FPGA Static Timing Analysis Enhancement Based on Real Operating Conditions

Marc Alexandre Kacou*†, Fakhreddine Ghaffari†, Olivier Romain†, Bruno Condamin*
*Valeo Siemens eAutomotive, Cergy-Pontoise, France
{marc.kacou, bruno.condamin}.jv@valeo-siemens.com
†ETIS, UMR 8051,Univ Paris Seine, Univ Cergy-Pontoise, ENSEA, CNRS, France
{assi-marc.kacou, fakhreddine.ghaffari, olivier.romain}@ensea.fr

***Abstract*—FPGAs are very sensitive to their operating conditions which can induce runtime errors. To prevent timing errors, FPGA manufacturers propose static timing analysis tools to ensure that the application to be implemented in the FPGA will work correctly at the expected frequency. However, that static timing analysis is corner-based and is thus valid for a set of optimal or recommended operating conditions. In the same time, the real operating conditions can be outside these corners when the FPGA is used in a harsh environment. In this paper, we propose a static timing analysis enhancement technique based on the real operating conditions that the FPGA will encounter. Thus, the static timing analysis can be done outside the predefined corners to ensure that the FPGA application will execute correctly in the real operating conditions. We also present some results showing the accuracy of our method and its application to an automotive application intended to be deployed in an aggressive environment.**

***Index Terms*—Field Programmable Gate Array (FPGA), Static Timing Analysis, Operating conditions, Voltage, Temperature, Runtime Error, Automotive.**

I. Introduction

FPGAs offer a good alternative over ASIC, microcontroller and DSP thanks to a low cost and a fast time-to-market, an ability to be easily reconfigurable and by allowing to effectively parallelize tasks. Hence, their use in the industry covers a wide range of applications in several domains such as aerospace, defense, medical and automotive. Among these applications, some represent an aggressive environment for the FPGA. When these applications are also safety-critical, it is essential for the FPGA to execute the tasks properly.

In contrast to microcontroller and DSP, but like for ASIC development, it is the responsibility of the hardware designers to ensure that the device will work properly at the required frequency. A critical and mandatory step in that sense is the static timing analysis (STA) of the design being implemented. It gives the maximum frequency at which a design will run reliably, without any timing error. STA tools provided by FPGA manufacturers are designed to deal with the effects of Process, Voltage and Temperature variation (PVT) [1] [2] [3] [4] [5]. Process variation comes from the manufacturing of the FPGA itself and takes into account inter-die and intra-die variations due to process deviation. By taking into account the process variation, it ensures the FPGA designer that its design will run safely on any FPGA chip of the same family and speed grade. It is a parameter out of control for an FPGA end user. In contrast, voltage and temperature are parameters that can be controlled by the FPGA end user. Any CMOS device exhibits a voltage and temperature dependency on its internal parameters such as the propagation delay [6]. Since the aim of the timing analysis is to verify that each signal in the design will be synchronized correctly at a given clock frequency, it is important to carefully take into account these variations. For that reason, FPGA manufacturers tools perform static timing analysis in several corners. When STA is successfully passed, the design is intended to work in any operating conditions encompassed by the defined corners. For automotive applications, the design is generally guaranteed to be timing error free in the temperature range from -40°C to +125°C and for a supply voltage in a range around the nominal voltage, typically $\pm 5\%$ of the nominal voltage. With technology downscaling, that voltage range is now about tens of millivolts. Hence, once the real operating conditions of the FPGA can be sometimes outside these ranges, the correct execution of the design can no longer be guaranteed by the standard STA tools. Voltage variation can occur when the FPGA is used in an aggressive electromagnetic environment such as in electric vehicles applications [7] or due to a drift of the FPGA voltage regulator output over time.

