0000 2B55 0000 BFC7
/testbench/sig1 0000 0001 9BD5
/testbench/sig2 00 00 29
/testbench/status 00 01 11 02
/testbench/data_port 80 81 8B
/testbench/control_port F F F
/testbench/data 1
/testbench/prclear 1
/testbench/clock2 1
/testbench/sig_clearos 0
/testbench/sig_clear 0
/testbench/gate 0
/testbench/gateos 0
/testbench/gate1 0
/testbench/gateos1 0

Figure (10): *Timing diagram of the testing operation in the PRT mode.*

3.2 Explanation of the design of the prtpg cell

The hardware part of the portable ATE is equipped with the PRTPG as a stimulus in the PRT mode. This generator is the Autonomous Linear Feedback Shift Register (ALFSR). The length of this generator is 79 bit. The PRTPG designed in the portable ATE is composed of five SR16CE modules and an XNOR gate. It is illustrated as shown in Figure (11). The SR16CE module is a 16-bit serial-in parallel-out shift register. The shift in this register is a shift-left serial input (SLI), parallel outputs (Q), and Clock Enable (CE) and asynchronous clear (CLR) inputs. When CE is High and CLR is Low, the data on the SLI input is loaded into the first bit of the shift register during the rising edge clock (C) and appears on the Q0 output. During subsequent rising edge clocks, data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth). Five registers are cascaded by connecting the last Q output (Q15 for SR16CE) of one module to the SLI input of the next module and connecting clock, CE, and CLR in parallel. This generator has primitive polynomial $1 + x^9 + x^{79}$ [1, 8]. The feedback from the stage (9) of the first cell and the stage (15) of the last cell are connected to the input of an XNOR gate, and the output of the XNOR gate is connected to the SLI input of the first module. This generator forms the ALFSR with primitive polynomial $1 + x^9 + x^{79}$.

The main concern with the pseudorandom testing approach is to choose the right test pattern generator to provide high fault coverage, while keeping the test set length short. For a given pseudorandom test set, fault coverage is generally obtained by simulation (fault coverage can be determined by fault simulation). If the fault coverage is found unacceptable, the length of the test set is increased. For large circuits, when simulation is expensive, probabilistic measures are used to compute the length of the test set for the desired fault coverage. The length of test set for a desired level of fault coverage should be much shorter than that required for exhaustive testing. It is requires to get a PRTPG with high fault coverage. This generator has test length 100000 test patterns. The period of the internal clock equals 1 μs, therefore the testing gate in the PRT mode spends 100 ms. This test length guarantees 95 % fault coverage of stuck-at fault model [1]. The output of the PRTPG can drive different technologies such as TTL and CMOS digital ICs through the suitable drivers.

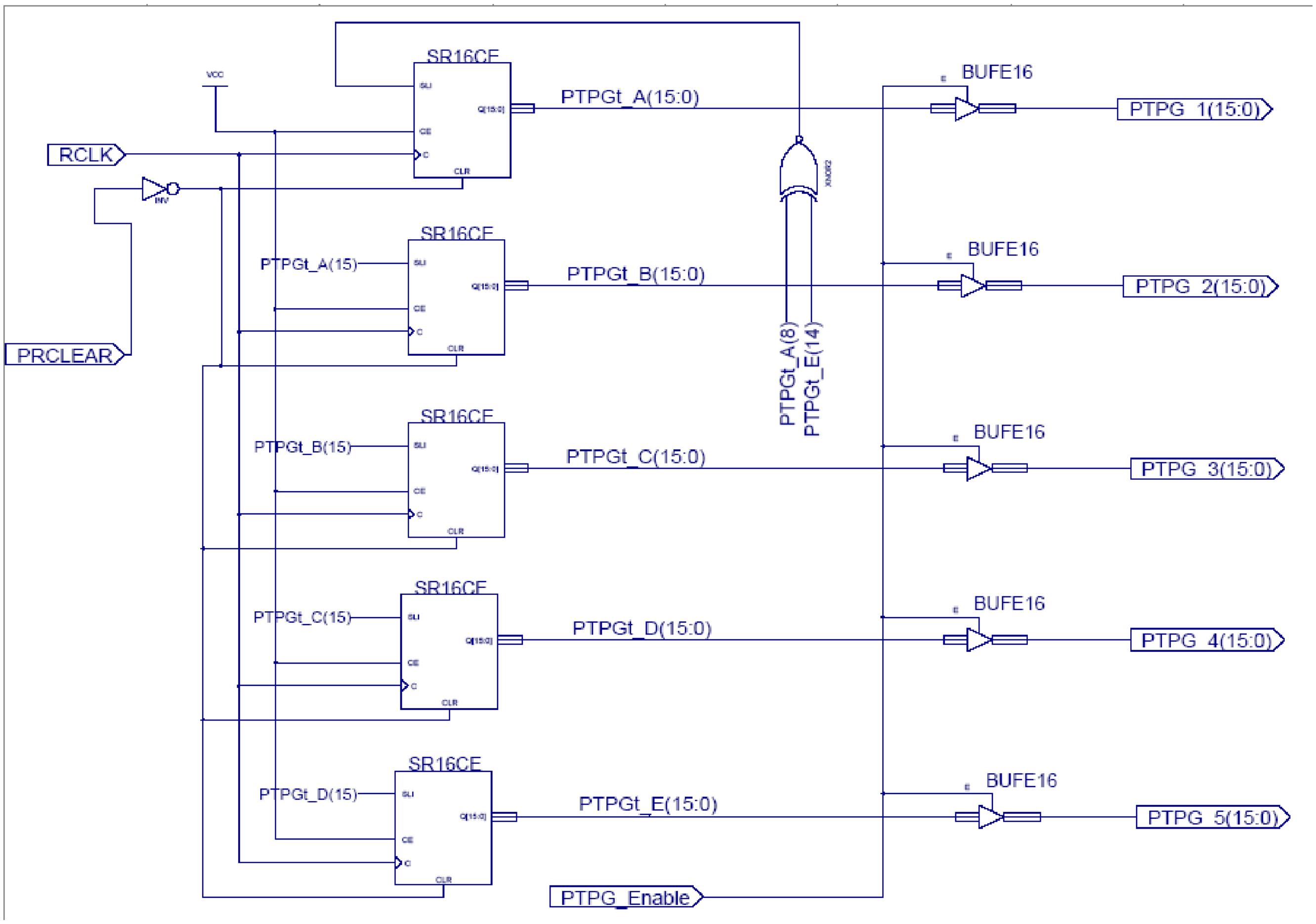


Figure (11): *Schematic diagram of the prtpg cell.*

3.3 Explanation of the design of the decoder_4_to_16 cell

Autonomous control signals, generated using the *timing_controller* cell, are utilized in the PRT mode. Other programmable control signals of the testing cycle in the DT mode are generated through the parallel port of the personal computer that is found on the back of your personal computer. It is accessed via 3 consecutive 8-bit ports in the processor's I/O space; *DATA Port*, *STATUS Port*, and *Control Port* [1].

