International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 25 Issue 03, MAR, 2025

REALIZATION OF BUILT-IN SELF TEST (BIST) ENABLED MEMORY (RAM) USING VERILOG HDL

[1]Bysani Priyanka, [2]Guduru Lakshmi Devi, [3]R L B R Prasad Reddy

[1]PG Scholar, Dept. Of ECE, Srinivasa Institute of Technology and Science, Kadapa – 516001.
[2]Assistant Professor, Dept. Of ECE, Srinivasa Institute of Technology and Science, Kadapa – 516001.
[3]Associate Professor, Dept. Of ECE, Srinivasa Institute of Technology and Science, Kadapa – 516001.
[1]priyanka.bysani@gmail.com, [2]lakshmi0449@gmail.com, [3]rajendra.409@gmail.com

Abstract

Built-In Self-Test (BIST) is a technique that allows a set up to check itself for any error on its own. BIST is a screening mechanism that places the testing functions physically with the circuit under test (CUT). BIST can make the system-level design process much simpler in essential applications where system reliability is predominant, and failure is not an option. The decision to execute a critical mission must be made only if the complete system is running without any error. BIST structures generate pseudo random combinations and output results for an exclusive circuit under test are compared. BIST can be implemented on entire designs, design blocks or structures within design blocks. Memory is a complex architecture (fabrication wise) and used in a large number of applications. BIST is basically used to help in the testing of memory with the help of a few extra pins. In fact, while testing a memory using BIST, applying a simple clock signal along with a few pins helps test the entire memory IC. The proposed BIST enabled RAM and controller is designed using Verilog HDL.

Keywords-BIST, Memory, Verilog HDL.

I. INTRODUCTION

The first 4-bit processor was TMS 1000 developed by Texas Instruments, the world's first single-chip microprocessor, was a4-bit CPU; it had Harvard architecture, with an on-chip instruction ROM with 8-bit-wide instructions and an on-chip data RAM with 4-bit words. But, The Intel 4004 is generally regarded as the first commercially available microprocessor. The Intel 4004 is having following specifications:

1. Separate program and data storage.
2. 12-bit addresses
3. 8-bit instructions
4. 4-bit data words
5. Instruction set contained 46 instructions (of which 41 were8 bits wide and 5 were 16 bits wide)
6. Register set contained 16 registers of 4 bits each

Besides this we had refereed the architecture of various other processor like Intel 4040, Atmel MARC4 core [4] Mature, Toshiba TLCS-47 series, HP Saturn (microprocessor), NECPD75X, Epson S1C63 family, etc. From the reference it was clear that processor mainly contains 3 core elements

Arithmetic and Logic Unit (ALU): It is a digital circuit that performs integer arithmetic and logical operations.

Memory & Registers: The RAM or Random Access Memory is used to store the incoming data, intermediate data originated under execution, output of processor which has to be stored for Further execution.

Instruction Decoder: The instruction decoder is the part of the CPU that converts the bits stored in the instruction register into the control signals that control the other parts of the CPU. So if these three elements are designed properly, then a main module of processor can be created (as per problem statement) which will include all the core modules like ALU, RAM, Instruction Decoder, etc.

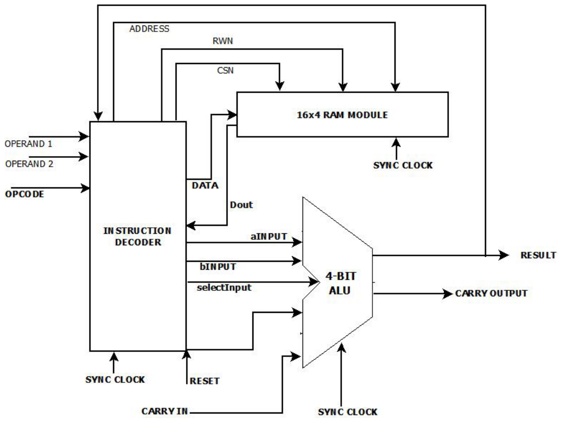


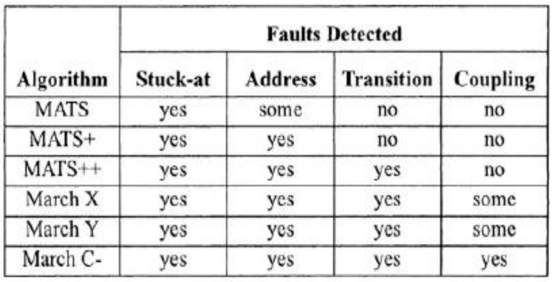
Figure-1: Architecture

A **microprocessor** is a computer processor that incorporates the functions of a central processing unit on a single integrated circuit (IC), or at most a few integrated circuits.[2] The microprocessor is a multipurpose, clock driven, register based, digital integrated circuit that accepts binary data as input, processes it according to instructions stored in its memory and provides results as output. Microprocessors contain both combinational logic and sequential digital logic. Microprocessors operate on numbers and symbols represented in the binary number system.