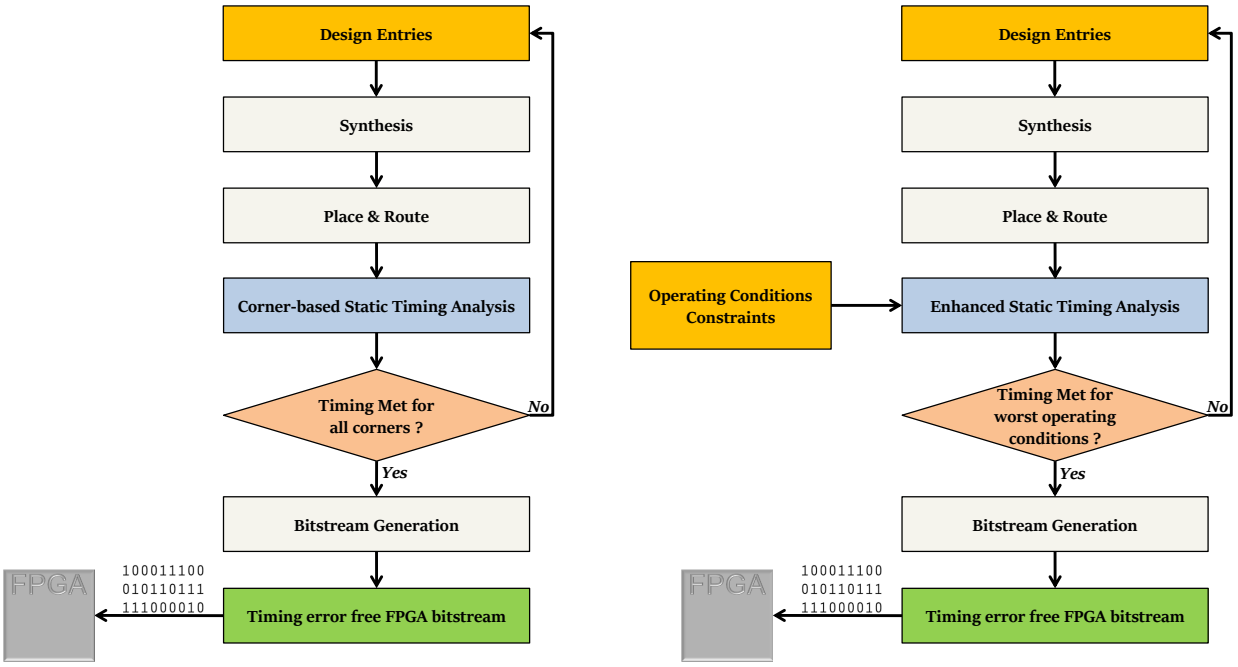
In this paper, a timing analysis enhancement technique based on the real operating conditions is proposed to compute the maximum allowable frequency at which the design can work without any timing error. The temperature is still limited to the standard ranges but the voltage is no more limited to a small range around the nominal as explained before, in order to take into account the effect of unwanted voltage variations.

Section II describes the principle of the technique and its implementation as an addition to the FPGA manufacturers STA tools. In Section III, the measurement of some FPGA blocks dependency to voltage and temperature is presented as it is required for taking into account extended operating conditions. Finally in Section IV, validation results of the proposed method are presented as well as its application to an automotive design.

II. Timing Analysis Enhancement

The standard compilation flow of an FPGA design is presented in Fig.1a. From the HDL (Hardware Description Language) description of an application, a synthesis is performed

978-1-5386-1127-2/17/$31.00 ©2017 IEEE



(a) Standard Flow.

(b) Flow with enhanced STA.

Figure 1. FPGA compilation flow.

to translate that description into Boolean operations. Once the synthesis is achieved, a netlist describing the application is obtained. Then, the FPGA place-and-route tool maps the application netlist onto the FPGA blocks. Finally the static timing analysis is executed to ensure that the achieved placement guarantees the user timing constraints. The bitstream is generated and can then be downloaded into the FPGA to execute the application. The FPGA designer has to modify the design entries and perform these steps again as long as the timing requirements are not met.

Any synchronous design can be represented by Fig.2. It consists in input paths, register-to-register paths and output paths. The timing analysis of such a design boils down to ensure that every path is correctly synchronized at the user-defined clock frequency. Each path delay depends on the voltage and the temperature and as described in [7], operating conditions such as electromagnetic field can induce delay variation inside the FPGA by altering the FPGA supply voltage. This delay variation can then lead to timing errors if not considered during the timing analysis of the corresponding design.

The idea of our proposed method is to enhance the timing analysis of the application by taking into account the real operating conditions rather than using the standard corner-based analysis as presented in Fig.1b. It is implemented as an extension of the corner-based STA tool and can be fully integrated in the compilation flow. It is based on the results of the timing analysis performed by the standard STA tool which are improved by adding the delay variation associated with the user selected operating conditions for each path of the design.