The decoder (D4_16E) presented in Figure (12) accepts four input signals Control_PORT(3:0) from *Control Port* of the parallel port. The generated output signals from the decoder are the encoded signals that control the proper operation of the DTPG cell presented in section 3.4. The different delays of the signals of the control port generate glitches in the outputs of the decoder. To compensate that glitches, the Decoder_Enable signal disables the decoder in the time of the changing states of Control_PORT(3:0). This situation eliminates glitches and provides stable operation. Also, other encoded signals control two programmable control ports. The first one *Prog_Control_1(7:0)* is used to control the ATE system and the other *Prog_Control_2(7:0)* is used to the control the CUT.

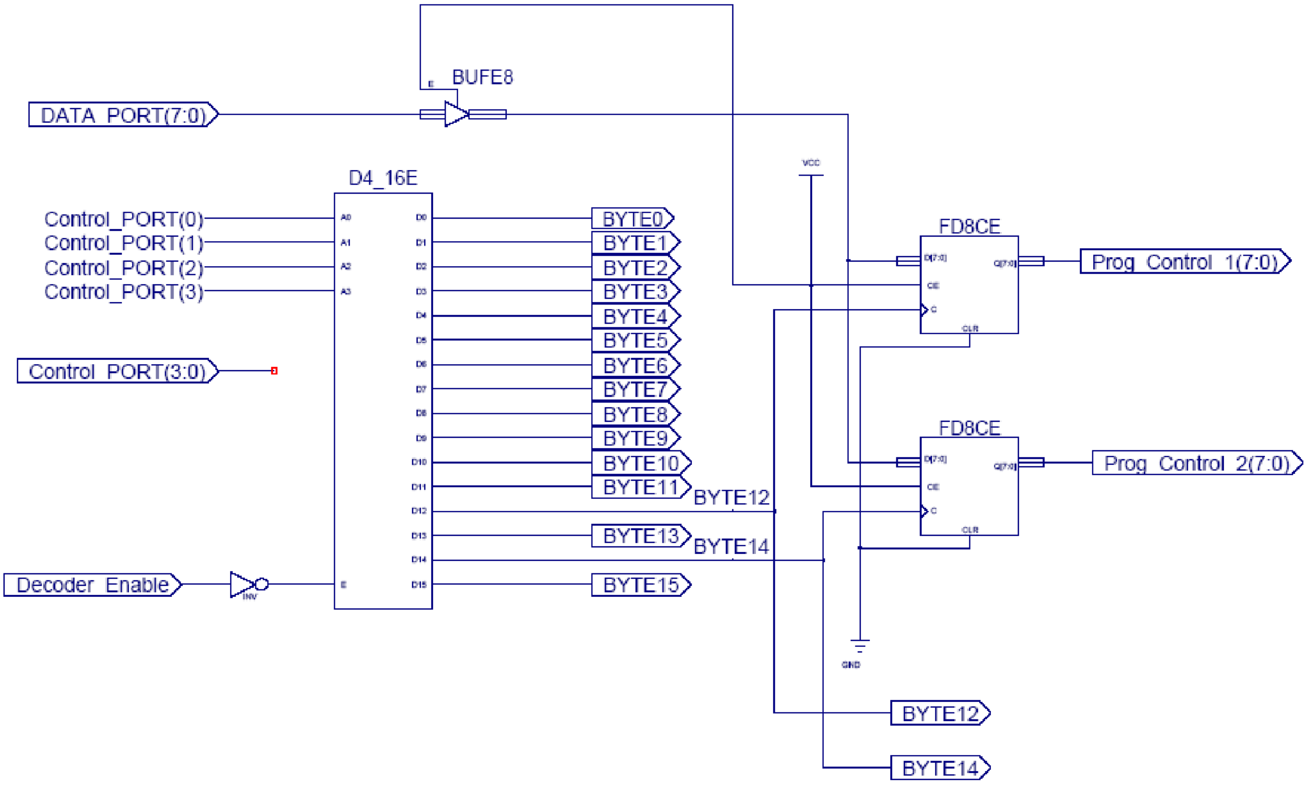


Figure (12): *Schematic diagram of the decoder_4_to_16 module.*

Figure (13) illustrates the timing diagram of the programmable control port *Prog_Control_1(7:0)*. The first bit of it is used to clear the *signature analyzer* and clear the Flip-Flops in the CUT in the DT mode, referred to DCLEAR. The next three bits are used to automatically transfer the signature to the personal computer as will be shown in section 3.5. The data port DATA_PORT(7:0) in this cell is used to write the command in the programmable control ports. Fig. 4 in section 3 illustrates that the signal DATA_PORT(7) which is the most significant bit disables the decoder in the time of the changing states of Control_PORT(3:0). Also, the data port DATA_PORT(7:0) is used to generate the deterministic test pattern vectors as will be shown in section 3.4.

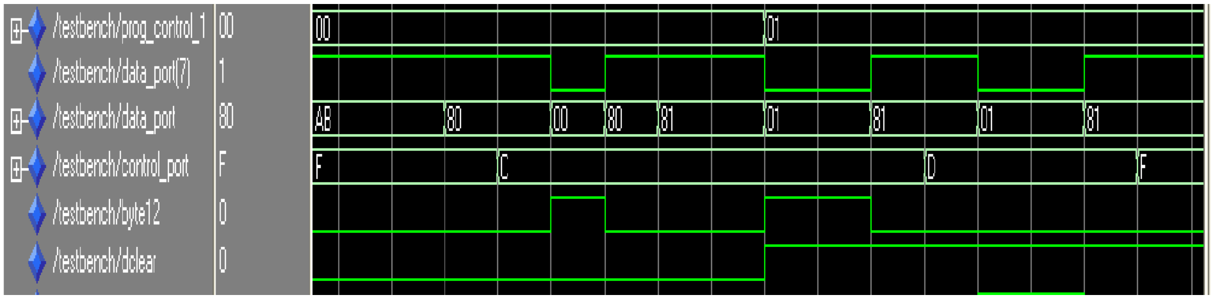


Figure (13): *Timing diagram of the testing operation in the programming mode.*

3.4 Explanation of the design of the dtpg cell

Some CUT inputs of the electronic cards needs specific logic state not pseudo-random binary signals. The TPG, based on deterministic testing, is a DTPG whose its outputs is 80 pins. It has arbitrary test length according to the test patterns calculated from algorithmic methods that detects different fault models of the CUT [9, 11-12]. In the DT mode, the test patterns of the DTPG are generated from the personal computer through the parallel port. The software part with the *decoder_4_to_16* cell is responsible to automatically transfer these test patterns to the CUT inputs. The start, stop, and all control signals of the test cycle in this mode are generated from the personal computer through the *decoder_4_to_16* cell. In this case, the number of the clocks inside the gate is variable depends on the required test patterns to deterministically exercise the CUT.

