Military Technical College
Kobry El-kobbah,
Cairo, Egypt

5th International Conference
on Electrical Engineering
ICEENG 2006

NEW AUTOMATIC TESTING ARCHITECTURE FOR INTEGRATED CIRCUITS

Sherif Anas*, Mohamed H. El-Mahlawy*, Ehab A. El-Sehely*, and Al-Emam S. Ragab*

ABSTRACT

In this paper, a complete example for BIST (Built-In Self-Test) boundary scan architecture and 16-bit multiplier as the CUT is presented. Adding BIST boundary scan capabilities to the digital VLSI integrated circuit design makes the electronic card testable from five pins TMS, TCK, TDI, TDO and TRST* that is optional. The simulation and then design download are presented on the Spartan Xilinx X2C100 chip. The hardware implementation is tested using the interfacing through the parallel port of the personal computer that supplies required five control pins. This approach will lead to the concept of the portable ATE (Automatic Test Equipment). All required test circuitry is embedded in the integrated circuits and the control of the test circuitry is supplied from the TAP (Test Access Port) controller. Finally, the TAP controller is controlled from the parallel port of the personal computer. So, the personal computer is used as a master controller and the TAP controller is used as a slave controller. The presented idea of the new BIST testing architecture solves the testing problem of the digital VLSI circuits using the traditional ATE.

KEYWORDS: Design-for-test (DFT), Automatic Test Equipment, Testing of electronic circuits.

1. INTRODUCTION

Advances in VLSI technology have led to the fabrication of chips that contain a very large number of transistors, integrated on a single chip. So, printed circuit boards (PCBs) become more complex. The need for testing becomes more important. In the 70's and 80's the test systems consist of "bed of nails" and external test probes. The component technology of that era was dual in line and the PCBs had maximum two layers. It was easy to access all the soldering points. The nails hit various points of the PCBs and measured the different values.
As electronic production is getting more and more complicated, one single chip can contain more than 10 millions transistors and over 500 inputs/outputs on a very small area. The technology of the multi-layer production for printed circuit board is available today. To check the functionality of the components, several self-testing techniques are developed [1-13]. The traditional In-Circuit tester can not be used due to the lack of the accessibility of the nodes. Boundary Scan IEEE 1149.1 standard, which is a design for testability technique, describes a special method for test of digital components and electronic boards [14]. It offers a convenient alternative to physical probing by effectively migrating the test probe circuitry into the chip. It enables a non-contact method of accessing chip pins for testing.

---

*Egyptian Armed Forces

In boundary scan technique, the virtual probing increases the controllability and observability. Using an appropriate set of boundary scan test patterns, most of the interconnections on a circuit board can be checked for the continuity and short circuits [14].

The IEEE Standard 1149.1 defines an interface for boundary scan testing at device and assembled circuit board level. It defines four (or optionally, five) new pins; through which the boundary scan test features are controlled, forming the test access port (TAP). Two of them (test clock, TCK, and test mode select, TMS) are used to control the protocol, while the remaining two pins (test data input TDI and test data output TDO) are employed to shift data into and out of the chip serially. The optional pin (test reset input TRST*) forces the test logic into its reset state asynchronously to ensure normal system operation at power up when the test logic is not in use.

Due to the cost of testing VLSI devices, the incorporation of BIST capability inside a chip is increasingly desirable. The incorporation of BIST capabilities into the boundary scan (BS) architecture, defined in the IEEE-1149.1 standard has been investigated in [11-13]. New boundary scan architecture for BIST was presented. The Boundary Scan Register (BSR) input cells have been configured to operate as a Test Pattern Generator (TPG) in the BIST mode [15-17]. The BSR output cells have been configured to operate as a Muti-Input Shift Register (MISR) in the BIST mode [17]. This configuration supports BIST for either the cascaded or non-cascaded BSR input and output cells. The Test Access Port Controller (TAPC) controls the BIST process. Instructions for BIST process were proposed. This configuration supports the BIST for both the Built-In Logic Block Observer (BILBO) register [1, 18] and the register transfer level.

BIST requires hardware overhead to incorporate a TPG, a test response compactor, and a BIST controller into the system (core) logic. For the register transfer level, the insertion of segmentation cells in the case of pseudo-exhaustive testing [19-20] may affect the system timing due to the unequal sequential depth in the BIST mode, so it is required to insert delay flip-flops which add significant area overhead and degrade circuit performance. Also, transferring every flip-flop into BIST flip-flop adds area overhead and degrades circuit performance in the normal mode of the chip. To compensate these problems, two design solutions were presented to convert the problem into one combinational block (combinational equivalent) [13]. The incorporation of BIST capabilities into the boundary scan architecture, defined in the IEEE-1149.1 standard, with these two solutions was presented in [13].

In this paper, a new testing approach for BIST is presented, implemented on FPGA Spartan Xilinx X2C100. The concept of the portable ATE is introduced to reduce the complexity of the traditional ATE. A 16-bit Parallel Pipelined Multiplier, with BIST using Boundary Scan, is developed as an example to demonstrate the new testing approach.

We will begin with the concept of the presented testing architecture in section 2. Design and implementation of a testable 16-bit parallel pipelined multiplier will be presented in section 3. The experimental results will be discussed in section 5 and the conclusion will be presented in section 6.

2. THE CONCEPT OF THE PRESENTED TESTING ARCHITECTURE

The complete FPGA design flow is an iterative process of entering; implementing and verifying the design until it is correct and complete [21]. Fig. 1 illustrates the testing design on chip. It is composed of personal computer and a protocol interface with software package to perform the required tests.

'The personal computer is used as ATE controller to control the testing operation where its parallel port is used to apply different digital test vectors to a FPGA chip on the download card. The computer is used as a signal generator and a signal capture. The simulated test vector signal is applied directly to the FPGA chip after building a computer interface to handle this process. The data is applied in serial format. The serial test data output is pulled by the parallel port and compared with the pre-stored data on the personal computer in order to check its validity. The parallel port originally has three ports, data, control, and status port. These ports consist of separately transmitter and receiver unit. The transmitter is an output 8-bit port (data port) and an output 4-bit port (control port). The receiver is an input 5-bit port (status port). Controlling the data lines of the parallel port will be accomplished by using C programs that will be executed as test program. A protocol interface with C program is built on the computer to handle the serial input data and control signals from the personal computer, and the serial output data from the FPGA design to PC. The measured results are investigated and compared with the previous simulated one.

