

# CAD Tool Flow for Variation-Tolerant Non-Volatile STT-MRAM LUT based FPGA

Jeongbin Kim

Dept. of Electrical and Electronic Eng., Dept. of Electrical and Electronic Eng.,  
Yonsei University  
Seoul, Republic of Korea  
+822-2123-7826  
xtankx123@yonsei.ac.kr

Ki Tae Kim

Dept. of Electrical and Electronic Eng., Dept. of Electrical and Electronic Eng.,  
Yonsei University  
Seoul, Republic of Korea  
+822-2123-7826  
ktkim1116@yonsei.ac.kr

Eui-Young Chung

Dept. of Electrical and Electronic Eng., Dept. of Electrical and Electronic Eng.,  
Yonsei University  
Seoul, Republic of Korea  
+822-2123-5866  
eychung@yonsei.ac.kr

## ABSTRACT

Field Programmable Gate Array (FPGA) is a reconfigurable circuit and it is used for various applications such as image processing, digital signal processing and neural network. FPGA adopts a logic circuit called Look-Up Table (LUT) as a basic circuit structure. Commonly used FPGAs have volatile characteristic because it consists of SRAM based LUT that adopts SRAM as a memory cell. Volatile FPGAs have a disadvantage in terms of power management efficiency. Variation-Tolerant Non-Volatile STT-MRAM (VTNV) LUT has been studied for a non-volatile FPGAs and it has unique characteristics that can only operate in the half clock period. Accordingly, VTNV LUT based FPGA cannot operate normally with conventional FPGA CAD tool flow. We propose FPGA CAD (Computer Aided Design) tool flow for VTNV LUT based FPGA with supporting unique characteristic of VTNV LUT, and implement a non-volatile FPGA. Through proposed FPGA CAD tool flow, non-volatile FPGA based on VTNV LUT could operate normally. Because of high parameters of VTNV LUT, experimental results show that power increases by 29% and critical path delay increases by 16%, but it'll be improved sufficiently by future VTNV LUT researches.

## CCS Concepts

- Hardware → Electronic design automation
- Hardware → Reconfigurable logic and FPGAs.

## Keywords

Computer aided design (CAD); field programmable gate array (FPGA); CAD tool flow; non-volatile FPGA;

## 1. INTRODUCTION

Field Programmable Gate Arrays (FPGAs) [1] are reconfigurable circuits that have fast performance of Application Specific Integrated Circuit (ASIC) while have flexibility of Central Processing Unit (CPU) [2]. Recently, FPGA is used for various applications such as image processing, digital signal processing, neural network. FPGA adopts a logic circuit called Look-Up Table (LUT) as a basic circuit structure, and LUTs consist of several memory cells. Commonly used FPGAs consist of SRAM

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ICSCA 2018, February 8–10, 2018, Kuantan, Malaysia

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5414-1/18/02...\$15.00

<https://doi.org/10.1145/3185089.3185134>

based LUT which adopts SRAM as a memory cell. When FPGA shut off, all mapped circuits are erased since SRAM is a volatile memory. This disadvantage is extremely critical for recent devices such as mobile and server which use FPGAs as a co-processor because they can be turned off suddenly.

Spin transfer torque magnetic random access memory (STT-MRAM) is a type of non-volatile memory and it has similar performance to SRAM [3]. To implement a non-volatile FPGA, LUTs that consist of STT-MRAM have been studied. Latch-based STT-MRAM (LBS) LUT [4] has limited functionality because it must operate synchronized with the clock signal. Voltage-divider-based (VDB) LUT [5] is not limited in functionality but has a problem with large static currents. Variation-Tolerant Non-Volatile STT-MRAM (VTNV) LUT [6] is a LUT which solves the problem of the functionality limitations of the LBS LUT and the large static current of the VDB LUT. However, the VTNV LUT has a unique characteristic that can only operate in the half clock period.