From Figure (12), signal BYTE13, called DCLK in Figure (14), is used to synchronize and control the sample rate of the probe input DATA and the output of the DTPG. So, the input DATA to the signature analyzer (SA) is processed every DCLK cycle within the gate interval as shown in Figure (15). The output of the DTPG is asserted in the rising edge of DCLK and the output of the SA is asserted in the falling edge. This provides the test pattern enough time to propagate through the ICs of the CUT before the acquisition of the DATA.

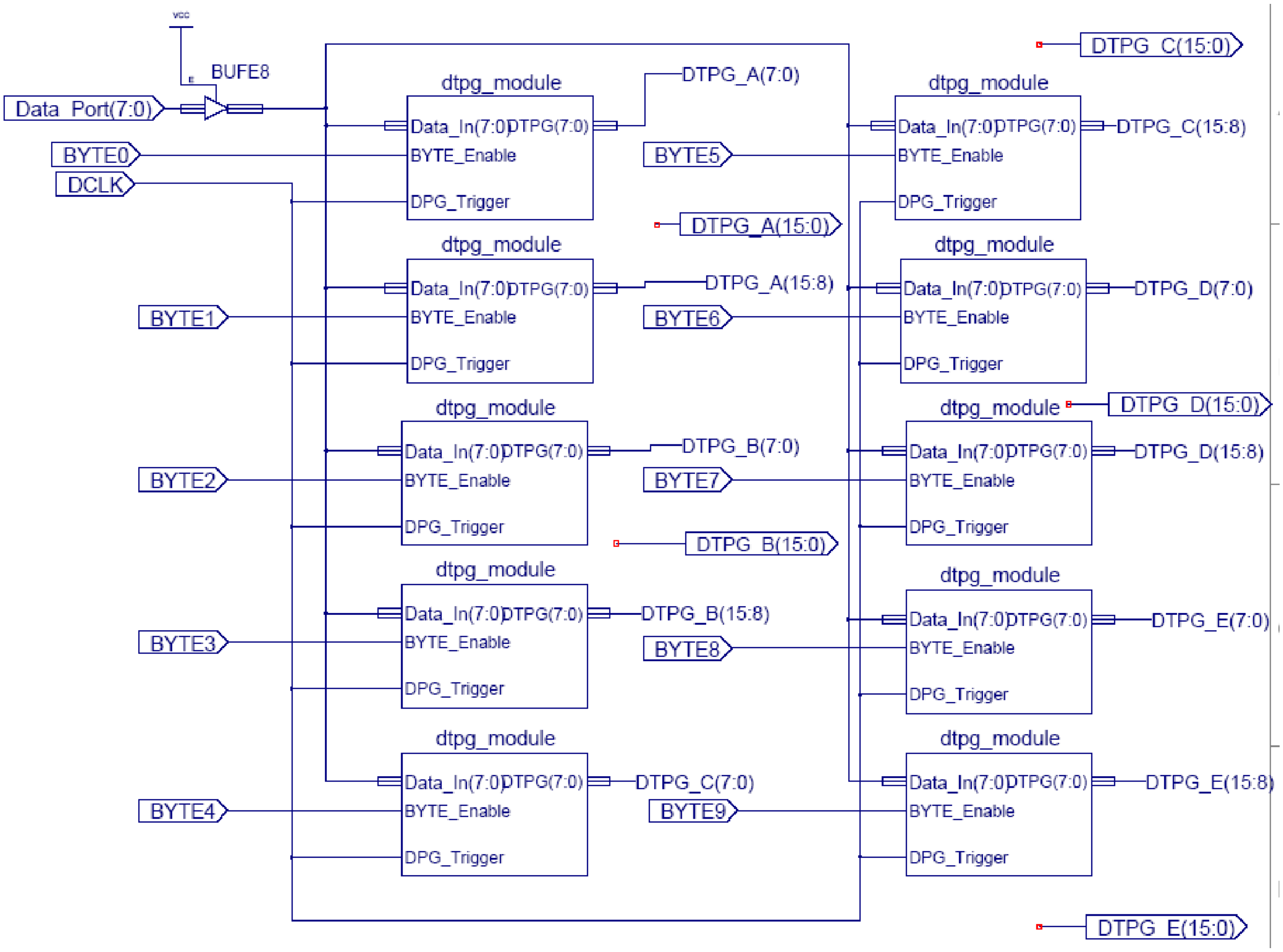
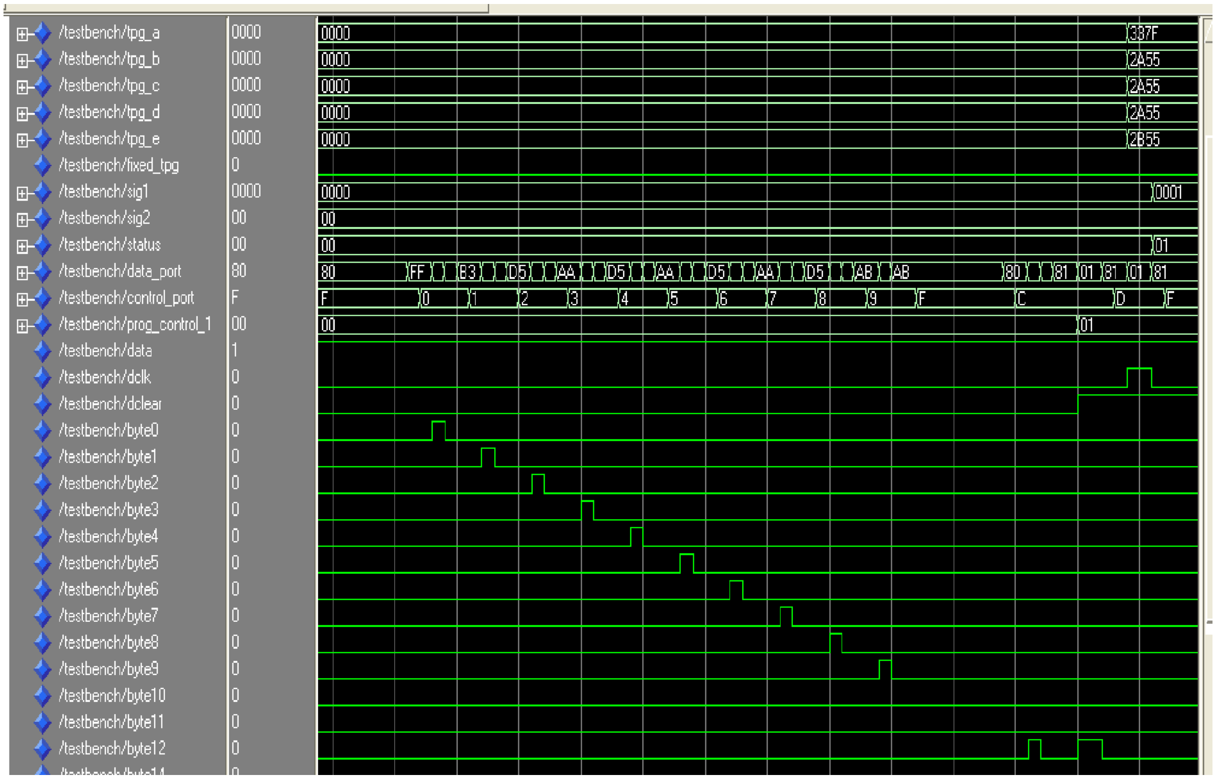