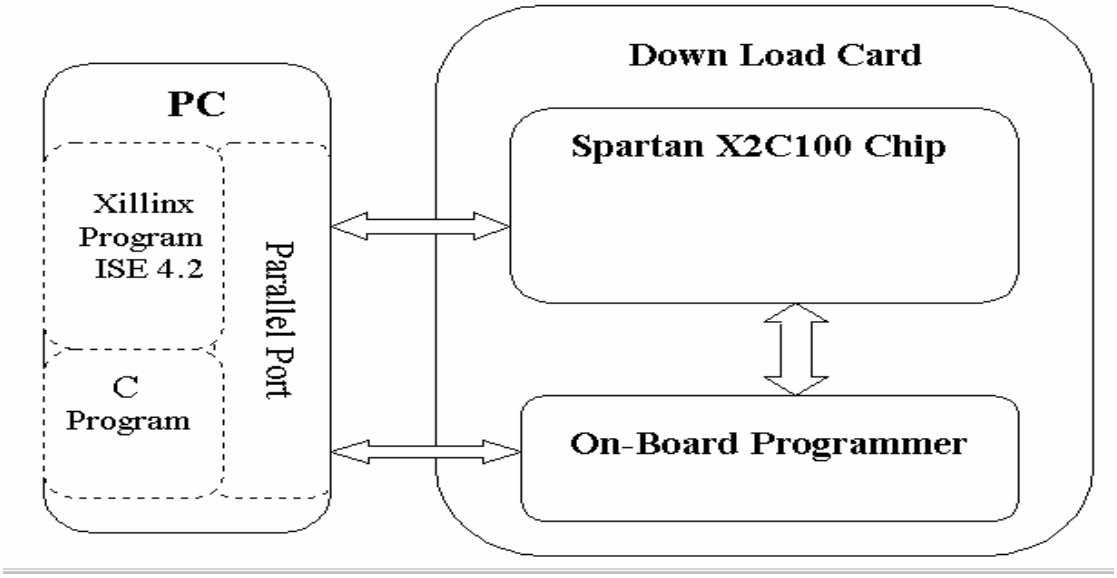


Fig. 1. Block diagram of the testing design on chip.

In this section, we will discuss how the design on the chip is tested as a real time application. An interface between the computer and the hardware design is built for that purpose. Design interface through parallel port of the personal computer enables the required control signals; TRST, TMS, TCK, TDI and Chip_CLK to combine all the transitions as a digital vector input to the design in decimal format in a text file. Then, the computer is used to apply these signals to the design on chip and to capture the data results (TDO) from the design on chip using the interface circuitry. The measured results are investigated using the computer to compare with the expected results. This interface consists of two parts hardware and software parts.

2.1 Hardware Interface

The computer generates the digital test vector using its output port H378 (D0 to D4 of the data port). This test vector is composed of TRST, TMS, TCK, TDI, and Chip_CLK. TDO is captured by input port H379 (bit 5 of the status port). Fig. 2 illustrates the hardware interface part and Fig. 3 illustrates the block diagram of the hardware interface part.

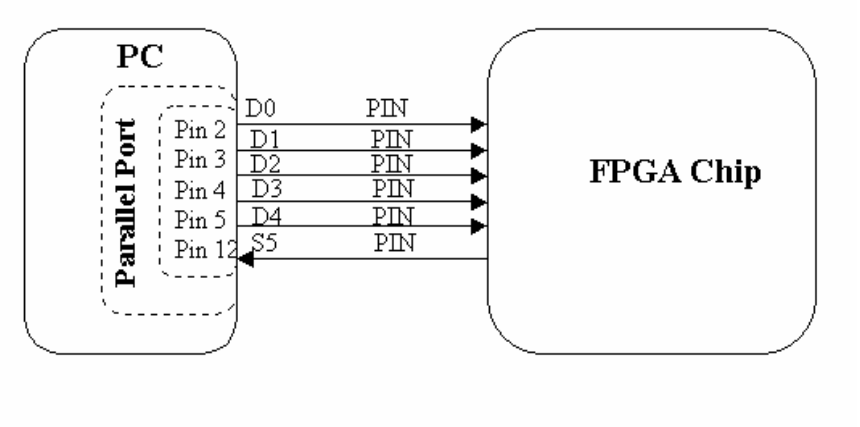


Fig. 2. Hardware interface part.

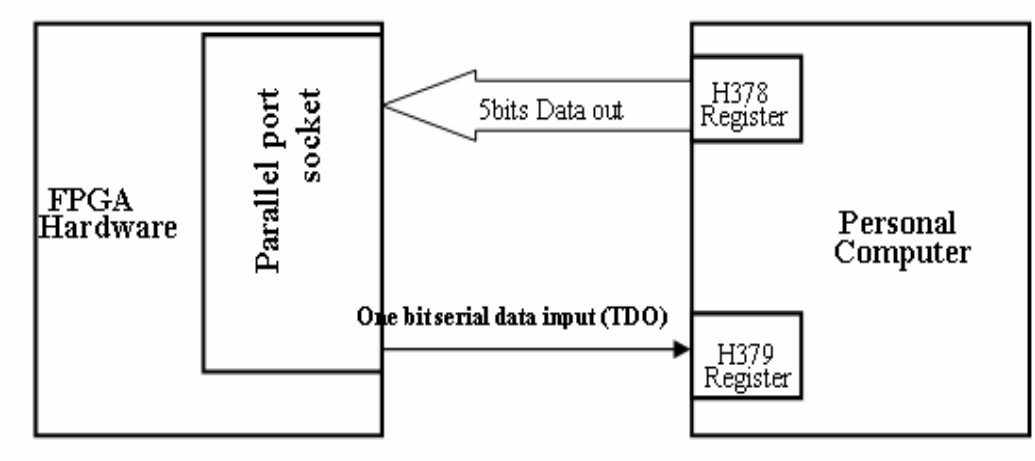


Fig. 3. Block diagram of the hardware interface part.