To do that, it is fundamental to model the FPGA dependency to voltage and temperature. For a given FPGA, all used blocks should be analyzed to evaluate and model the corresponding voltage and temperature dependency. Fig.3 presents the steps to compute the new timing results and consists, for each path, in: 1) Get the STA-based nominal delay T_{pnom}. 2) Identify the different blocks of the path. 3) Map the operating conditions

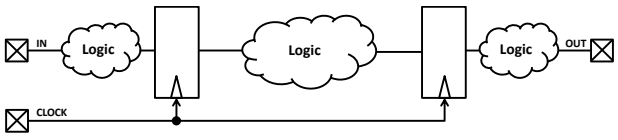


Figure 2. Simple synchronous design.

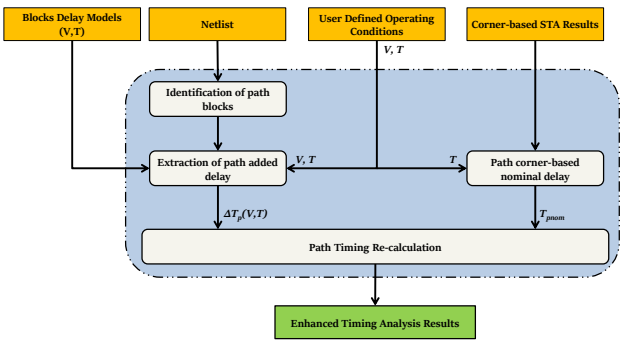


Figure 3. The different steps to calculate the new path timing.

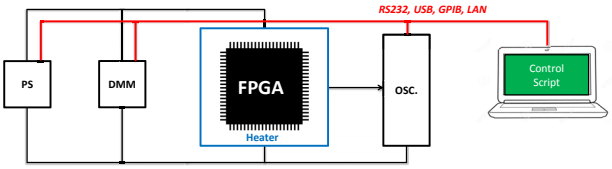


Figure 4. The measurement platform used to measure the FPGA voltage and temperature dependency.

on each block model in order to get the corresponding added delay $\Delta T_p(V,T)$. 4) Compute the new path delay by adding the sum of each $\Delta T_p(V,T)$ to T_{pnom}. Standard STA-based results are used in our methodology in order to take into account, inter-die and intra-die variation to guarantee that it can be used for any FPGA chip of the same family and speed grade, and not only the ones used to establish the model. In fact, once FPGA chips are fabricated, they are tested and classified in speed bins to determine the slowest and the fastest ones as process variation lead to performance variations. Variations even happened in devices of the same speed-grade. To reduce the computation effort, FPGA manufacturers use the slowest and fastest chips to perform delay characterization and extract the timing models used for their corner-based STA [1]. There are timing models for worst-case to perform setup and recovery analysis, and timing models for best-case to perform hold and removal analysis. Hence, any chip of intermediary speed will properly execute the design once the timing analysis is passed for both the worst-case and best-case. Thus, taking these results into account in our methodology gives our enhanced timing analysis, the same coverage of all FPGA chips of same family and speed grade, whatever slow or fast they are.

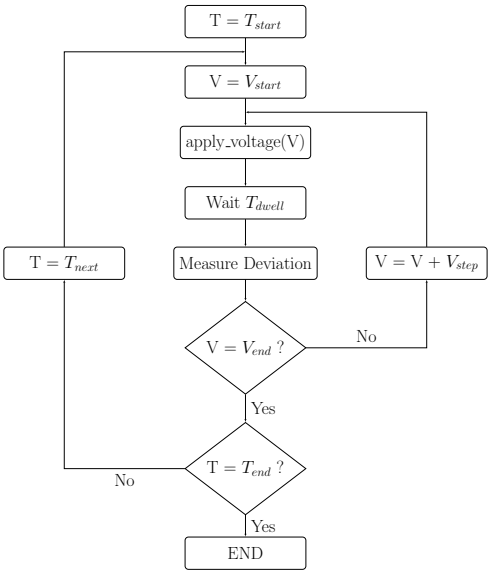


Figure 5. The different steps to measure the voltage and temperature dependency of the FPGA.