Figure (14): *Schematic diagram of the dtpg cell.*

In Figure (14), there are ten cells (*dtpg_module*). Each cell has two control signals. Control signal BYTE0 through BYTE9 control the latching Data_Port(7:0) in sequence. The DCLK triggers all these cells to transfer their contents to DTPG_A(15:0), DTPG_B(15:0), DTPG_C(15:0), DTPG_D(15:0), and DTPG_E(15:0) simultaneously as shown in Figure (15). This sequence is repeated each test pattern vector. The glitch free of the design interface can accurately control the processing of the testing cycle in this mode.

Some CUT inputs of the electronic cards needs fixed logic state and the other can be tested using the random binary signals. The hybrid between PRT mode and DT mode will be available through the good utilization of the programmable control port *Prog_Control_1(7:0)*. It divides the CUT inputs into two sets. One set of CUT inputs that needs fixed logic state is connected to single deterministic test pattern generated from DTPG and the other set is connected to the PRTPG. The test patterns applied to CUT inputs is the concatenation of the PRTPG and DTPG.



/testbench/tpg_a	0000
/testbench/tpg_b	0000
/testbench/tpg_c	0000
/testbench/tpg_d	0000
/testbench/tpg_e	0000
/testbench/fixed_tpg	0
/testbench/sig1	0000
/testbench/sig2	00
/testbench/status	00
/testbench/data_port	80
/testbench/control_port	F
/testbench/prog_control_1	00
/testbench/data	1
/testbench/dclk	0
/testbench/dclear	0
/testbench/byte0	0
/testbench/byte1	0
/testbench/byte2	0
/testbench/byte3	0
/testbench/byte4	0
/testbench/byte5	0
/testbench/byte6	0
/testbench/byte7	0
/testbench/byte8	0
/testbench/byte9	0
/testbench/byte10	0
/testbench/byte11	0
/testbench/byte12	0

0000 337F
0000 2A55
0000 2A55
0000 2A55
0000 2B55
0000 0001
00
00 01
80 FF B3 D5 AA D5 AA D5 AA D5 AB AB 80 81 01 81 01 81
F 0 1 2 3 4 5 6 7 8 9 F C D F
00 01

Figure (15): *Timing diagram of the testing operation in the DT mode.*

3.5 Explanation of the design of the signature_analyzer cell

The SA is based on the LFSR compactor circuit as shown in Fig. 1 (see section 1). From Fig. 16, the signature analyzer module has four parts. The first part is the 23-bit signature analyzer generator. The implementation of this analyzer is based on the primitive polynomial $1 + x^5 + x^{23}$ as shown in Figure (16) with the aliasing error probability = 2^{-23} = 0.000012 % [1]. Data compaction is achieved by probing a test node from which data asserts at the falling edge for each clock within the time period of the gate. Bits of sequence being measured are summed in modulo-2 with the LFSR feedbacks. Input sequence may be with different lengths but at the end of the measurement only the signature is the residue of the SA.

The second part is the selection part that switches between the required clock and reset signals in the PRT mode and the DT mode using the control signal *Testing_Mode*. *RCK* and *Sig_CLEAR* are used in the PRT mode when the control signal *Testing_Mode* is logic 0. *RCK* clocks the PRTPG at the rising edge clocks, and the SA generator at the falling edge. *Sig_CLEAR* clears the SA generator at the start of each gate interval as shown in the timing diagram of Figure (8). At the end of the gate interval, the signature is generated in the hexadecimal form. So, this operation repeats 5 times per second in the free-running mode and repeats once in the one-shot mode. The stability of the signature in the free-running mode assures the repeatability of the SA. In the DT mode, DCLK and DCLEAR are assigned. DCLK clocks the DTPG at the rising edge clocks and the SA at the falling edge as illustrated in Figure (15). DCLEAR generated from the first bit of the programmable control port *Prog_Control_1(7:0)* clears the SA at the start of each gate interval.