1.1 BIST architecture

There are a few of test algorithms for RAM testing that are rather simple for BIST implementations include variations of the Modified Algorithmic Test Sequence (MATS) and March tests including the March Y algorithm that used for the FSM based TPG [4]. Among these test algorithms, the March C- algorithm is practical to offer the highest fault coverage [4]. For detail, the capabilities of the fault detection of March test algorithms are summarized in Table 1. The test algorithms are planned for RAMs with one data bit per word. Nonetheless, multiple bits per word can be applied as well. In order to increase the fault detection rate, modification on the algorithm is required for sensitivity and coupling faults in RAM [5].

Table-1: The Summarized Fault Detection of March Test Algorithms

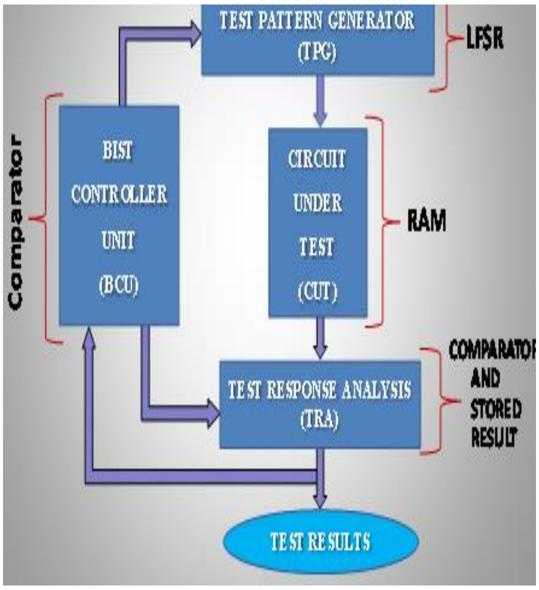


Algorithm	Faults Detected			
	Stuck-at	Address	Transition	Coupling
MATS	yes	some	no	no
MATS+	yes	yes	no	no
MATS++	yes	yes	yes	no
March X	yes	yes	yes	some
March Y	yes	yes	yes	some
March C-	yes	yes	yes	yes

II. EXISTING METHOD

The fundamental BIST construction needs the inclusion of additional three h/w building blocks to a digital circuit with Circuit under test (CUT). They are i) A test pattern generator ii) A response analyzer and iii) A test controller as shown in Figure 2. The test pattern generator creates the test patterns for the CUT. In standard way of exercise, the CUT accepts its inputs from other building blocks and execute the task for which it was planned. During test mode, pseudo random test patterns are given to the CUT, and the output test results are evaluated by comparator. For the proposed work CUT is standard RAM, Test Pattern Generator (TPG) is implemented by simple LFSR, BIST Controller Unit (BCU) and Test Response Analyzer (TRA) is implemented by comparator.

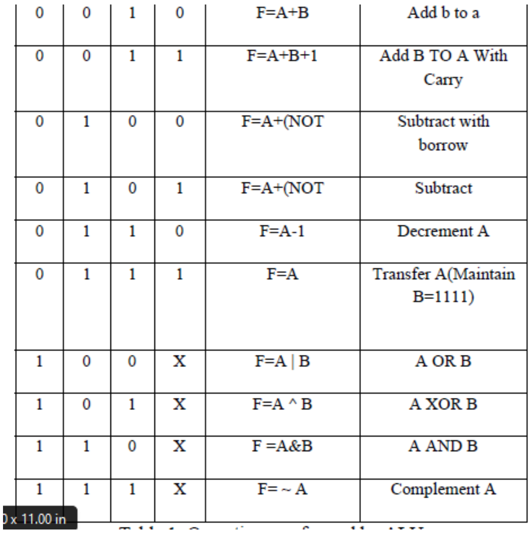
Figure-2: General BIST Construction


TEST PATTERN GENERATOR (TPG)
LFSR
Comparator
BIST CONTROLLER UNIT (BCU)
CIRCUIT UNDER TEST (CUT)
RAM
COMPARATOR AND STORED RESULT
TEST RESPONSE ANALYSIS (TRA)
TEST RESULTS

III. DESIGN METHODLOGY

The test begins starting with the BIST Controller that gives input and determine the circuitry either in BIST mode or normal mode. As the counter receives input from BIST Controller, it starts to count from all 0s until reaches all 1s. The test pattern generated by counter is written into memory starting from 0s that stored in location 0. The counter increments when the same pattern is stored into all RAM location and cause the LSB of the address to turn over to all 0s. The RW bar signal becomes '1' at the same time when the same data being read from all memory location and the address loop back at 0. Then, the next test pattern begins to be written into all locations. This process iterate for the eight test patterns generated by decoder.

