

## SIMULINK MODEL FOR CONTROLLABILITY AND OBSERVABILITY OF VLSI CIRCUITS

Gurinder Pal Singh<sup>\*1</sup> and Balwinder Singh<sup>2</sup>

\*<sup>1,2</sup> ACS Division, CDAC (Centre for Development of Advanced Computing), Mohali, India

gurinderpalsingh900@gmail.com<sup>1</sup>

balwinder\_cdacmohali@yahoo.com<sup>2</sup>

**Abstract:** This paper determines the controllability and observability in MATLAB/simulink, which is very easy way for handling complex VLSI circuits. As the complexity of VLSI circuits is increasing, calculations for controllability and observability by mathematical formulas is becoming tedious job for each and every gate in a complex circuit, and further if circuit is sequential then additional mathematical calculations for flip flops are required. To reduce the calculation complexity for user, two custom libraries are created, one for combinational circuits and another for sequential elements. Then a user interactive environment a GUI (Graphic User Interface) in MATLAB is presented, for forcing the inputs from outside of the simulink. Combinational and sequential circuit of each type is modelled with user custom library and controllability and observability of these circuits is verified.

**Keywords:** GUI, simulink, Controllability and Observability, CC (combinational controllability), CO (combinational Observability, SC (sequential Controllability), VLSI (very large scale integrated circuits testing).

### INTRODUCTION

As the VLSI circuit's integration level increases, the testing task becomes more intricate due to decrease in gate dimensions. The gate size is very small, logical blocks are beyond the reach of testing, and as consequence, testing of VLSI circuits becomes more sophisticated and expensive. Two essential notions are used in the analysis of VLSI circuit's testability: Observability of a circuit node and Controllability of a circuit node. These terms are discussed in next sections.

Here MATLAB/simulink is used to calculate the Controllability and Observability of the circuit by designing a custom or user's library on the bases of the mathematical testing formulas. Simulink provides an environment for intellectual properties (IP) building block based circuit engineering design as well as project simulation environment. A user interactive GUI (Graphic user interface) is created, which consists of window containing menus, buttons, text etc., which is manipulated interactively with the mouse and keyboard.

The MathWorks has provided for MATLAB programmers with a set of structured event driven components in the form of user interface controls (uicontrols) and menus (uimenus) that are easily assembled and used to create GUIs. This tool is used to increase the productivity of a user or to provide a window to the sophistication and power of a MATLAB application for people with little or no MATLAB programming experience. With MATLAB there are two main principle elements required to create a GUI

### COMPONENT

Each item on a MATLAB GUI (pushbuttons, labels, edit boxes, etc.) is a graphical component. The types of components include graphical controls, static elements (frames and text strings), menus, and axes.

### CALLBACKS

Callback is an event, which is generated when a user clicks on a button. Whenever an event occurs due to the click by the user, that event performs its function and causes the MATLAB code that implements the function of the button to be executed.

### RELATED WORK

The idea of calculating the controllability and observability and its role in design for test has been investigated in a few papers and books. In[6], testability requirement about their underlying test parameters that are controllability and observability has explained. In[5], Ruthan's System Model was used for calculating the controllability and observability measures. General formulas for calculation of controllability and observability of sequential elements (flip-flops), and all gates has explained in[2]. In [1 & 2], development of simulink model has discussed. MATLAB GUIs designing is explained in [8 & 9].

### COMBINATIONAL CONTROLLABILITY AND OBSERVABILITY ANALYSIS

Two calculation options exists for the combinational circuits nodes:

a). Calculation of combinational controllability as the number of elementary operations necessary for setting the primary inputs (PI) to the zero that is 'CC0' and or to unity'CC1'. For this we progress through the circuit in a forward pass, in level order. EXNOR is taken as an example

and it's CC (combinational controllability) formulas are shown in equation1 and 2.