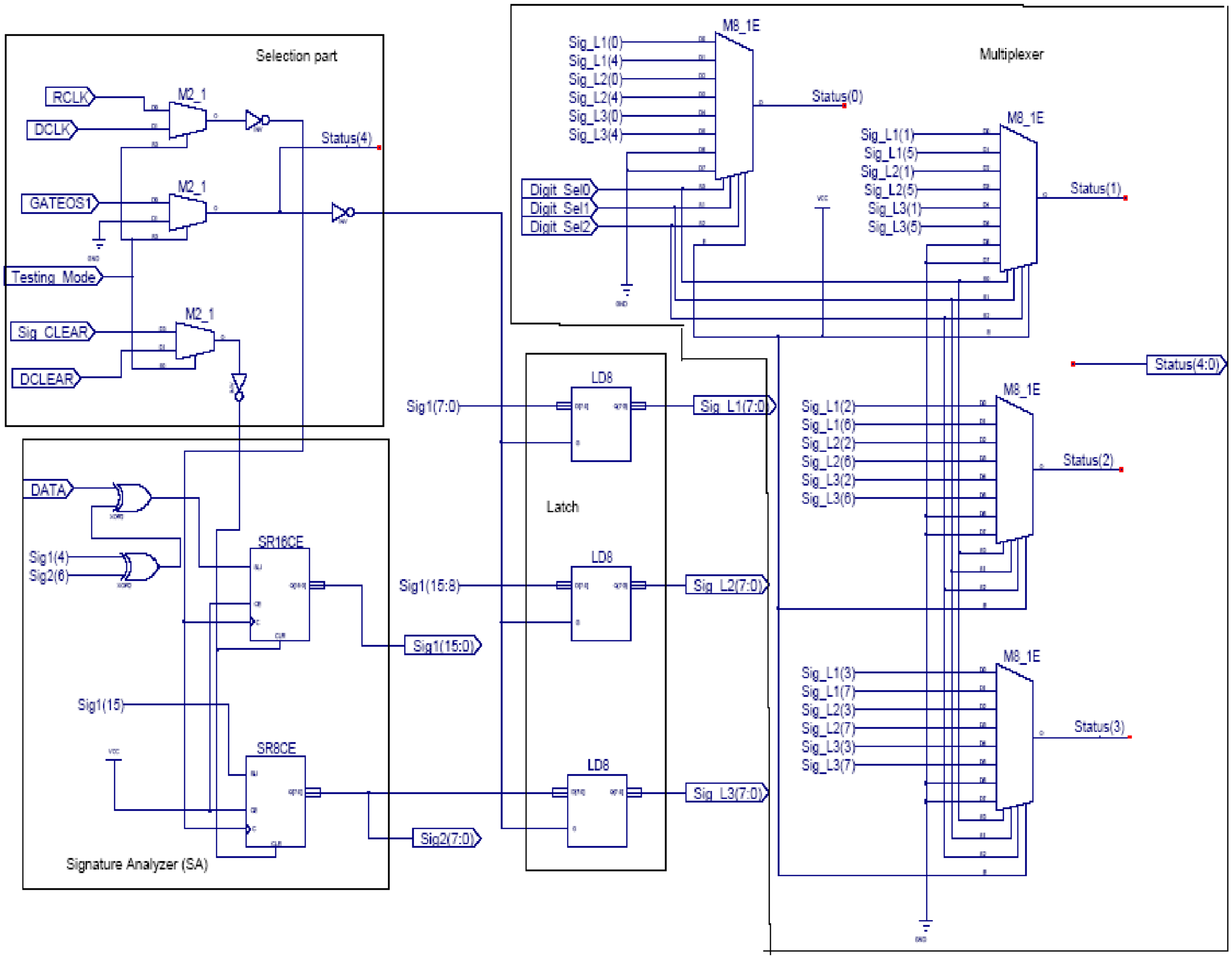


Figure (16): *Schematic diagram of the signature_analyzer cell.*

The third part is the latch. It latches the signature when the GATEOS1 is low. The fourth part is the multiplexer. The multiplexer is controlled by three control signals; *Digit_Sel0*, *Digit_Sel1*, and *Digit_Sel2*. These signals are controlled from the second bit to fourth bit of the programmable control port *Prog_Control_1(7:0)* as shown in Figure (4). Due to the limited number of the input signals of the status port of the parallel port of the computer, the signature is serially transferred digit by digit as shown in Figure (17). The software part of the system is used to receive this signature and store it in a file for fault detection and fault location processing.

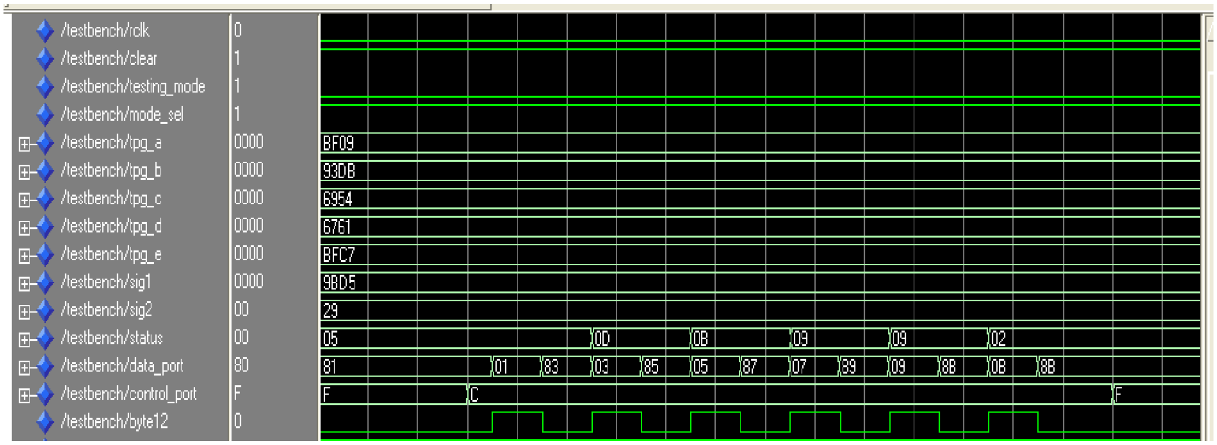

/testbench/rclk 0
/testbench/clear 1
/testbench/testing_mode 1
/testbench/mode_sel 1
/testbench/tpg_a 0000 BF09
/testbench/tpg_b 0000 93DB
/testbench/tpg_c 0000 6954
/testbench/tpg_d 0000 6761
/testbench/tpg_e 0000 BFC7
/testbench/sig1 0000 9BD5
/testbench/sig2 00 29
/testbench/status 00 05 0D 0B 09 09 02
/testbench/data_port 80 81 01 83 03 85 05 87 07 89 09 8B 0B 8B
/testbench/control_port F F C F
/testbench/byte12 0

Figure (17): *Timing diagram of the serial transfer of the signature.*

All the cells for this portable automatic testing system are connected altogether and implemented on FPGA chip Xilinx (X2S400EFT256-7). The timing simulation of the complete design is presented to verify proper operation. The timing diagrams represented here in this paper are based on the timing simulation. The report generated due to this implementation is given as follows.