A 4 bit ALU has been designed to perform various arithmetic and logical operations on 4 bit data. The ALU has three selection bits (S0S1S0), one carry input bit (C_{in}), two 4-bit data(A and B) and an active low reset as input pins and it has four bit storage (F) and one bit storage (C_{out}) as output pins. The table given below describes the various operations performed by processor and its pseudo.



0	0	1	0	F=A+B	Add b to a
0	0	1	1	F=A+B+1	Add B TO A With Carry
0	1	0	0	F=A+(NOT	Subtract with borrow
0	1	0	1	F=A+(NOT	Subtract
0	1	1	0	F=A-1	Decrement A
0	1	1	1	F=A	Transfer A(Maintain B=1111)
1	0	0	X	F=A | B	A OR B
1	0	1	X	F=A ^ B	A XOR B
1	1	0	X	F =A&B	A AND B
1	1	1	X	F= ~ A	Complement A

Table - 2: Operations performed by ALU

The processor instructions can be broadly divided into two classes. Arithmetic operation and Logical operation. At the end of execution of each instruction memory write operation is performed to store the result back to the address specified by the instruction decoder. As shown in adjoining figure (2), the LHS of RTL (Register Transfer Level) Schematic of Arithmetic and Logical unit has various inputs like two 4 bits 'a' & 'b' signals, 3 bits operation selection signal, carry input from previous execution if multiple fetch and execute cycles are required, etc. On the other hand the 4 bit result is available along with OVERFLOW flag.

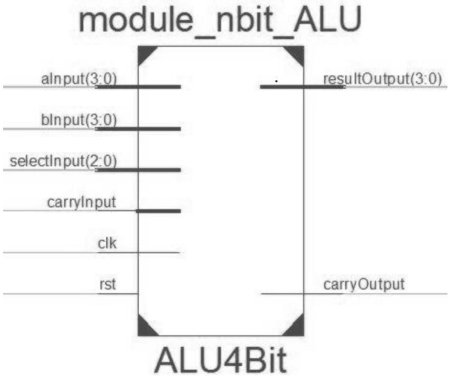


Figure- 3: Pin diagram of 4 bit ALU

3.1 Design of 16x4 Ram

The RAM is required to store various data helpful for instruction execution purpose. It also contains the intermediate result for multiple fetch and executes operations. Term 16x4 corresponds that the memory module is 4-bit wide and 16-bitdeep. Therefore the 16x4 RAM stores maximum of 8 byte data.

We had used two dimensional array declarations to define memory blocks. The declaration by this fashion also helps in synthesis as Technology schematic will represents the four,16x1 memory blocks. Whenever Data out is not used i.e., when memory is in write mode or if chip select is high, the Data out is set to "ZZZZ" (meaning nothing is driven onto bus).

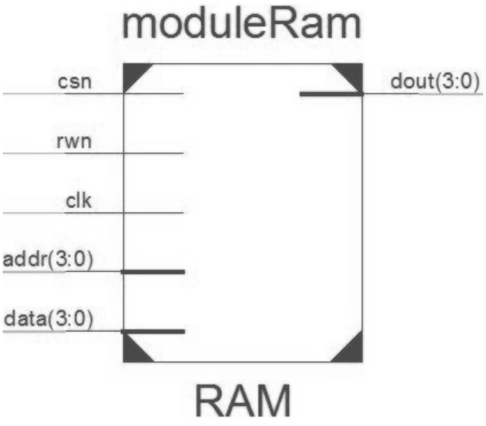


Figure-4: Pin diagram of 4 bit RAM

3.2 Design of Instruction Decoder

The instruction decoder can be thought of as a finite state machine which changes its state from the present state to next state on the rising edge of the clock signal. The instruction decoder module takes input compromising of a three bit opcode and two 4-bit operands. When asynchronous reset is high the decoder unit goes to init state. With the positive edge of every subsequent clock the unit cycles between Fetch, Execute and Load state. At each of these states the operations to be performed are as given below**:**

Fetch – Fetches data from the location specified by the first operand. The first operand specifies the reference by memory allocation. This data is store in operand an Input.

Execute – Execute the Instruction using ALU after setting all control lines depending on instruction. Select Input lines are set to the value of opcode, a Input is same as obtained in the Fetch
state and b Input is an immediate data supplied to the instruction.

Load – After completion of ALU operation, result is contained in 4-bit storage (F) which is stored back into the location specified by first operand. International Conference for Convergence of Technology.