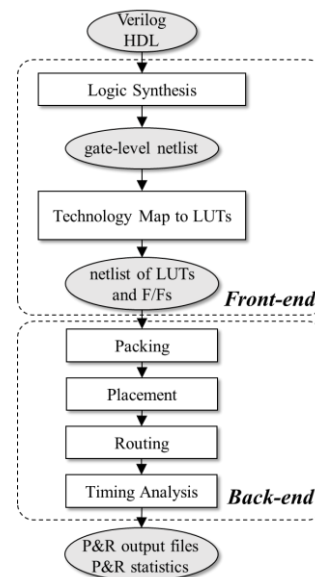


Figure 1. CAD Tool Flow For FPGA

FPGA CAD (Computer Aided Design) tool flow is a sequence of tool that is required to design FPGA circuit. FPGA CAD tool flow is commonly composed as shown in Fig 1. It consists of two parts: Front-end which converts Verilog HDL circuits into a LUT-level netlist, and back-end which mapping LUT-level netlist to the FPGA. More details will be explained in the next section.

As mentioned above, the VTNV LUT has unique characteristics that can only operate in the half clock period. Accordingly, VTNV LUT based FPGA will not operate normally if it designs the circuit through conventional FPGA CAD tool flow. Because conventional FPGA CAD tool flow is based on common LUT and it operates normally irrespective of clock state. Consequently, FPGA CAD Tool Flow based on VTNV LUT should consider the unique characteristics of the VTNV LUT.

In this paper, we propose the FPGA CAD tool flow supporting the VTNV LUT based FPGA. This allows us to design a non-volatile FPGA, which isn't erased even when power is shut down and follows the performance of SRAM LUT based FPGA.

This paper is organized as follows. Section 2 provides backgrounds of FPGA CAD tool flow of FPGA and VTNV LUT. Section 3 describes the proposed FPGA CAD tool flow for VTNV LUT based FPGA. And in section 4, we experiment VTNV LUT based FPGA with our proposed FPGA CAD tool flow.

## 2. BACKGROUND

### 2.1 CAD Tool Flow for FPGA

FPGA CAD tool flow is a tool-chain that allow circuit composed of Verilog HDL to mapping into the FPGA, and it is an essential for FPGA circuit design. It consists of two parts: Front-end and back-end. In this paper, we adopt VTR (Verilog-To-Routing) tool [7] to research the FPGA CAD tool flow, which is widely used for FPGA CAD research. Front-end part includes ODIN II [8] and ABC Tool [9], and back-end part includes VPR Tool [10]. Detailed FPGA CAD tool flow will be explained based on VTR Tool.

Front-end part converts circuit composed of Verilog HDL into LUT-level netlists and consists of the following stages: logic synthesis stage which generates gate-level netlist, and technology mapping stage which yields LUT-level netlist.

Back-end part designs FPGA architecture using LUT-level netlist and consists of following stages: packing stage which integrates LUTs into CLBs (Configurable Logic Block; upper logic units of LUT), placement stage which place each element (i.e. CLB, I/O pad, Memory, etc.) in the FPGA, routing stage which connects each element, and timing analysis stage which determines the clock frequency.

### 2.2 Variation-Tolerant Non-Volatile STT-MRAM LUT

VTNV LUT has a unique characteristic that is different from the common LUT as mentioned in Section 1. This LUT was developed based on VDB LUT and improved the large static current problem of VDB LUT, by supplying power to the half of memory cells.

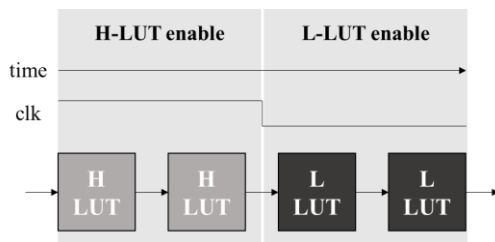


Figure 2. Unique characteristic of VTNV LUT

In consequence of this, VTNV LUT has the characteristic to operate only in the half clock period. It means that input signals of LUT are only propagated to output signal during half clock period. Therefore, as shown in Figure 2, the VTNV LUT is divided into a High-LUT (H-LUT) and Low-LUT (L-LUT) that operates only during high clock period and low clock period.

## 3. CAD TOOL FLOW FOR VTNV LUT BASED FPGA

As mentioned above, VTNV LUT based FPGA will not operate normally with conventional FPGA CAD tool flow. FPGA CAD tool flow specific to VTNV LUT based FPGA is necessary, which support VTNV LUT's unique characteristics. And not only VTNV LUT based FPGA, other FPGAs that have same characteristics with VTNV LUT based FPGA can be designed through this CAD tool flow. We modify technology mapping, packing, placement, timing analysis stages from conventional FPGA CAD tool flow for support VTNV LUT based FPGA.