2.2 Software Part of the Interface

The software part of the interface is based on a software program (C program) for the communication between the design on chip Spartan Xilinx chip (X2C100) and the parallel port of the personal computer. The control signals; TRST, TMS, TCK, TDI and Chip_CLK are combined and recorded all the transitions as a digital test vector input to the design in decimal format. These vectors are inserted in a text file. A simple example will illustrate the idea as shown in Fig. 4 and Table 1. Table 1 illustrates the input signals (TRST, TMS, TCK, TDI) as a binary format from Fig. 4. The output results of TDO are extracted from the timing simulation and stored in a file as expected results to be compared with the measured output of the chip.

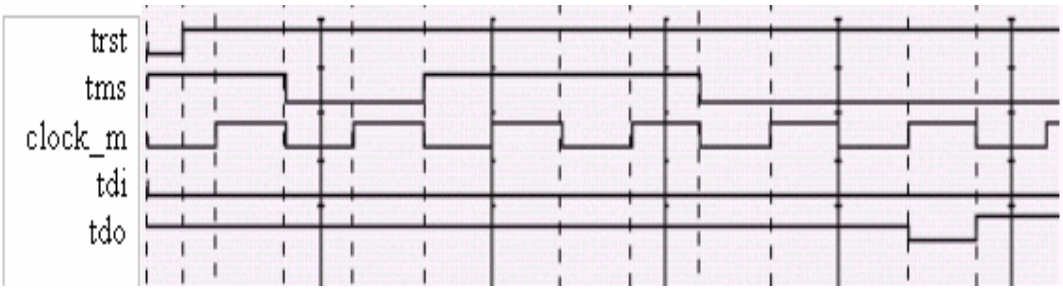


Fig. 4. A simple example extracted from timing simulation.

Table 1. Binary format of the digital test vector.

n	*Input*				*Value in decimal*	*Output*
	TRST	*TMS*	*T_CLK*	*TDI*		*TDO*
1	0	1	0	0	2	z
2	1	1	0	0	3	z
3	1	1	1	0	7	z
4	1	0	0	0	1	z
5	1	0	1	0	5	z
6	1	1	0	0	3	z
7	1	1	1	0	7	z
8	1	1	0	0	3	z
9	1	1	1	0	7	z
10	1	0	0	0	1	z
11	1	0	1	0	5	z
12	1	0	0	0	1	0
13	1	0	1	0	5	1

The program (written with C language) reads the digital test vector and converts it to the binary form. The binary form of the digital test vector is transferred to data port H378. The response from the chip will be extracted through status port H379. The actual binary value of TDO will be deduced. The binary value of TDO is stored in measured file. The measured values are compared with the expected one. The results are displayed PASS if they are identical or FAIL when they are different. The processing is repeated according to the number of digital test vectors.

3. DESIGN AND IMPLEMENTATION OF A TESTABLE 16-BIT PARALLEL PIPELINED MULTIPLIER

This section presents a complete design of a boundary scan testable circuit. The system (core) logic implements 16-bit parallel pipelined multiplier. The complete testable 16-bit parallel pipelined multiplier with BIST boundary scan architecture is presented in section 3.2.

3.1 Design and Implementation of 16-Bit Parallel Pipelined Multiplier

The basic form of multiplication consists of forming the product of two positive binary numbers. The valid result for the multiplication operation will be available after sixteen clocks. The timing simulation of the complete design, shown in Fig. 5, illustrates the proper operation under applying the input stimulus. The hexadecimal input 1D1C multiplies the hexadecimal input 009C that equals 0011BD10 after sixteen clocks.

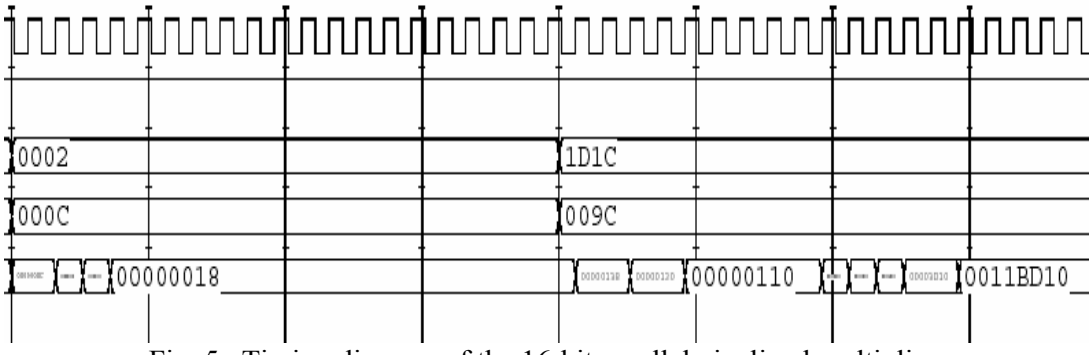


Fig. 5. Timing diagram of the 16-bit parallel pipelined multiplier.

Fig. 6 illustrates the block diagram of 16-bit parallel pipelined multiplier. The block *16-bit* is used as one clock shift (register). The system (core) logic implements 16-bit parallel pipelined multiplier. M-16x1 consists of two 16-bit register cell, one 16x1 multiplier cell, and 16-bit adder cell. All these cells are connected altogether and implemented on Spartan Xilinx (X2C100).

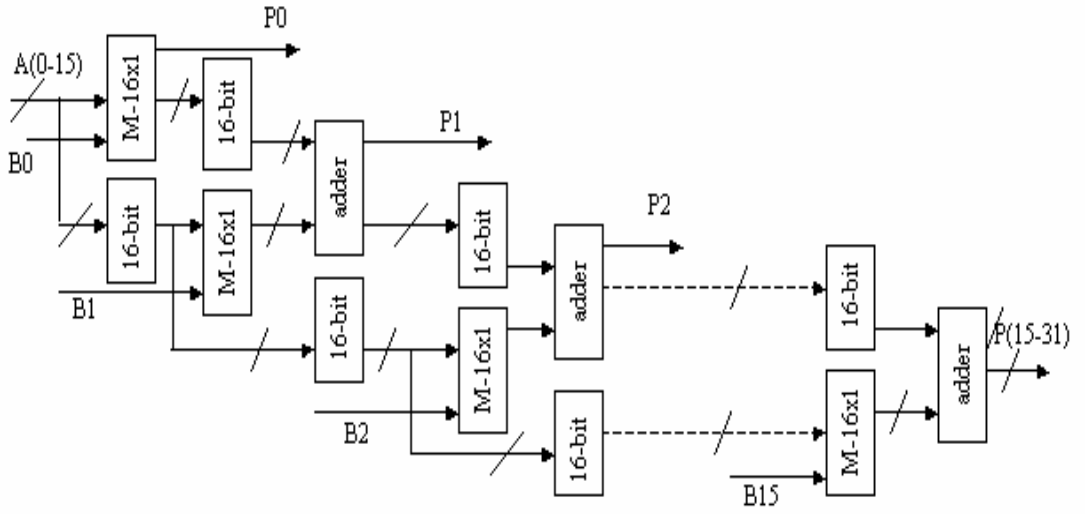


Fig. 6. Block diagram of 16-bit parallel pipelined multiplier.

