

VERIFICATION COVERAGE AND SYNTHESIS OF AN ETHERNET ASIC

Lúcio R. Prade, Marco A. B. Hennes, Vitor A. P. Righi, Robert Torrel, Rafael R. dos Santos

{lucio, hennes, vitorrighi, roberttorrel}@mx2.unisc.br
rsantos@unisc.br

Universidade de Santa Cruz do Sul – UNISC
Av. Independência 2293, Bloco 53
Santa Cruz do Sul, RS, 96815-900, Brazil

ABSTRACT

This paper presents the verification coverage and logic synthesis results of Ethernet MAC Controller logic. A testplan for the functional verification was developed and the resulting coverage in terms of statements, branches and conditions was evaluated. Furthermore, the design was synthesized to a Virtex II Pro FPGA (for validation purposes) and later to an ASIC using the TSMC 0.35 technology.

1. INTRODUCTION

One of the most popular data communication techniques is the Ethernet. It was originally based on the idea of computers communicating over a broadcast transmission media. The coaxial cable was later replaced by a twisted-pair of wires connecting computers through hubs or switches.

Nowadays, several embedded systems are using such technology to perform data exchange among different systems. The specification of the MAC Ethernet (Media Access Control Ethernet) is described by the IEEE 802.3 standard and it defines the scheme known as the carrier sense multiple access with collision detection (CSMA/CD) which determines the way computers access a shared channel [4].

In this paper we revisit the normal ASIC flow and focus on the functional verification coverage and logic synthesis of a subset of the MAC Ethernet logic.

2. DESIGN FLOW

There are four basic stages in the design flow of an ASIC (Application-specific Integrated Circuit): design, test, fabrication and packaging [2] (Figure 1). The design stage is usually divided into three major tasks: conceptualization and modeling, synthesis and optimization and finally validation. On the conceptualization and modeling an idea is transformed into a model, which captures the functionality that the resulting circuit would perform. The synthesis consists of refining the model, from an abstract level into a more

detailed one that defines the necessary features and characteristics for fabrication. The validation is the task where the model and its lower level representations are verified against the original specification.

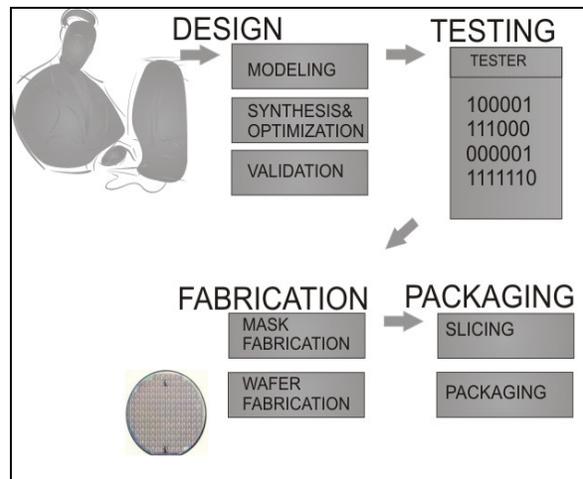


Figure 1. The four stages of the design flow of an integrated circuit

An actual ASIC design is usually taken from the highest level of abstraction through the lowest level. The first representation of an idea is normally coded using an HLL (High Level Programming Language) or an HDL (Hardware description Language) [1]. This first description details the main functionality without the associated details. In other words, the high level model describes the specific functionality which is intended to be built into the circuit.

The circuit synthesis adds to the initial model by appending more detailed information about the actual circuit needed to realize the functionality being modeled. The model is a representation of the functionality without details. As the synthesis proceeds generating lower levels of abstractions, more details are added to the model until the actual geometric representation of the circuit is obtained.

3. THE ETHERNET MAC

This study was developed based on a public IP-Core available at the Opencores repository [6]. The MAC core was originally written in Verilog and it represents the full functionality of a MAC Ethernet according to the IEEE 802.3 standard [4].

Figure 2 shows the block diagram of the MAC core where the top-level (eth_top.v) and its submodules are presented. The complete MAC model is represented by seven submodules named: eth_mii, eth_registers, eth_maccontrol, eth_txethmac, eth_rxethmac, eth_wishbone, eth_macstatus and some logic used for synchronization, multiplexing and registering outputs.