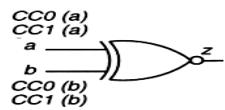


Figure.1 EXNOR-gate for its controllability

$$CC0(z) = \min(CC1(a)+CC0(b), CC0(a)+CC1(b))+1 \dots\dots(1)$$

$$CC1(z) = \min(CC0(a)+CC(b), CC1(a)+CC0(b))+1 \dots\dots(2)$$

Text of manuscript should be arranged in the following order: Title, Abstract, Introduction, Body Text, Results and Discussion, Conclusion, Acknowledgements and References.  
a).For a logic gate with an input signal that needs to be observed at the output called controlled Observability 'CO' of the particular input. The difficulty of observing the input equals the Observability of the output plus the difficulty of setting all other inputs to non-controlling values plus 1 to account for the logic depth.

$$CO(a) = CO(z) + \min(CC0(b), CC1(b))+1 \dots\dots(3)$$

$$CO(b) = CO(z) + \min(CC0(a), CC1(a))+1 \dots\dots(4)$$

The controlabilities ranges between 1 to infinity, and observabilities lie between 0 to infinity. The higher the measures, more difficult it will be to control or observe.

#### **Development of MATLAB/simulink model to calculate of Controllability and Observability:**

By taking EXNOR as an example a simulink model is designed for controllability and Observability. This model is designed in MATLAB/simulink by using the blocks, Add block, min/max block .Source from the workspace, sink to the workspace, and sink as display block from simulink library. And we get a model as shown in Figure.2. Block named "from workspace" is used either to take values from command window or form GUI and block named "to workspace" is used to send the final result either at command window or to GUI edit text component, to see the final output value directly at GUI. For this purpose in the property of "simulink to work space" save format is set on "array". In Similar manner the models for all other gates are designed.

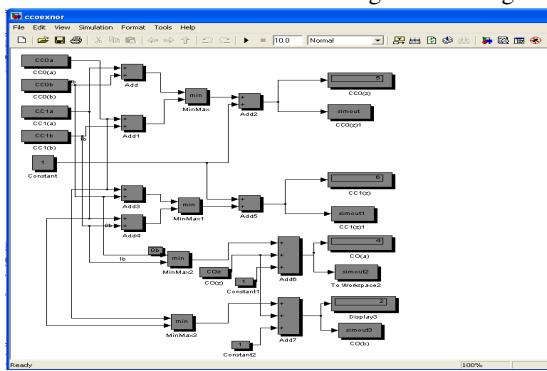


Figure.2 Simulink model for EXNOR-gate.

#### **Development of MATLAB GUI for designed simulink model**

To control the Model from outside of simulink workspace environment a GUI is created. All the input values for all primary inputs are provided from GUI in the form of dimensional array, simulation time that is, step time, sample time and sim time all provided from GUI. GUI callbacks are written in ".m-file". GUI communicates with simulink by the following command which was written in .m-file under the push button callback:

```
Options=simset('SrcWorkspace','current');
```

```
sim('ccoexnor.mdl',[],options);
```

Here simset is used to set the simulink model to take data from the workspace, and 'sim' designates for simulink, and 'ccoexnor.mdl' is the name of simulink model. When data entered into editable text in GUI, and after pressing push button named here as push for controllability and Observability we get the output as shown in Figure3.

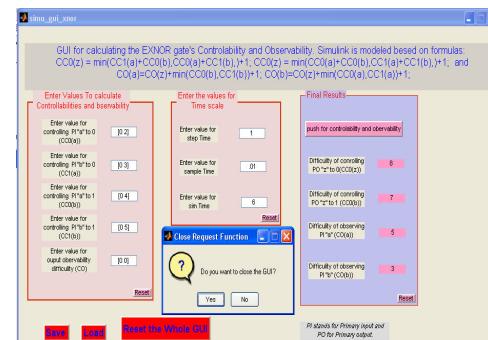


Figure 3 GUI interface for simulink EXNOR-gate

To make GUI more user friendly some extra functions are added like save, load, exit on demand. Then after this by using deployment tool in MATALB .exe file is generated which is helpful for those user who do not have MATLAB on their systems. If circuit to be modeled has very large number of gates then its modeling is very complicated, so to reduce the level of complexity, subsystems of all gates are created, and tested for different values as shown in table1. After creating subsystem that system is masked.