The report generated due to this implementation is given as follows:

Minimum period: 36.418ns (Maximum Frequency: 27.459MHz)
Number of Slices: 758 out of 1,200 63%
Number of Slices containing unrelated logic: 0 out of 758 0%
Number of Slice Flip Flops: 79 out of 2,400 19%
Number of 4 input LUTs: 1,240 out of 2,400 51%
Number of bonded IOBs: 65 out of 140 46%
Number of GCLKs: 1 out of 4 25%
Number of GCLKIOBs: 1 out of 4 25%
Total equivalent gate count for design: 11,272

3.2 Design and Implementation of a Testable 16-Bit Parallel Pipelined Multiplier

The 16-bit parallel pipelined multiplier is chosen to verify test circuitry. We will start with the discussion of the test circuitry and then the testable 16-bit parallel pipelined multiplier.

The test circuitry added to the multiplier here is based on the design of [13]. The test pattern, applied to the combinational equivalence, is held for *d* clock cycles, where *d* is the maximum sequential depth of the circuit [12-13]. The method in [1, 8, and 10] to test the combinational circuits pseudo-exhaustively can also be applied to this type of the sequential circuit without timing violation and less hardware overhead. In [13], a design solution to support these types of circuits was presented. The boundary scan cells are re-configured in the BIST mode. BIST BSR (Build-In Self-test for boundary scan) input cells are configured as a TPG and BIST BSR (Build-In Self-test for boundary scan) output cells are configured as an MISR (Multi-Input Shift Register) [11]. All boundary scan cells are clocked by *TCK* (Test Clock). All registers in the system (core) logic are clocked by *Chip_CK* in the normal mode and by *TCK* in the test mode. *Chip_CK* and *TCK* are asynchronous clocks. Fig. 7 illustrates the clock distribution. A programmable control unit, PCU, (Fig. 8) had been designed which enables the TPG and the MISR, to hold each test pattern for at least *d* clock cycles, where *d* is the

maximum sequential depth. It enables the architecture to convert the problem of sequential circuits into one combinational block.

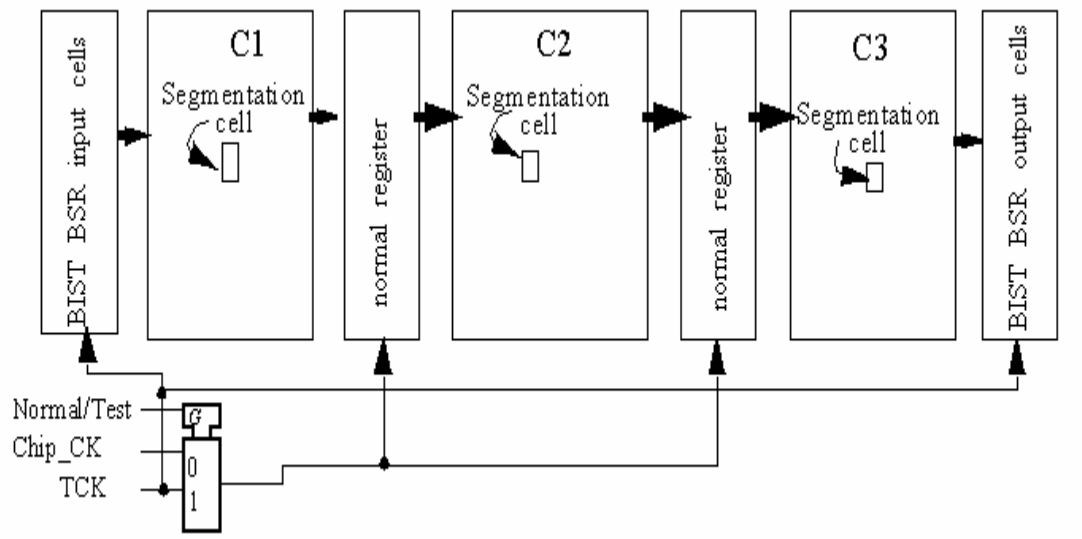


Fig. 7. Clock distribution.

The control of the PCU is based on *SYNC* instruction (user-defined instruction [1, 8]). It sets control signal *Enable_Sync* to high and switches between the chip clock and the TCK. The address field of the *SYNC* instruction is loaded to the programmable counter, *PC*, at inputs, *P0, P1, P2, P3* when both control signals *Enable_Sync*, and *Run-Test-Idle* go high. (Control signal *Run-Test-Idle* is active high in the Run-Test/Idle state [1, 8].) These inputs represent the equivalent binary form of *d*. For example, for $d = 3$, the binary form is 0010 ($d - 1 = 2$), and *P3, P2, P1, P0* is 1101. The carry signal, generated from *PC*, will be high for a clock cycle every *d* clock cycles. The *PC* is enabled when control signal *BIST_mode* goes high. The control signals *BIST_mode_O* and *BIST_mode_I* , which feed the BIST BSR output and input cells, respectively, and *HOLD_BILBO_in* which feeds the BILBO registers, are shown in Fig. 8 and Fig. 9 ($d = 16$). The control signals *DR_CapShf* and *DR_Shf*, which feeds all BIST BSR cells, are also shown in Fig. 8. These control signals will enable the BIST BSR input/output cells and the BILBO registers to hold for *d-1* clock cycles in the BIST mode. Control signal *Enable_Sync* control multiplexer *M1*. For normal operation and before loading the *SYNC* instruction, the Chip_CK feeds the system logic. When the *SYNC* instruction is loaded in the test mode, *Enable_Sync* goes high on the falling edge of *TCK*. The *SyEnable flag* is set on the following rising edge of TCK, the *DivRun flag* is set on the following falling edge of TCK. Multiplexer *M1* is switched to select its second channel (*TCK*), which feeds the system logic.