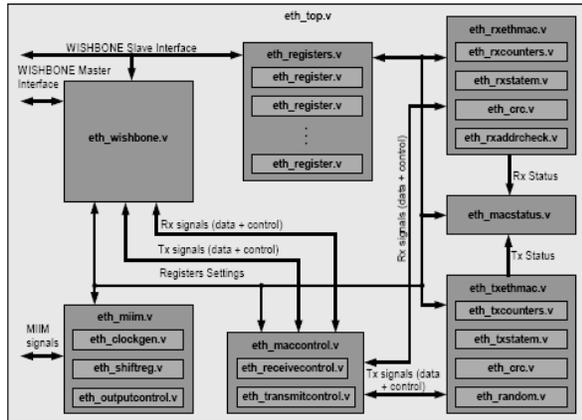


Figure 2. Block diagram of the MAC IP-core [6].

The goal of this work is to carry a design throughout all the stages. In order to achieve that without spending too much time designing such a complex device, the MII (Media Independent Interface) submodule was chosen to get started with. The main reason behind this choice is that by designing the MII submodule it will be possible to fabricate a device and test it without having to design the complete MAC module.

The MII submodule is an interface to the external Ethernet PHY chip. It is used for setting the PHY's configuration registers and to access its status. On top of that, a whole set of other functions necessary to work with the external PHY are also implemented in the MII such as a clock divider and logic to synchronize signals between the PHY and the MAC.

The MII interface consists mainly of two signals: clock (MDC) and the bi-directional data signal (MDIO). The MDIO signal needs to combine the input signal Mdi, the output signal Mdo and the enable signal MdoEn.

4. FUNCTIONAL VERIFICATION

A testplan was developed in order to verify the functionality [3] of the MII submodule where a set of 17 testcases cover the main aspects of this submodule. The software Questa [7] by Mentor Graphics was used to

measure the verification coverage of such testcases. Table 1 presents the coverage results of this analysis.

Verification Coverage - MII									
	Stmts			Branches			Conditions		
	Active	Hits	%Cov	Active	Hits	%Cov	Active	Hits	%Cov
eth_mii	83	83	100	46	45	97.8	18	18	100
eth_clockgen	10	10	100	10	10	100	0	0	100
eth_outputcontrol	15	15	100	8	8	100	0	0	100
eth_shiftreg	12	12	100	16	16	100	0	0	100

Table 1. MII verification coverage results

The results show that 100% of the statements, 97.8% of the Branches were hit and 100% of the conditions were also hit. This represents a very high level of coverage for the MII submodule where only one Branch has not been hit by the stimulus generated through the testcases used for verification.

The testplan will be augmented to add the testcases suggested by the IOL InterOperability Laboratory [5].

5. LOGICAL SYNTHESIS

The Leonardo Spectrum [7] was used to synthesize the submodule for both ASIC and FPGA. Tables 2 and 3 show the cells count and synthesis results, respectively for the ASIC synthesis using the TSMC 0.35 micron technology [8] and for the Virtex II Pro FPGA.

Cell	References	Total Area
and02	1 x 1	1 gates
and03	2 x 2	3 gates
aoi21	11 x 1	14 gates
aoi22	3 x 1	4 gates
aoi221	2 x 2	4 gates
aoi222	6 x 2	13 gates
buf02	4 x 1	4 gates
dffr	75 x 6	419 gates
dffs_ni	1 x 6	6 gates
inv01	16 x 1	12 gates
inv02	19 x 1	14 gates
mux21	6 x 2	11 gates
mux21_ni	43 x 2	78 gates
nand02	15 x 1	15 gates
nand03	5 x 1	6 gates
nand04	8 x 1	12 gates
nor02_2x	2 x 1	2 gates
nor02ii	10 x 1	12 gates
nor03_2x	5 x 1	6 gates
nor04	10 x 1	15 gates
oai21	16 x 1	20 gates
oai22	1 x 1	1 gates

Cell	References	Total Area
oai222	2 x 2	4 gates
oai32	1 x 2	2 gates
oai422	1 x 3	3 gates
or02	5 x 1	6 gates
or03	5 x 2	8 gates
or04	1 x 2	2 gates
xnor2	5 x 2	10 gates
xor2	5 x 2	11 gates

Table 2. Cells count - TSMC 0.35.

Number of ports :	66
Number of nets :	350
Number of instances :	286
Number of gates :	718

Table 3. Synthesis results - TSMC 0.35.

The Leonardo Spectrum [7] was also used to synthesize the same circuit for the FPGA Virtex II Pro [9].

This FPGA was used to prototype the module and will also be used for testing the device when it gets back from the fab. Tables 4 and 5 present the synthesis results for the targeted FPGA.