Table1. Outputs of EXNOR subsystem for different values

CC Primary Inputs for EXNOR	CC0a	2	1	5
	CC1a	3	6	2
	CC0b	4	5	1
	CC1b	5	3	3
	COz	0	0	0
<b>RESULTS</b>				
	CC0z	CC1z	C0a	C0b
EXNOR	8	7	5	3
	5	7	4	2
	4	6	2	3

#### **Custom Libraries Generation**

To make all designed subsystems user friendly, a custom or user library is created. This library includes subsystems as

blocks of all gates (AND, OR, NAND, NOR, EXOR, EXNOR, NOT and 3 Primary input NOR gate). To make more convenient to user, we add this library into simulink Browser, as shown in Figure 4.

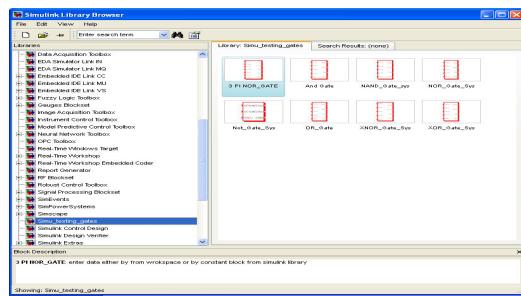


Figure 4: Simulink user modified library

To do this a “slblocks.m” file is generated and this file is saved in the same directory which include the custom library. “slblocks.m” file includes:

```
function blkStruct=slblocks
blkStruct.Browser(1).Library='Gate_library';
blkStruct.Browser(1).Name='Simu_testing_gates';
Where 'Gate_library' is the name of custom library, and "Simu_testing_gates" is the name under which "gate_library" is shown in simulink browser.
```

#### **Combinational controllability and observability of combinational circuit using custom library**

Using a circuit as shown in Figure5, is taken for verification of the user-designed library. In this the gates are directly loaded into workspace from simulink browser, in which our newly designed library exists as shown in Figure4.

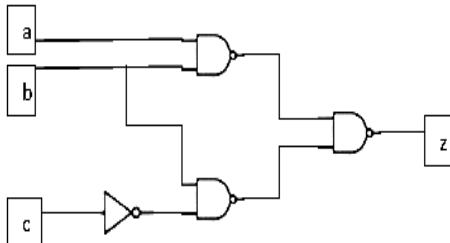


Figure 5: Combinational Circuit to be implemented

In above circuit there are three primary inputs named a, b, c and z is the primary output. Circuit includes three NAND gates and one NOT gate. The Circuit is designed and simulated in simulink and final results are observed on display block as shown in Figure6.

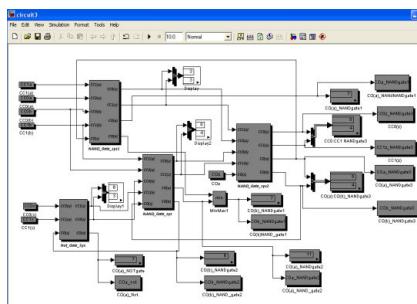


Figure 6: Simulink model using Custom library

Inputs of this model in simulink are controlled from the outside of the workspace, here we have controlled this by designing a GUI of this circuit similar to Figure3, and different values are forced and corresponding output are observed that are given in table2, to verify the model..