### 3.1 Technology Mapping

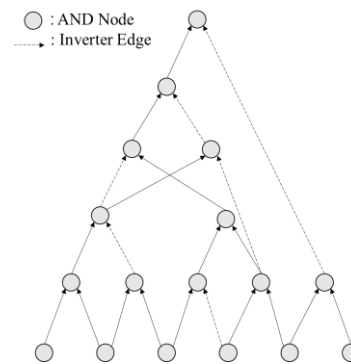


Figure 3. And-Inverter Graph

The main purpose of the technology mapping stage is to generate the LUT-level netlist from gate-level netlist. Technology mapping stage in the VTR Tool converts the gate-level netlist into the And-Inverter Graph (AIG) [11]. As shown in Figure 3, AIG represents circuits by AND gates and inverters. LUTs are created by grouping several nodes in AIG. In conventional technology mapping stage, it is performed with considering delays and areas. There are delay/area optimize mode, and it depends on which factor is considered first.

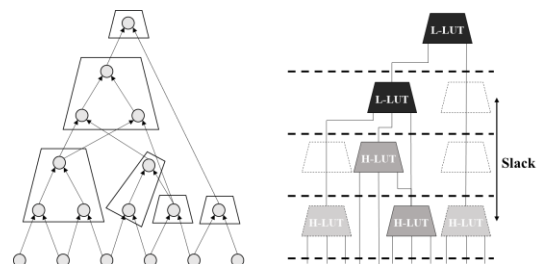
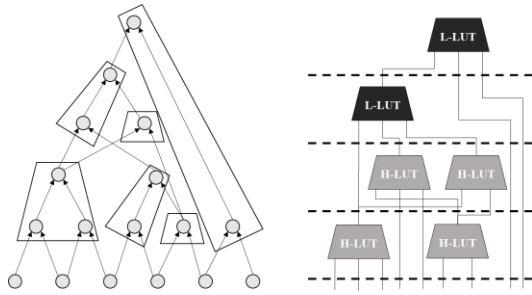


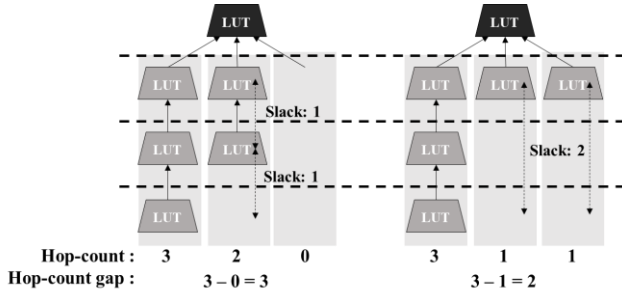
Figure 4. Technology mapping with considering slack



**Figure 5. Technology mapping without considering slack**

In the VTNV LUT based FPGA, the physical ratios of H-LUT and L-LUT are predefined. Improving flexibility of LUT mapping alleviates the physical limitation. The flexibility can be enhanced in the technology mapping stage by enabling more LUTs in both H-LUT and L-LUT.

As shown in Figure 4, slack is number of LUT layer in which a LUT can move. The flexibility of LUT mapping can be quantification through the slack, and it is enhanced through increasing slacks of LUT circuit as can be seen in Figure 4 and 5. We tried to improve the flexibility in the technology mapping stage by increasing slack. But slack is known only after all technology mapping has been completed, it is difficult to consider the slack in the process of technology mapping stage.



**Figure 6. Hop-count and hop-count gap of LUT**

In this paper, we introduce a factor called hop-count gap to increase slack. The hop-count gap represents the largest difference in the hop-count, and the hop-count represents the delay of the circuit connected to the input node of the LUT. In Figure 6, the hop-count for left black LUT is {3, 2, 0} and the hop-count gap is 3. The hop-count for right black LUT is {3,1,1} and the hop-count gap is 2. That is, the slack can be increased by reducing the hop-count gap of black LUT.

Consequently, hop-count gap optimize mode is added into technology mapping stage in the FPGA CAD tool flow for VTNV LUT. It considers hop-count gap first to increase the slack by reducing hop-count gap when grouping AIG nodes.