Using programmable control unit (PCU) for register transfer level, the following steps will summarize the BIST process:

1. Initialize the FSM of the TAPC to the Test-Logic-Reset state and load the IR (Instruction Register) with the PRELOAD instruction.
2. Shift the initial seed into the BIST BSR cells.
3. Load the IR with the SYNC instruction. Set *d* to 16 (*P3 P2 P1 P0* = 0000). Now, all registers in the system logic are clocked by TCK.
4. Load the IR with the BIST-BSR instruction. The IR may be loaded with the BIST-BSR instruction again to enable the propagation of the response of the initial seed through the normal registers to the inputs of the BIST BSR output cells.

5. Go to the Run-Test/Idle state. In this state, the control signals *BIST_mode_O*, *BIST_mode_I*, *HOLD_BILBO_in*, *DR_CapShf*, and *DR_Shf,* generated from PCU, will be as shown in Fig. 9. Stay in this state for the required number of clock cycles.

6. When the FSM leaves this state, the signature, generated in BIST BSR cells, needs to be shifted out.

7. Go to the Test-Logic-Reset state and halt the test.

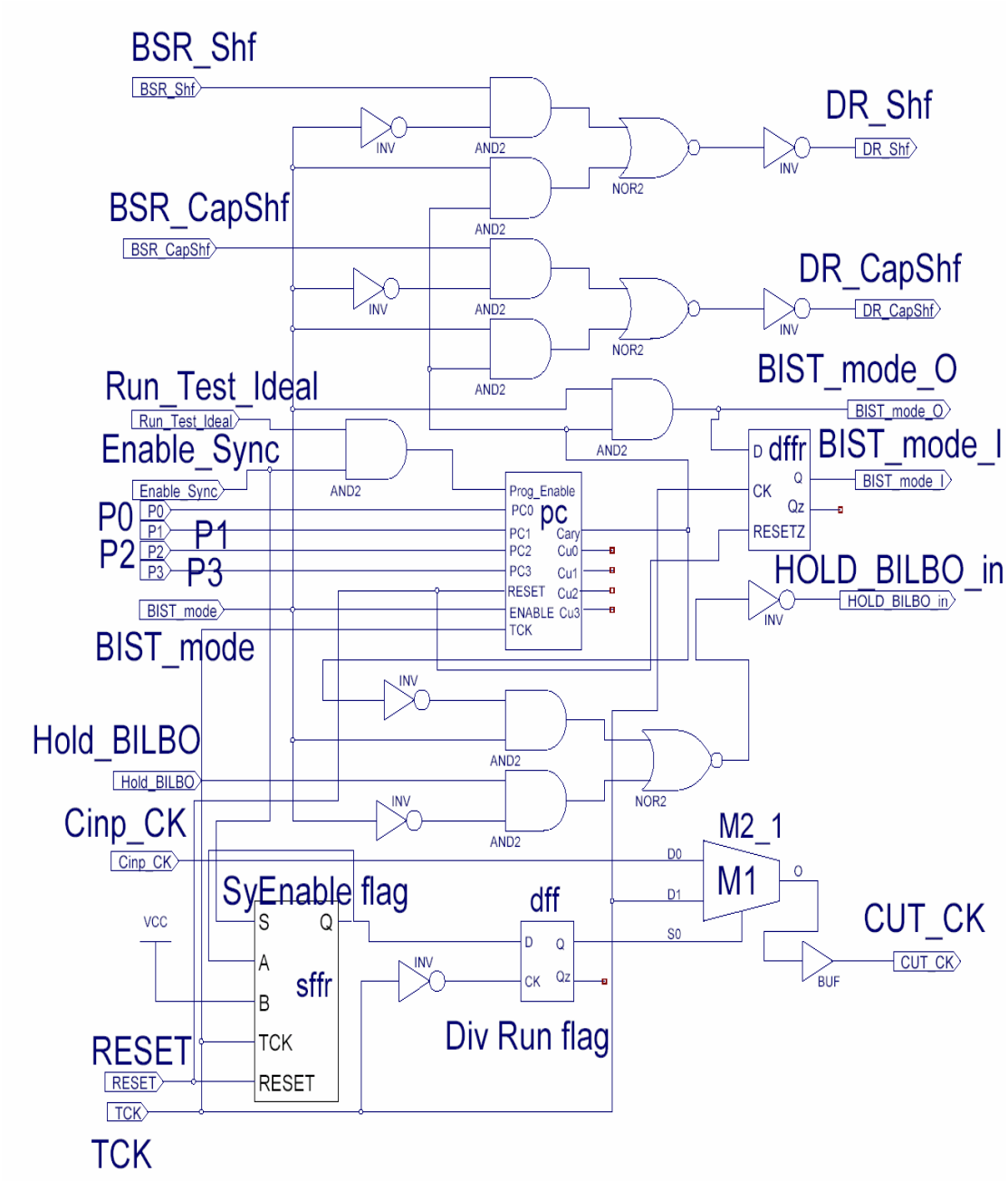


Fig. 8. Schematic diagram of the programmable control unit.

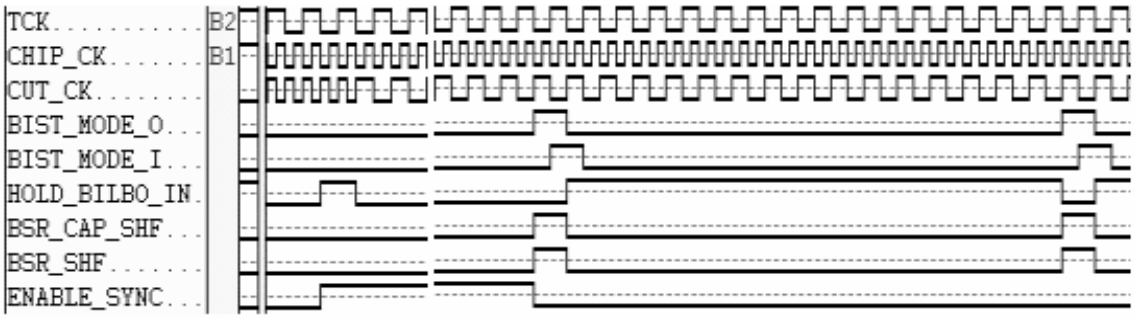


Fig. 9. Timing diagram using the programmable control unit.