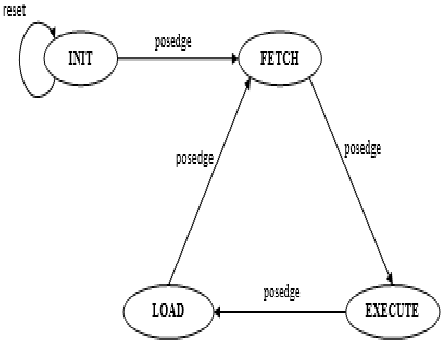


Figure-5: FSM design for instruction decoder

The figure 6 depicts the RTL (Register Transfer Level) Schematic Pin-Diagram of Instruction Decoder; the LHS pins are Input ports to the instruction decoder. The operand 1 is reference by address value of memory, whereas the operand 2is reference by immediate value. The opcode is 3 bit long so that all the ALU operations can be enclosed. On the other hand on output side, a Input, b Input& select are provided to ALU, rest all are given to 16x4 RAM.

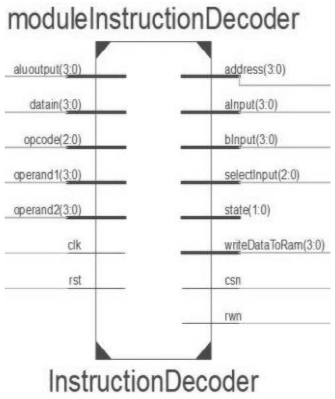


Figure - 6: Pin diagram of Instruction Decoder

IV. SIMULATION RESULTS

The entire simulation and result obtaining using test vectors are done with the help of Xilinx software.

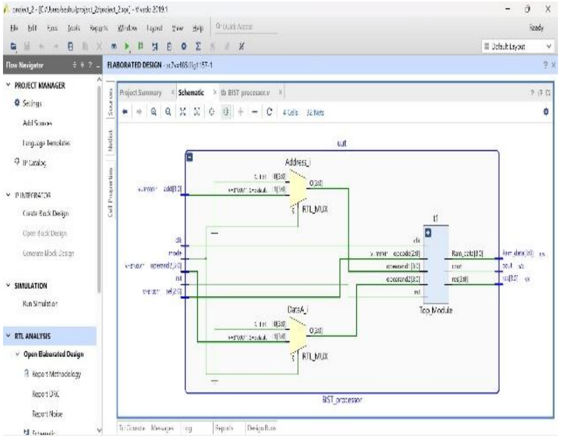


Figure – 7: Proposed System Design

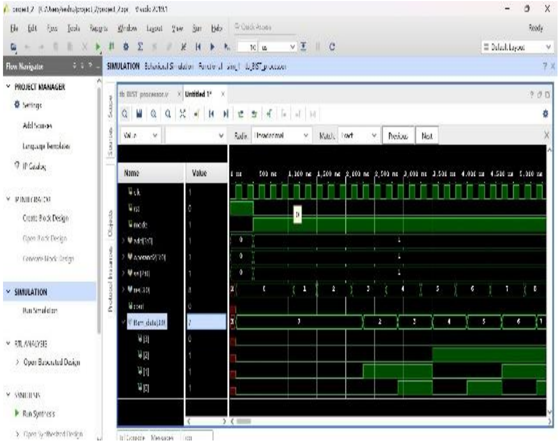


Figure – 8: Simulation Outputs

V. CONCLUSION

We have designed and verified the functionality of basic four-bit microprocessor using structural modeling which combines three elementary modules i.e. four-bit ALU,16x4 RAM and Instruction Decoder. The processor design that we carried out is very basic but for understanding and developing design of new age complex processor architectures like multi-core design, the knowledge of basic design is inevitable.

As a part of future enhancement to the architecture, various addition is possible like dual-core processor design, to include more variety of operations to be carried out by ALU, different assortment of DSP (Digital Signal Processing) functions like convolution, time shifting, Fast Fourier transform operations can be added. We can also include serial interface block so that RS232 port can be attached to processor.

REFERENCES

[1] N. Kavvadias, P. Neofotistos, S. Nikolaidis, C. A. Kosmatopoulos, andT. Lao Poulos, “Measurements analysis of the software-related power consumption in microprocessors,” IEEE Trans. Instrum. Meas., vol. 53,no. 4, pp. 1106–1112, Aug. 2003.

[2] J. Teifel and R. Manohar, “An Asynchronous Dataflow FPGA Architecture,” IEEE Transactions on Computers, vol. 53, no. 11, pp.1376–1392, 2004.

[3] Bal Pande, R.S. and Keote, R.S., "Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor," in Proc.of International Conference on Communication Systems and Network Technologies, pp. 409-413, 2011.

[4] Joaquín Olivares, et all, “Teaching Microprocessor Design Using FPGAs”, IEEE EDUCON Education Engineering 2010, April 2009,Spain.

[5] Xilinx® Inc., Pico Blaze 8-bit Embedded Microcontroller User Guide, Xilinx® 2005

[6] Wang Min, The Principle of Computer Organization, Electronics Industry Publishing House, Beijing, 2003