### 3.2 Packing

Main purpose of conventional packing stage is to integrate LUTs resulting from the technology mapping into CLBs. It proceeds packing to minimize critical path delay. The critical path delay in the packing stage is an approximate value because exact critical path delay can be only calculated after routing stage. In the packing stage, it only considers delay of CLBs for calculate critical path delay.

In the packing stage for VTNV LUT based FPGA, before integrating the CLB to the LUT, mark the H-LUT and L-LUT first

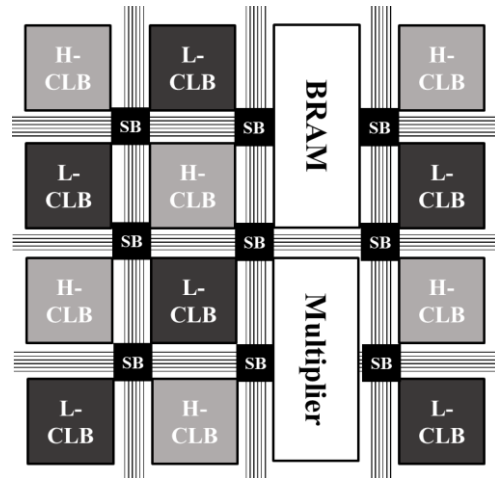
based on the critical path delay calculated at the LUT level. LUTs that can be marked either side are mapped to the side which has smaller number, thereby making the number of H-LUT and L-LUT similar.

After the marking of the H-LUT and the L-LUT completes, the H-LUTs and the L-LUTs are integrated in the H-CLB and L-CLB. H-CLB is a CLB operating only in a high clock period, and L-CLB is a CLB operating only in a low clock period. Except for the above conditions, packing is performed like the conventional packing stage.

### 3.3 Placement

The main purpose of the conventional placement stage is to place the circuit with minimal critical path delay, and proceed in the following order.

- i. Randomly place CLBs in FPGA (Placement 1).
- ii. Calculate the critical path delay of placed architecture. (Delay 1)
- iii. Swap the CLB's position randomly (Placement 2)
- iv. Calculate the critical path delay of swapped architecture. (Delay 2)
- v. Placement proceeds with has smaller delay (Placement 1 or 2), and i ~ iv is repeated the specified number of times.



**Figure 7. VTNV LUT based FPGA Architecture**

The layout of the VTNV LUT based FPGA is island-style architecture [1] as shown in Figure 7. In FPGA CAD tool flow for VTNV based FPGA, placement stage should be done by distinguishing H-CLB and L-CLB. By limiting the place of the CLBs according to each type in step i and iii, the FPGAs can be operated normally by mapping each CLB physically to the location where the H-CLB and L-CLB are existed.

### 3.4 Timing Analysis

The main purpose of the timing analysis stage is to calculate the clock frequency which FPGA operates normally. In the conventional timing analysis stage, the clock frequency is calculated as follows. All paths which send the data from input pad to output pad are explored, and delays of these paths are measured. The longest delay is determined as the critical path delay. The clock frequency is calculated as the equation below.

$$\text{Clock Frequency} = \frac{1}{\text{Critical Path Delay}}$$

It is necessary to calculate the accurate critical path delay for obtaining the operating clock frequency.

In the VNTV LUT based FPGA, each H-CLB and L-CLB must operate entirely during high and low clock period for normal operation. Therefore, timing analysis stage for VNTV LUT based FPGA calculates the clock frequency through following steps. First, the critical path delay is measured for each high CLB circuit and low CLB. Then the clock frequency is calculated using the following equation.

$$\text{Clock Frequency} = \frac{1}{\max\{H\text{-CLB critical path delay}, L\text{-CLB critical path delay}\} \times 2}$$

Through this, timing analysis stage calculates the clock frequency that H-CLB operates fully during high clock period, and L-CLB operates fully during low clock period.

Finally, we build up the VNTV FPGA CAD tool flow which can operate the VNTV LUT based FPGA normally by modifying the 4 stages in the conventional FPGA CAD tool flow as above.