Cell	Library	References	Total Area
BUFGP	xcv2p	1 x 1	1 BUFGP
FDC	xcv2p	75 x 1	75 Dffs or Latches
FDP	xcv2p	1 x 1	1 Dffs or Latches
IBUF	xcv2p	39 x 1	39 IBUF
LUT1	xcv2p	3 x 1	3 Function Generators
LUT2	xcv2p	18 x 1	18 Function Generators
LUT3	xcv2p	34 x 1	34 Function Generators
LUT4	xcv2p	96 x 1	96 Function Generators
MUXF5	xcv2p	5 x 1	5 MUXF5
OBUF	xcv2p	25 x 1	25 OBUF

Table 4. Cells count - Virtex II Pro.

Number of ports :	66
Number of nets :	337
Number of instances :	297
Number of references to this view :	0
Number of BUFGP :	1
Number of Dffs or Latches :	76

Number of ports :	66
Number of Function Generators :	151
Number of IBUF :	39
Number of MUXF5 :	5
Number of OBUF :	25
Number of gates :	151
Number of accumulated instances :	297

Table 5. Synthesis results - Virtex II Pro.

6. FUTURE WORK

In order to test the device designed, a test platform will be built using the Virtex II Pro board. The MII ASIC will be placed in between the existing PHY and the FPGA. The MII submodule will be subtracted from the original RTL MAC module.

An external board will be designed to place the MII ASIC and make the necessary connections between the FPGA, the MII ASIC and the PHY. The FPGA will hold the MAC (less the MII) and it will drive signals to the MII ASIC which will then drive signals to the PHY, and vice-versa. When transmitting data, the FPGA will send signals to the MII ASIC which will pass down to the PHY. When receiving data, the MII ASIC will get signals from the PHY, process them and send up to the FPGA.

With this strategy, it will be possible to put the MII ASIC into an actual environment where real Ethernet frames will be sent and received through a real data network. Figure 3 shows the block diagram of the test environment.

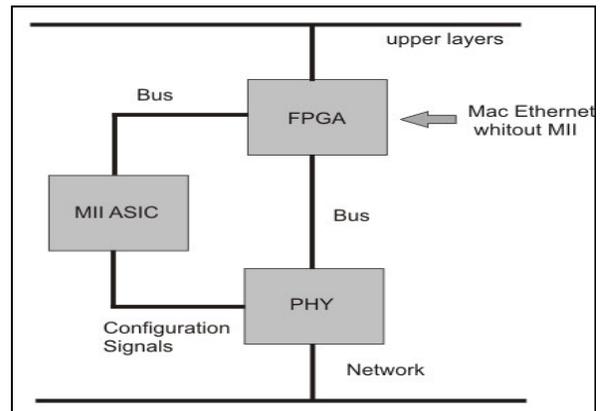


Figure 3. - MII ASIC test environment.

7. CONCLUSIONS

In this paper it was presented a brief description of the design flow with emphasis on the functional verification coverage and the synthesis of a MII module for both FPGA and ASIC. The results showed that a high degree of coverage was achieved and the circuit was successfully synthesized for both FPGA and ASIC.

The next step is to augment the testplan in order to add testcases specified by the IOL and carry the design

through the electrical characterization and layout to then send it to the foundry. At last, the test environment under development will be used to carry out the testing of the resulting circuit.

8. ACKNOWLEDGMENTS

The authors gratefully thank the UNISC support in the form of scholarships and the CEITEC technical support.

9. REFERENCES

- [1] Ciletti, M. D., Advanced Digital Design with the Verilog HDL, Pearson Education, New Jersey, 2003.
- [2] Micheli, G. D., Synthesis and Optimization of Digital Circuits, McGraw-Hill, USA, 1994.
- [3] Wile, B., Goss, J. C. , Roesner, W., Comprehensive Functional Verification - The Complete Industry Cycle, Elsevier, San Francisco, 2005.
- [4] IEEE 802.3 (Institute of Electrical and Electronics Engineers, Inc.), <http://www.ieee.org>.
- [5] IOL InterOperability Laboratory, <http://www.iol.unh.edu/services/testing/fe/testsuites/X>.
- [6] Opencores, <http://www.opencores.org>.
- [7] Mentror Graphics, www.mentor.com.
- [8] Taiwan Semiconductors Manufacturing Company Limited, http://www.tsmc.com/english/b_technology/b_technology_index.htm.
- [9] Xilinx Virtex II Pro , www.xilinx.com.