#### **Combinational Controllability and Sequential Controllability of sequential circuits and development of sequential Custom Library**

The sequential circuit as shown in Figure5 would have two types of calculations, one to calculate combinational controllability (CC) and second is sequential controllability (SC). Sequential controllability roughly measures the number of times various flip-flops must be clocked to control a signal. The sequential controllability equations for basic logic gates differ from the equations of combinational gates only in a way that, a 1 is not added as we move from one level of logic to another, but rather a 1 is added when a signal passes through a flip-flop, as shown in equation number 7 and 8.

$$CC1(Q) = CC1(D) + CC1(C) + CC0(C) + CC0(\text{reset}) \dots (5)$$

Table2. Controllability Observability for different

CC PI Primary Inputs	CC0a	2	3	1
CC1a	1	2	2	
CC0b	3	1	1	
CC1b	1	5	1	
CC0c	5	2	2	
CC1c	2	2	3	
COz	0	0	0	
<b>RESULTS</b>				
GATES	CC0z	CC1z	C0a	C0b
NAND gate2	8	4	5	4
	5	9	3	3
	5	5	3	3
NAND gate1	3	3	7	7
	8	2	9	6
	4	2	5	6
NAND gate	3	3	1	6
	9	2	7	9
	5	2	5	7
NOT gate	6	3	7	
	3	3	10	
	3	4	6	

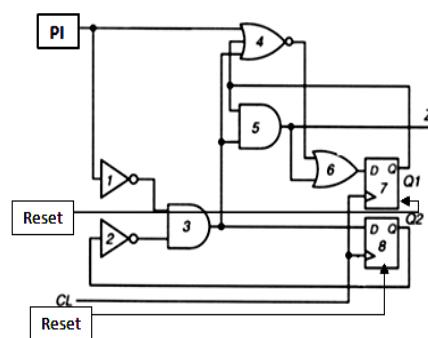


Figure7: Sequential Circuit to be implemented.

$$CC0(Q) = \min [CC1(\text{reset}) + CC1(C) + CC0(C), CC0(D) + CC1(C) + CC1\odot + CC(C)] \dots (6)$$

Here  $CC1(Q)$  and  $CC0(Q)$  are the combinational difficulties of controlling output( $Q$ ) which measures how many lines in the circuit must be set to make output( $Q$ ) as 1 and 0 respectively. As flip-flop are also in the circuit so for calculation of sequential controllability, we have equations 7 and 8.

$$SC(Q) = (SC1(D) + SC1(C) + SC0(C) + SC0(\text{reset})) + 1 \dots (7)$$

$$SC0(Q) = \min [SC1(\text{reset}) + SC1(C) + SC0(C), SC0(D) + SC1(C) + SC1(C) + SC0(C)] + 1 \dots (8)$$

Here  $SC0(Q)$  is sequential difficulties of controlling output( $Q$ ) to zero and  $SC1(Q)$  measures how many flip-flops in the circuit must be clocked to set  $Q$  to 1. For sequential circuits a new library is generated because of lacking right-hand additional unity as compared to logic gates in combinational circuits. Newly generated library is added to simulink browser as similar to Figure 4. Now to implement the custom library of the circuit shown in Figure 5 is modeled in the simulink and outputs are observed on the display block, as shown in Figure

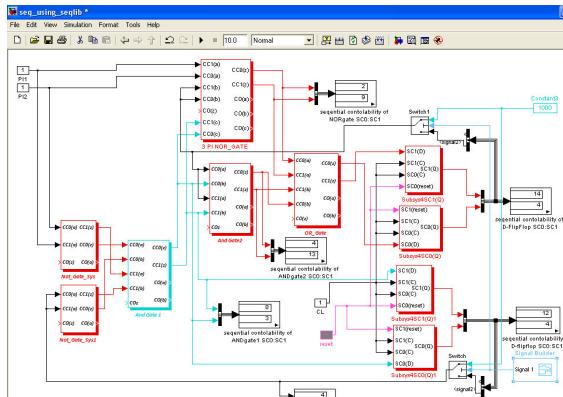


Figure 8. Simulink model for Sequential Controllability

## RESULT AND DISCUSSION

From this designed libraries it is clear that level of complexity is decreased during the calculation of controllability and observability. Because, now instead of using the formulas for these parameters we are using custom libraries that encapsulates that formulas, and attachment a GUI make the calculations easier. Results are shown in tabular for in table no.1 and2.