## 4. EXPERIMENT

### 4.1 Experimental Setup

**Table 1. Benchmarks**

Benchmark name	Usage
bm_expr_all_mod	Math calculation
cf_cordic_v_8_8_8	Mathematics processor
diffeq_f_systemC	Differential equation solver
diffeq_paj_convert	Differential equation solver
diffeq2	Differential equation solver
iir_filter	Infinite impulse response filter
mkPktMerge	Packet processing
paj_framebuftop	Image processing

In this paper, we implemented the proposed method in the VTR Tool, so all experiments are proceeded with modified VTR tool. The LUT input size is set to 6, and the number of LUTs per CLB is set to 20. As shown in Table 1, Verilog HDL circuits in the VTR Tool are used as the benchmarks, which are usually mapped in the FPGA.

**Table 2. Parameter of LUTs**

$R_p/R_{ap}$	VNTV LUT				SRAM LUT
	3k/6k	6k/12k	12k/24k	24k/48k	
Delay (ns)	0.99	1.06	1.28	1.73	0.73
Read Power (uW)	36.62	31.45	27.46	25.65	47.19

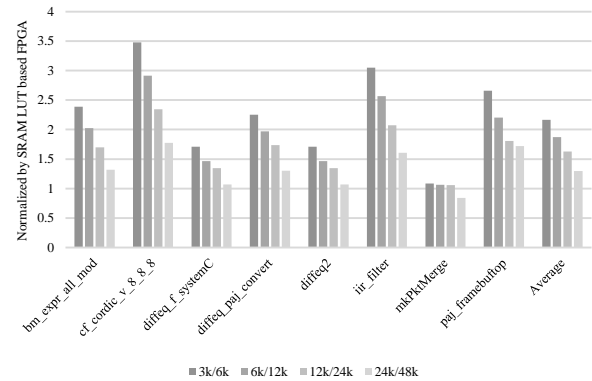
Static Power(uW)	25.39	20.12	15.26	11.25	1.25
------------------	-------	-------	-------	-------	------

Table 2 shows the parameters of each LUTs. Delay, read power, and static power of the SRAM-based and VNTV LUT were obtained from [6].

There is delay/power trade-off according to  $R_p/R_{ap}$  value, so VNTV LUT based FPGA can be configured to optimized for delay or power according to importance. X-axis of Figure 8 and 9 represents the result of each benchmarks, and all experiment results are normalized to SRAM LUT based FPGA.

### 4.2 Experimental Result

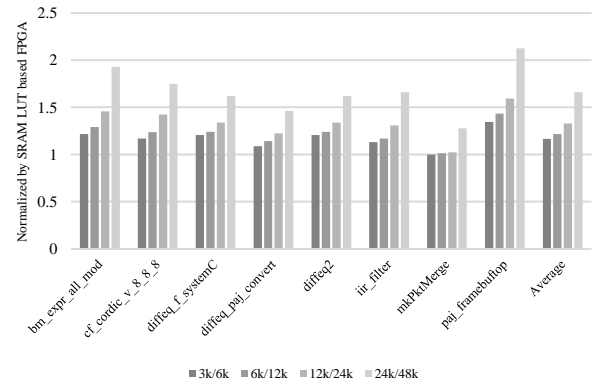
#### 4.2.1 Power of VNTV LUT based FPGA



**Figure 8. Normalized power of VNTV LUT based FPGA**

Figure 8 shows the result of power of VNTV LUT based FPGA normalized by SRAM LUT based FPGA. In terms of FPGA operation, static power has a far greater impact on FPGA power consumption than read power. This is because the transition ratio of conventional circuits is only about 10% to 20%. When considering the LUT parameter, the static power of the VNTV LUT is 9 to 20 times higher than the static power of the SRAM LUT. On the perspective of an entire FPGA, however, the power consumption only increases by 29% to 116% on the average of benchmarks compared to SRAM LUT based FPGA. The finest power result is obtained when the  $R_p/R_{ap}$  value is 24k/48k.

#### 4.2.2 Critical Path Delay of VNTV LUT based FPGA



**Figure 9. Normalized critical path delay of VNTV LUT based FPGA**

Figure 9 shows the result of critical path delay of VTNV LUT based FPGA normalized by SRAM LUT based FPGA. When looking at the value of the LUT parameter, the delay of the VTNV LUT is at least 1.36 to 2.36 times higher than the delay of the SRAM LUT. On the perspective of an entire FPGA, however, there is only a critical path delay increase of 16% to 66% on the average of benchmarks compared to SRAM LUT based FPGA. The finest delay result is obtained when the  $R_p/R_{ap}$  value is 3k/6k.