As shown in Fig. 10, all cells of test circuitry (FSM, TAPC, IR and BYPAS) are explained carefully in [11], the BIST BSR input cells and BIST BSR output cells are the same as in [11] but connected as 32 cells in series format to achieve the input and output pins of the testable multiplier, the 32-input pins are connected to a constant value (high or low) for test reason point of view to overcome the problem of floating pins.

All the cells for the multiplier and test circuitry are connected altogether and implemented on FPGA chip Xilinx (X2C100). The timing simulation of the complete design is presented to verify proper operation. It is clear from the simulation as shown in Fig. 11 that the generated design netlist operated correctly under applying the input stimulus. It represents the timing simulation of the testable 16-bit parallel pipelined multiplier. Enable_Sync control signal goes high when the Sync instruction is applied; the clock applied to CUT is switched to TCK from chip clock. The PCU hold the *DR_Shf*, *DR_CapShf, BIST_Mode_O,* and *BIST_Mode_I* sixteen clocks for proper timing.

The report generated due to implementation is given as follows:

Minimum period: 119.976ns (Maximum Frequency: 8.335MHz)
Number of Slices: 1,012 out of 1,200 84%
Number of Slices containing Unrelated logic: 0 out of 1,012 0%
Number of Slice Flip Flops: 654 out of 2,400 27%
Number of 4 input LUTs: 1,689 out of 2,400 70%
Number of bonded IOBs: 109 out of 140 77%
Number of GCLKs: 2 out of 4 50%
Number of GCLKIOBs: 2 out of 4 50%
Total equivalent gate count for design: 15,372

The report generated due to implementation shows that the 16-bit parallel pipelined multiplier runs at about 27.459MHz clock speed and it occupies about 11272 gate equivalent (GE) on Spartan X2C100 device. The hardware implementation results show that the testable 16-bit parallel pipelined multiplier runs at about 8.335MHz clock speed and it occupies about 15372 GE on Spartan X2C100 device. The hardware overhead due to the adding test circuitry is 4100 GE, and the maximum clock speed is reduced in the case of testable design.
The hardware overhead of the both BIST BSR I/O is 46 GE [11] and the total number of the BIST BSR implemented in the testable design is 64 cells. The hardware overhead due to the BIST BSR is 2944 GE. The overhead of the BIST BSR input/output cells is high compared to the hardware required by the rest of the boundary scan circuitry. By placing the BIST BSR input/output cells in the area between the I/O pad and core logic, the area overhead in the core area can be reduced. So, the hardware overhead due to the rest boundary scan test circuitry, which is nearly constant for every design, is 1156 GE. This hardware overhead is low with respect to the complete testable design.

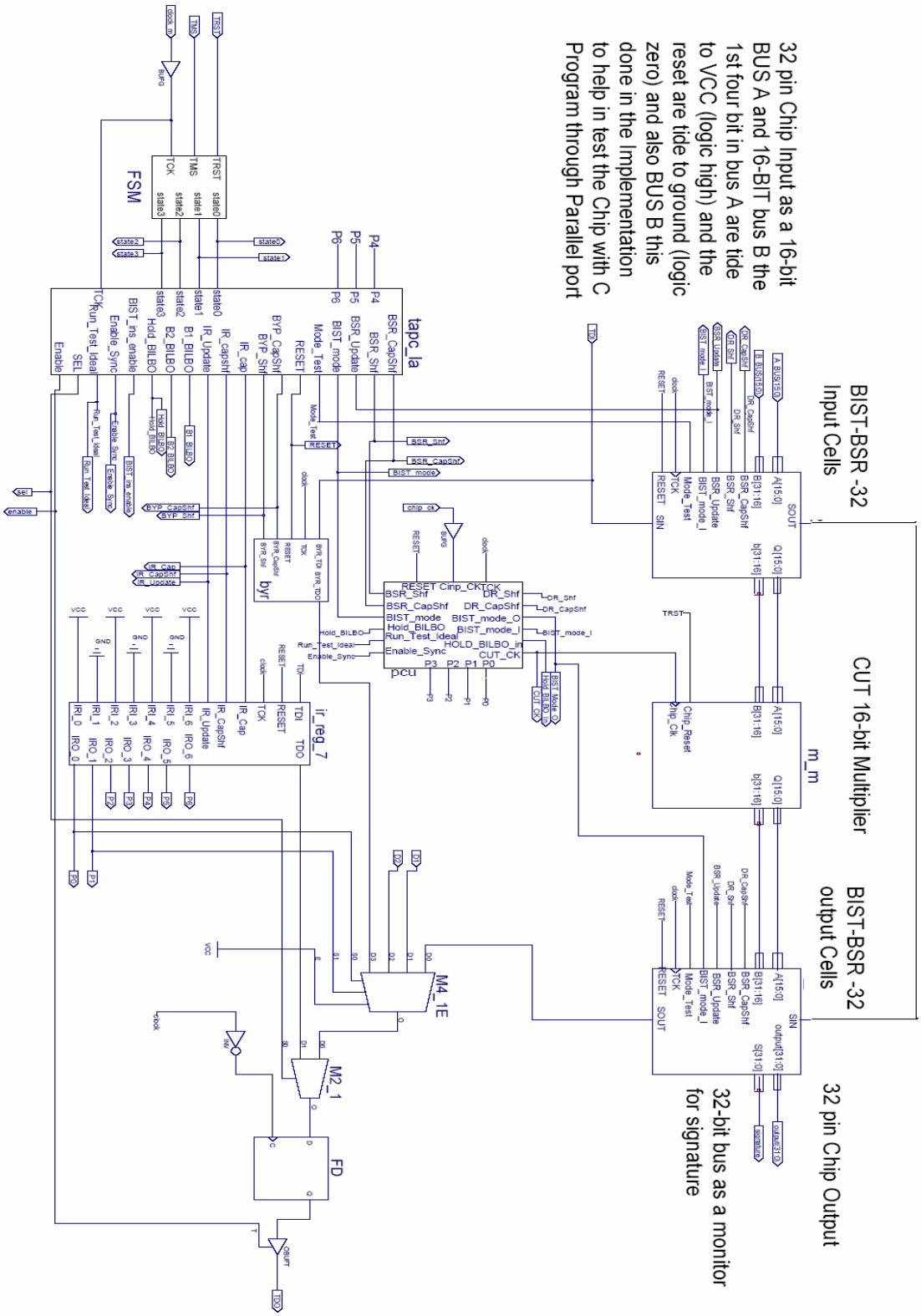


Fig. 10. Schematic diagram of the testable 16-bit parallel pipelined multiplier.