Cell Usage:

# BELS	**494**		
# AND2	126	# OR2	107
# AND2b1	90	# VCC	2
# AND3	22	# XOR2	28
# AND3b1	16	# INV	33
# AND4b2	6	# LUT1	16
# AND5	1	# MUXF5_L	8
# AND5b1	4	# MUXF6	4
# AND5b2	6		
# AND5b3	4		
# AND5b4	1		
# BUF	1		
# GND	1		

# FlipFlops/Latches	**349**
# FD	84
# FDC	17
# FDCE	144
# LD	104
# Tri-States	**96**
# BUFE	96
# Clock Buffers	**3**
# BUFG	3
# IO Buffers	**174**
# IBUF	18
# IBUFG	2
# OBUF	154

# Logical	**15**
# NAND2	8
# NAND2b1	3
# NAND4	2
# NAND4b3	1
# XNOR2	1

Device utilization summary:

Selected Device: 2S400EFT256-7

Number of Slices:	204	out of	4800	4%
Number of Slice Flip Flops:	349	out of	9600	3%
Number of 4 input LUTs:	16	out of	9600	0%
Number of bonded IOBs:	174	out of	182	95%
Number of TBUFs:	96	out of	4800	2%
Number of GCLKs:	3	out of	4	75%

Total equivalent gate count for design: 133,097

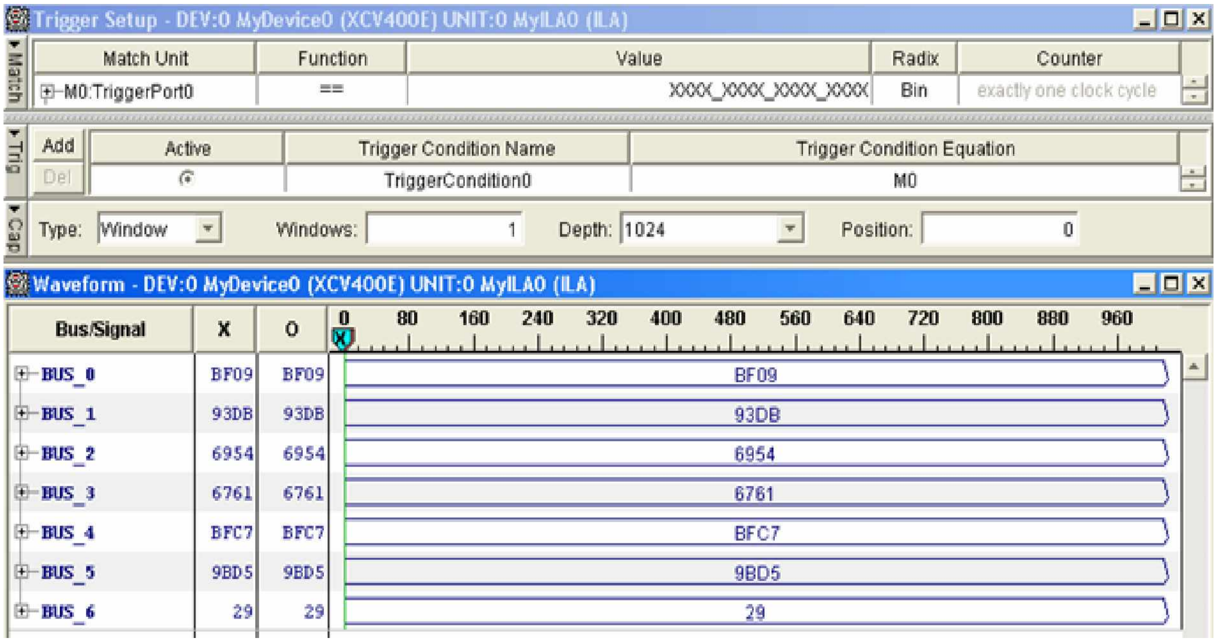
Timing Summary:

Minimum period: 14.774 ns (Maximum Frequency: 67.686 MHz)
Minimum input arrival time before clock: 4.477 ns
Maximum output required time after clock: 17.312 ns
Maximum combinational path delay: 17.191 ns

4. Experiments and results

In this section, the evaluation of the design performance on the FPGA chip is presented. In this paper, the hardware design implementation of the automatic testing system was presented with the FPGA chip Xilinx (X2S400EFT256-7). All designed cells are connected as shown in Fig. 4. The following timing diagrams represented in Figure (18) to Figure (21) are based on the real signals on the FPGA chip (X2S400EFT256-7).

Figure (18) represents the signals when the DATA input is HIGH. Figure (19) represents the outputs of the PRTPG asserted in the rising edge trigger of the clock and the outputs of the SA asserted in the falling edge trigger of the clock.



Trigger Setup - DEV:0 MyDevice0 (XCV400E) UNIT:0 MyILA0 (ILA)

Match Unit	Function	Value	Radix	Counter
M0:TriggerPort0	==	XXXX_XXXX_XXXX_XXXX	Bin	exactly one clock cycle

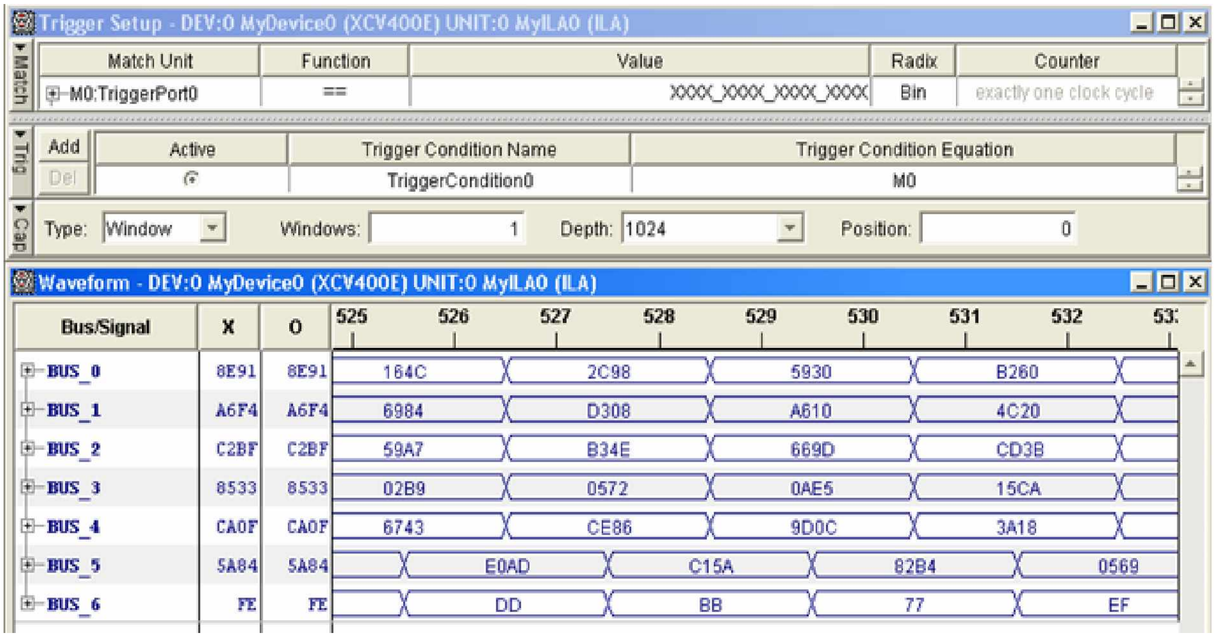
Active	Trigger Condition Name	Trigger Condition Equation
●	TriggerCondition0	M0