As a result, for VTNV LUT based FPGAs, power increased by at least 29% and critical path delay increased by at least 16%. This is due to the parameter of VTNV LUT itself is too high compare to SRAM LUT. Since the non-volatile FPGA is implemented through our proposed method, it can be very advantageous in terms of power utilization efficiency. Also, if the delay and power performance of the VTNV LUT are improved through continuous study, the performance of the VTNV LUT based FPGA can be better than that of the conventional SRAM LUT based FPGA.

## 5. CONCLUSION

In this paper, main contribution is to propose FPGA CAD tool flow that can operate VTNV LUT based FPGA normally, and implement the non-volatile FPGA. For FPGA CAD tool flow of VTNV LUT based FPGA, we apply the unique characteristics of VTNV LUT and modify the technology mapping, packing, placement, and timing analysis of conventional FPGA CAD tool flow. Experimental results of VTNV LUT based FPGA show that power increases by 29% and critical path delay increases by 16% and it is because of high parameters of VTNV LUT. As a result, we implement a non-volatile FPGA through proposed FPGA CAD tool flow. If the performance of the VTNV LUT is improved through continuous research, non-volatile FPGAs will have better performance than existing FPGAs.

## 6. ACKNOWLEDGMENTS

This research was supported by SK Hynix and the MOTIE(Ministry of Trade, Industry & Energy) (10080722) and KSRC(Korea Semiconductor Research Consortium) support program or the development of the future semiconductor device.

## 7. REFERENCES

- [1] Betz, V., Rose, J., & Marquardt, A. (2012). Architecture and CAD for deep-submicron FPGAs (Vol. 497). Springer Science & Business Media.
- [2] Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., & Marr, D. (2016, December). Accelerating Binarized Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC. In Field-Programmable Technology (FPT), 2016 International Conference on (pp. 77-84). IEEE.
- [3] Torres, L., Brum, R. M., Cargnini, L. V., & Sassatelli, G. (2013, May). Trends on the application of emerging nonvolatile memory to processors and programmable devices. In Circuits and Systems (ISCAS), 2013 IEEE International Symposium on (pp. 101-104). IEEE.
- [4] Zhao, W., Belhaire, E., Chappert, C., & Mazoyer, P. (2009). Spin transfer torque (STT)-MRAM--based runtime reconfiguration FPGA circuit. ACM Transactions on Embedded Computing Systems (TECS), 9(2), 14.
- [5] Paul, S., Mukhopadhyay, S., & Bhunia, S. (2008, November). Hybrid CMOS-STTRAM non-volatile FPGA: Design challenges and optimization approaches. In Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on (pp. 589-592). IEEE.
- [6] Jo, K., Cho, K., & Yoon, H. (2016, October). Variation-tolerant and low power look-up table (LUT) using spin-torque transfer magnetic RAM for non-volatile field programmable gate array (FPGA). In SoC Design Conference (ISOCC), 2016 International (pp. 101-102). IEEE.
- [7] Rose, J., Luu, J., Yu, C. W., Densmore, O., Goeders, J., Somerville, A., ... & Anderson, J. (2012, February). The VTR project: architecture and CAD for FPGAs from verilog to routing. In Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays (pp. 77-86). ACM.
- [8] Jamieson, P., Kent, K. B., Gharibian, F., & Shannon, L. (2010, May). Odin ii-an open-source verilog hdl synthesis tool for cad research. In Field-Programmable Custom Computing Machines (FCCM), 2010 18th IEEE Annual International Symposium on (pp. 149-156). IEEE.
- [9] Mishchenko, A. (2007). ABC: A system for sequential synthesis and verification. URL <http://www.eecs.berkeley.edu/~alanmi/abc>.
- [10] Betz, V., & Rose, J. (1997, September). VPR: A new packing, placement and routing tool for FPGA research. In International Workshop on Field Programmable Logic and Applications (pp. 213-222). Springer, Berlin, Heidelberg.
- [11] Brummayer, R., Cimatti, A., Claessen, K., Een, N., Herbstritt, M., Kim, H., ... & Soerenson, N. (2007). The AIGER And-Inverter Graph (AIG) Format Version 20070427.