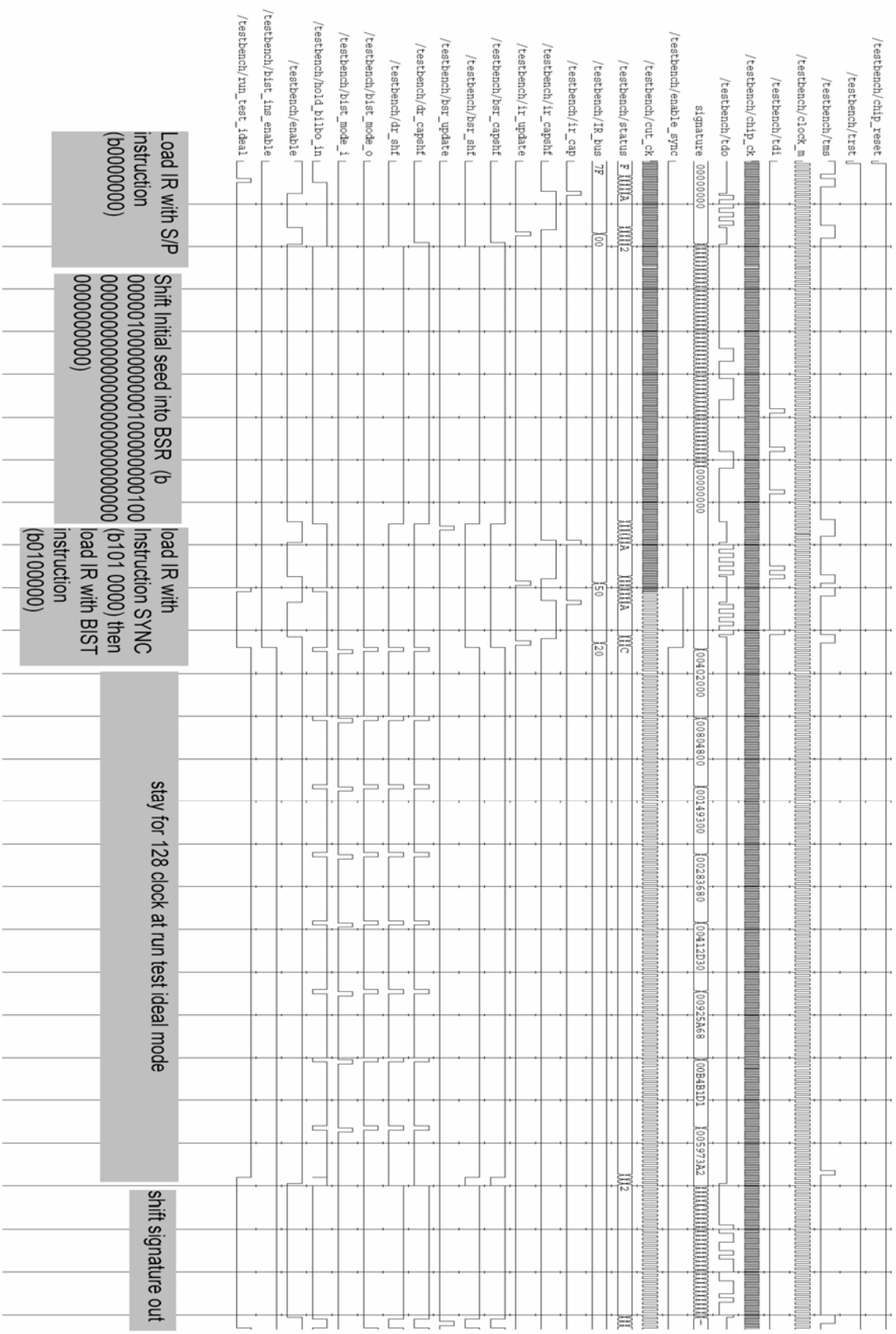


Fig. 11. Timing diagram of the testable 16-bit parallel pipelined multiplier.

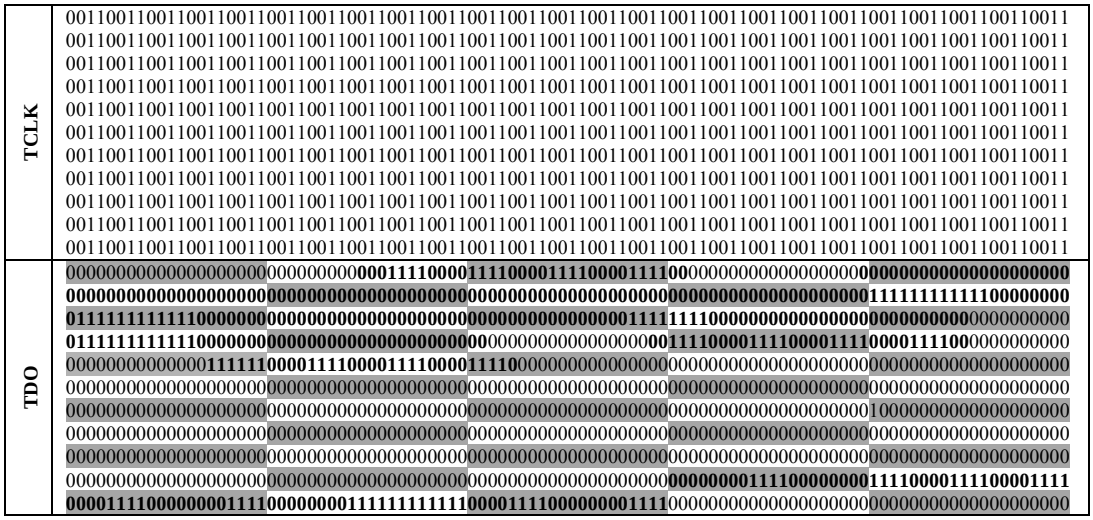
4. EXPERIMENTS AND RESULTS

In this section, the evaluation of the design performance on the FPGA chip as a real time application system is presented. An experiment is established for that purpose. In this paper, the BIST boundary scan architecture for register transfer level (RTL) was presented with the 16-bit parallel pipelined multiplier. All designed cells are connected as shown in Fig. 10. The timing diagram represents the simulation as shown in Fig. 11. All design steps are processed for the implementation on Spartan Xilinx chip X2C100. The timing simulation results are taken for the five pins (TRST, TMS, TCK, TDI, and Chip_CLK) at every transition clock to generate the test digital vector with the expected binary values of TDO and stored in a text file. The test digital vectors are applied to the parallel port, and then to Spartan Xilinx chip X2C100.