Type: Window Windows: 1 Depth: 1024 Position: 0

Waveform - DEV:0 MyDevice0 (XCV400E) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	
BUS_0	BF09	BF09	BF09
BUS_1	93DB	93DB	93DB
BUS_2	6954	6954	6954
BUS_3	6761	6761	6761
BUS_4	BFC7	BFC7	BFC7
BUS_5	9BD5	9BD5	9BD5
BUS_6	29	29	29

Figure (18): *The real signals on the FPGA chip when the DATA input is HIGH.*



Trigger Setup - DEV:0 MyDevice0 (XCV400E) UNIT:0 MyILA0 (ILA)

Match Unit	Function	Value	Radix	Counter
M0:TriggerPort0	==	XXXX_XXXX_XXXX_XXXX	Bin	exactly one clock cycle

Active	Trigger Condition Name	Trigger Condition Equation
○	TriggerCondition0	M0

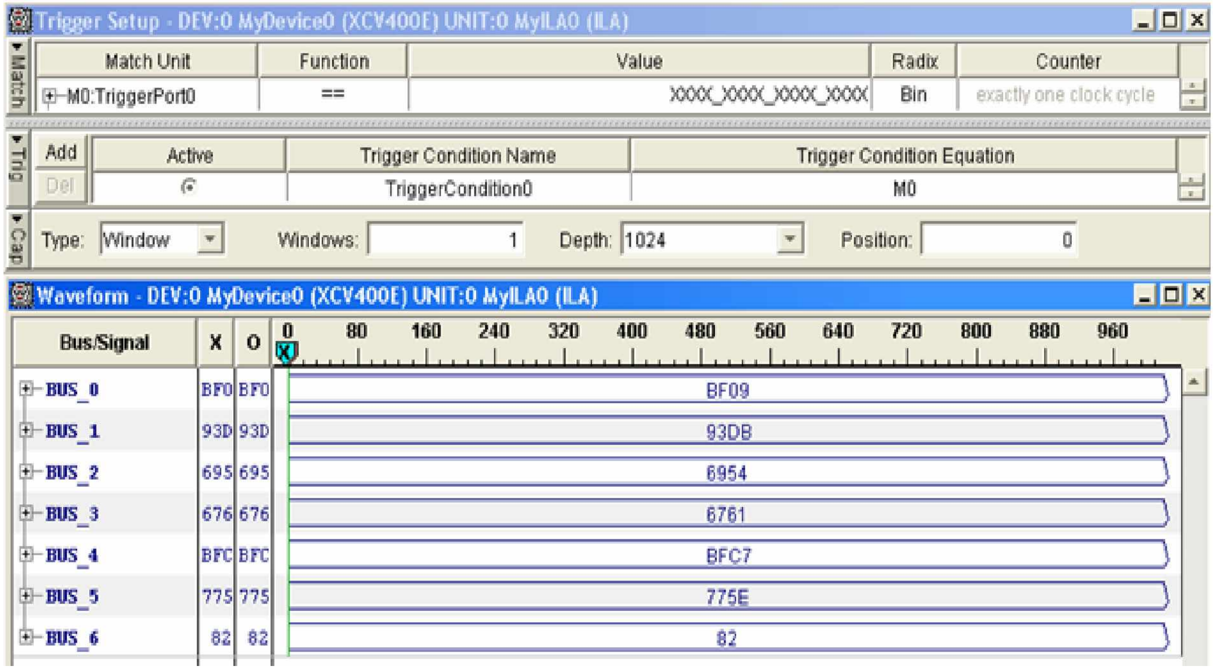
Type: Window Windows: 1 Depth: 1024 Position: 0

Waveform - DEV:0 MyDevice0 (XCV400E) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	Values (525–532)
BUS_0	8E91	8E91	164C, 2C98, 5930, B260
BUS_1	A6F4	A6F4	6984, D308, A610, 4C20
BUS_2	C2BF	C2BF	59A7, B34E, 669D, CD3B
BUS_3	8533	8533	02B9, 0572, 0AE5, 15CA
BUS_4	CA0F	CA0F	6743, CE86, 9D0C, 3A18
BUS_5	5A84	5A84	E0AD, C15A, 82B4, 0569
BUS_6	FE	FE	DD, BB, 77, EF

Figure (19): *The real signals on the FPGA chip represent the outputs of the PRTPG and the outputs of the SA.*

Figure (20) represents the signals when the DATA input is connected to the TPG_A(0). Figure (21) represents the signals when the DATA input is connected to the TPG_E(15). All these results are the same results of the system designed in paper [1] and all results presented in paper [1] achieved in this portable ATE.



Trigger Setup - DEV:0 MyDevice0 (XCV400E) UNIT:0 MyILA0 (ILA)

Match Unit	Function	Value	Radix	Counter
M0:TriggerPort0	==	XXXX_XXXX_XXXX_XXXX	Bin	exactly one clock cycle