## CONCLUSION

Controllability and Observability are incredibly useful in guiding ATPG algorithms during back tracing, but if circuit includes large number of logic gates and flip-flops than it is very difficult to calculate these parameters and it becomes more time consuming. To sort out all these problems, In this Paper a new way for calculation of controllability and observability is presented which is based on the formulas of these parameters. A custom library is designed in simulink at

both the combinational and sequential level. By using these libraries two circuits are modeled. Results are verified and finally shown in the tabular form. A Graphic User Interface (GUI) is designed in MATLAB and linked to the circuits, which are designed by using custom libraries, to give an interactive environment to the end user. By using this approach of custom designed library in MATLAB/simulink the complexity level of the ATPG algorithm that is D-algorithm and PODEM (path-oriented decision-making), LFSR and BIST techniques for VLSI testing, can be reduced to much extent.

## REFERENCES

- [1] Bushnell M.L., Agrawal V.D., 2004. Essential of Electronic Testing for Digital Memory & Mixed-SignaVLSI Circuits. Kluwer Academic Publishers, Dordrecht, The Netherlands,
- [2] Evgeni Perelroyzen, 2007. Digital Integrat Circuits Design for test Using Simulink and Stateflow. CRC Press Taylor&Francis Group, London,
- [3] Dabney J.B., Harman T.L. Mastering 4(.).2001, Prentice Hall, New York. 2<sup>nd</sup> edition.
- [4] Kenneth M.Butler; Rohit Kapur; M.Ray Mercer; Don E. Ross;" 1992. The Role Of Controllability and Observability in Design for Test",IEEE VLSI Test Symposium
- [5] Kirovski, D, Potkonjak, M. Guerra, L.M., 1999. Improving the Observability and controllability of datapaths or emulation based debugging IEEE conf. on Computer Aided Design of Integrated Circuits and systems.
- [6] Mohamed Jamoussi, Bozena Kamonsk, 1993. Controllability and Observability Measures For functional-Level Testability Evaluation, VLSI Test Symposium, Digest of Papers., Eleventh Annual IEEE
- [7] Natick, MA,2004,Using Simulink: Dynamic System Simulation for MATLAB. The Mathworks, Inc.
- [8] Patrick Marchand and O. Thomas Holland, 2003. Graphics and GUIs with MATLAB, Chapman and Hall/CRC, 3<sup>rd</sup> Edition.
- [9] Weinshenk, S., Jamar, P.Yeo, S. C., 1997. GUI Design Essentials, John Wiley and Sons, Inc., New York, NY.
- [10] Real-Time Windows Target: For Use with Real-Time Workshop. User's Guide The MathWorks, Inc., Natick, MA,2004.

## AUTHORS



**Gurinder pal Singh** received the B.Tech. (electronics and communication engineering) degree from the Punjab Technical University, Jalandhar in 2008, pursuing M.Tech. (VLSI design) degree from CDAC, Mohali, Punjab Technical University, 2009-11 batch.



**Balwinder Singh** has obtained his Bachelor of Technology degree from National Institute of Technology, Jalandhar and Master of Technology degree from University Centre for Inst. & Microelectronics (UCIM), Panjab University, Chandigarh in 2002 and 2004 respectively. He is currently serving as Design Engineer in Center for Development of Advanced Computing (CDAC), Mohali and is a part of the teaching faculty. He has 5+ years of teaching experience to both undergraduate and postgraduate students. Singh has published two books and many papers in the International & National Journal and Conferences. His current interest includes Genetic algorithms, Low Power techniques, VLSI Design & Testing, and System on Chip.