Table 2 represents the input/output binary signals generated from the parallel port with C program. The signals are TRST (D0), TMS (D1), TCK (D2), TDI (D3), Chip_CLK (D4) and TDO (S5).

Table 2. Input/output binary signals generated for case (3).

TRST	0111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
	1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
TMS	1111111111110000111111110000000000000000000000000000000000000000000000011111111000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000011111111111100000000000000000000000000000000111111110000
	1111111000000000000000000000000000000000001111111100000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000011110000000000000000000000000000000000001100000000000011111111000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000011110000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
Chip_CLK	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
	0101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
TDI	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000111100000000000000000000000000000011110000000000000000000000000000
	0000000011110000000000000000000000000000000000000000000000000000001111000000001111000011110000000000000000
	0000000000000000000000000000000000000000111100000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000



5. CONCLUSION

In this paper, the FPGA implementation of the BIST boundary scan architecture for 16-bit parallel pipelined multiplier was presented. The schematic based design methods were used to design the complete testable design. The hardware implementation results showed that the hardware overhead due to the adding test circuitry is 4100 gate equivalent, and the maximum clock speed is reduced in the case of the testable design. The hardware overhead of the BIST BSR input/output cells is high compared to the hardware required by the rest of the boundary scan test circuitry. By placing the BIST BSR input/output cells in the area between the I/O pad and core logic, the area overhead in the core area can be reduced. So, the hardware overhead due to the rest of the boundary scan test circuitry, which is constant in every design, is 1156 GE. This hardware overhead is low comparing to the complete testable design.

Designing testable circuits and interfacing it with the computer were presented to evaluate the design as a real time application. The hardware experiment was presented here to evaluate the design performance. The hardware experiment results were compared with the simulation results to verify the design performance.

This approach is considered to reduce the cost of the traditional ATE. The test circuitry is built in the VLSI chip with the slave controller (TAP controller). The personal computer, that is able to be portable, is used as a master programmable controller. So, our objective is to go step in the direction of the portable ATE. Today, technology enables us to do that. The processing of the manufacturing of the VLSI chips migrates towards the Nano-chips with 90 nm processing or less that increases the integration on the VLSI chips.

REFERENCES

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

[2] Mohamed H. El-Mahlawy, Automatic Measurement of Digital Circuits, M.Sc. thesis, Military Technical College, Egypt, (1995).

[3] 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).

[4] 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).
[5] Paul H. Bardell, Willian H. McAnney, Jacob Savir, "Built-In test for VLSI: pseudorandom techniques," John Wiley and Sons, (1987).
[6] Parag K. Lala, "Digital circuit testing and testability," ACADEMIC PRESS, (1997).
[7] Rochit Rajsuman, "Digital hardware testing: Transistor-level fault modelling and testing," Artech House, (1992).
[8] Janusz Rajski, Jerzy Tyszer, "Arithmetic Built-In Self-Test for Embedded Systems," Prentice Hall PTR, (1998).
[9] Manfered Weyerer, and Gerald goldemund, "Testability of electronic circuits," Prentice Hall, (1992).
[10] Alfred L. Crouch,"Design for Test", Prenticice-Hall Inc., (1999).
[11] Mohamed H. El-Mahlawy, "Architecture for BIST Boundary Scan." 10th International Conference on Aerospace Sciences & Aviation Technology, Cairo, Egypt, Session No. 18, 13-15 May (2003).
[12] Mohamed H. El-Mahlawy, Ehab A. El-Sehely, Al-Emam S. Ragab, and Sherif Anas, "Design and Implementation of a New Built-In Self-Test Boundary Scan Architecture." IEEE 15th International conference on Microelectronics, Cairo, Egypt, pp. 27-31, 9-11 Dec. (2003).
[13] Mohamed H. El-Mahlawy, Winston Waller, "A New BIST Boundary Scan Architecture for the Register Transfer Level." 4th International Conference of the Electrical Engineering, Military Technical College, Egypt, pp. 395-403, Nov. (2004).
[14] Kenneth P. Parker, "The boundary-scan handbook second edition analog and digital," Kluwer Academic Publishers, (1998).
[15] Mohamed H. El-Mahlawy, Winston Waller, "An efficient algorithm to design convolved LFSR/SR." IEEE 17th National Radio Science Conference, Minufiya, Egypt, pp. C23 (1-10), Feb. (2000).
[16] Mohamed H. El-Mahlawy, and Winston Waller, " A New Single Test Pattern Generator for Pseudoexhaustive Testing.", 11th International Conference on Aerospace Sciences & Aviation Technology, Military Technical College, Egypt, pp. 989-1002, 15-17 May (2005).
[17] S. W. Golomb, Shift Register Sequences, Laguna Hills, CA: Aegean Park Press, (1982).
[18] Sherif Anas, In-Circuit Testing for Electronic Board, M.Sc. thesis, Military Technical College, Egypt, (2006).
[19] Mohamed H. El-Mahlawy, Winston Waller, "An efficient algorithm to partition the combinational circuits for pseudoexhaustive testing." IEEE 17th National Radio Science Conference, Minufiya, Egypt, pp. C24 (1-11), 22-24 Feb. (2000).
[20] Mohamed H. El-Mahlawy, Winston Waller, "A New Segmentation Approach for Pseudoexhaustive Testing of Combinational Circuits." 4th International Conference of the Electrical Engineering, Military Technical College, Egypt, pp. 251-265, Nov. (2004).
[21] Pak K. Chan, Samiha Mourad "Digital design using field programmable gate arrays", Prentice Hall, (1994).