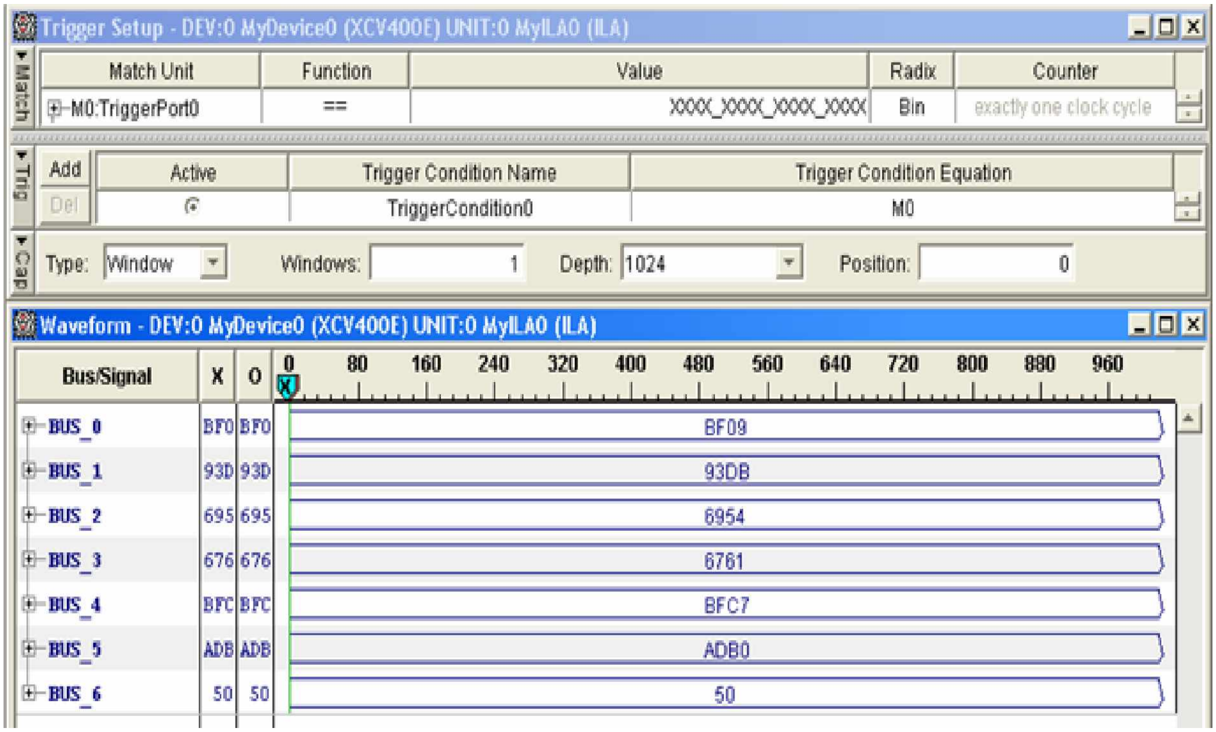
Active	Trigger Condition Name	Trigger Condition Equation
○	TriggerCondition0	M0

Type: Window Windows: 1 Depth: 1024 Position: 0

Waveform - DEV:0 MyDevice0 (XCV400E) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	Value (0–960)
BUS_0	BF0	BF0	BF09
BUS_1	93D	93D	93DB
BUS_2	695	695	6954
BUS_3	676	676	6761
BUS_4	BFC	BFC	BFC7
BUS_5	775	775	775E
BUS_6	82	82	82

Figure (20): *The real signals on the FPGA chip when the DATA input is connected to the TPG_A(0).*



Trigger Setup - DEV:0 MyDevice0 (XCV400E) UNIT:0 MyILA0 (ILA)

Match Unit	Function	Value	Radix	Counter
M0:TriggerPort0	==	XXXX_XXXX_XXXX_XXXX	Bin	exactly one clock cycle

Active	Trigger Condition Name	Trigger Condition Equation
●	TriggerCondition0	M0

Type: Window   Windows: 1   Depth: 1024   Position: 0

Waveform - DEV:0 MyDevice0 (XCV400E) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	0 … 960
BUS_0	BF0	BF0	BF09
BUS_1	93D	93D	93DB
BUS_2	695	695	6954
BUS_3	676	676	6761
BUS_4	BFC	BFC	BFC7
BUS_5	ADB	ADB	ADB0
BUS_6	50	50	50

Figure (21): *The real signals on the FPGA chip when the DATA input is connected to the TPG_E(15).*

Note: BUS_0, BUS1, BUS_2, BUS_3, and BUS_4 in Fig. 18 through Fig. 21 are referred to TPG_A(15:0), TPG_B(15:0), TPG_C(15:0), TPG_D(15:0), and TPG_E(15:0), respectively. Also, BUS_5, and BUS_6 are referred to Sig1(15:0), and Sig2(7:0), respectively.

5. Conclusions:

In this paper, the new portable automatic testing approach for digital ICs is implemented using FPGA technology. The schematic diagrams presented here were used to design the portable complete testable design. This system is capable of performing the same tasks of the ATE system presented in [1] in fault detection and source fault location.

The hardware experiment results were compared with the simulation results to verify the design performance. The measured signatures in the new design are compared with the measured signatures in [1]. The fault detection and location are identically achieved. The system is designed to act as a testing tool for different digital circuit cards. This system allows the trouble-shooting of digital circuits in the different fields. This approach is considered to reduce the cost of the traditional ATE. So, our objective is to go step in the direction of the portable ATE.

References:

[1] 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.

[2] Mohamed H. El-Mahlawy, *Automatic Measurment of Digital Circuits*, M.Sc. degree, MTC, Cairo, 1995.

[3] A. Cstover, *Automatic Test Equipment*, McGraw-Hill inc., 1984.

[4] A. Seddik, *Microcomputer Based Automatic Testing System through Signature Analysis*, M.Sc. Degree, MTC, Cairo, 1992.

[5] M. El Said Gohniemy, S. Fadel Bahgat, Mohamed H. El-Mahlawy, and E. E. M. Zouelfoukkar, *A Novel Microcomputer Based Digital Automatic Testing Equipment using Signature Analysis*, IEEE conference on Industrial Applications in Power Systems, Computer Science and Telecommunications, Bari, Italy, 13-16 May 1996.

[6] Mohamed H. El-Mahlawy, *Pseudo-Exhaustive Built-In Self-Test for Boundary Scan*, Ph.D. thesis, Kent University, U.K., 2000.

[7] Sherif Anas Mohamed, *In-Circuit Testing for Electronic Board*, M.Sc. degree, MTC, Cairo, 2006.

[8] Paul H. Bardell, Willian H. McAnney, Jacob Savir, *Built-In test for VLSI: pseudorandom techniques*, John Wiley and Sons, 1987.

[9] Miron Abramovici, Melvin A. Breuer, Arthur D. Friedman, *Digital systems testing and testable design*, IEEE PRESS the Institute of Electrical and Electronics Engineers, Inc., New York, 1990.

[10] Thomas W. Williams, Wilfried Daehn, Matthias gruetzner, and Cordt W. Starke, *Bounds and analysis of aliasing errors in linear feedback shift register*, IEEE transactions on computer-aided design, Vol. 7, NO. 1, pp 75-83, Jan. 1988.

[11] Rochit Rajsuman, *Digital hardware testing: Transistor-level fault modelling and testing*, Artech House, 1992.

[12] Parag K. Lala, *Digital circuit testing and testability*, Academic Press, 1997.