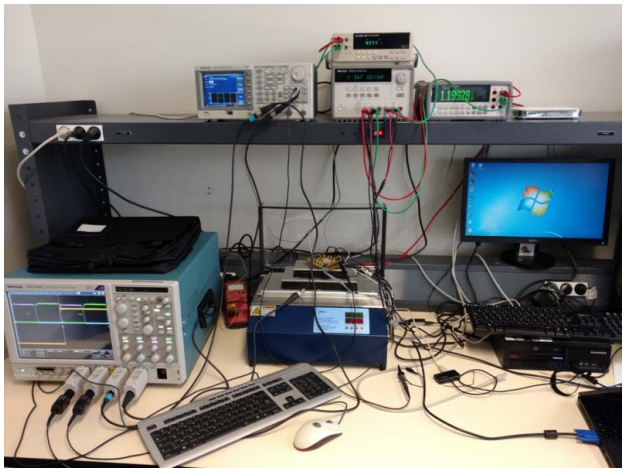


Figure 6. Measurement setup.

Table I
LIST OF THE INSTRUMENTS USED FOR THE MEASUREMENT PLATFORM.

Type	Reference
Digital Multimeter	Keysight 34450A
Power Supply	HP E3631A
Oscilloscope	Tektronix MSO 70404C
Heater	TECA AHP1200 CAS

III. Measurement of FPGA Voltage and Temperature Dependency

A. Principle

Inside an FPGA, there is a network of highly configurable cells, commonly called Configurable Logic Blocks (CLB). Each basic CLB contains at least one Look-Up Table (LUT), able to implement any combinatorial logic function and one D-type Flip-Flop to implement any sequential logic function. It is also common to have embedded DSP blocks allowing engineers to perform complex computation faster, as well as memory blocks (BRAM) and Phase-Locked Loop (PLL) blocks.

As previously explained, depending on the environment, the function of these blocks may be affected. In order to quantify this impact, the measurement platform presented in Fig.4 has been developed to measure the deviation of an FPGA internal parameters. It consists in varying the temperature and the voltage of the FPGA in order to monitor and measure the deviation. By measuring that deviation as a function of the temperature and the voltage, it is then possible to link an operating conditions set to a deviation point for each FPGA blocks, hence determine the effect of this operating condition set on the design implemented in the FPGA.

B. Measurement Setup

The measurement platform and the associated measurement sequence are presented in Fig.6 and Fig.5. A programmable power supply is used to apply a voltage to the FPGA. That voltage is applied in closed-loop with a digital multimeter reading back the applied voltage in order to guarantee that the

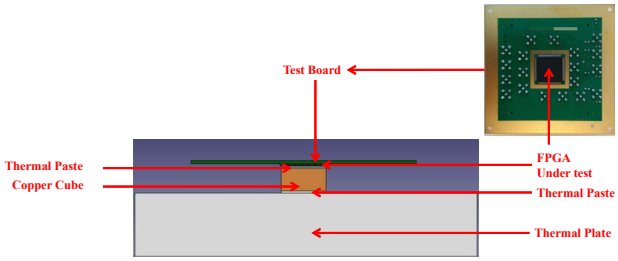


Figure 7. FPGA heating mechanism.

applied voltage is the desired one. Depending on the deviation to be measured, a design is implemented inside the FPGA with the output corresponding to the deviation to be quantified. Then an oscilloscope is used to measure the output of the design, hence the deviation inside the FPGA. The voltage is applied in different steps from the start voltage to the end voltage. For each step, it is maintained during a configurable dwell time, after which the deviation measurement is performed by the oscilloscope. After each voltage sweep, the temperature is changed and the voltage sweep takes place again. This setup has been realized with the instruments presented in Table I. The heater here is a thermoelectric plate able to maintain a stable temperature with an accuracy of 0.1°over a range from -50°to +150°. A copper cube is used to transfer the plate temperature to the FPGA through thermal paste to ensure good thermal conductivity as shown in Fig.7. The power supply used is able to apply voltage with an accuracy of 0.1 mV. A control script is used to automatically perform the voltage sweep while the temperature is set manually or via a dedicated software. According to this setup, we measured the deviation of two internal blocks of an FPGA commonly used in motor control application.

C. Measurement of LUT deviation versus voltage and temperature

In order to measure the deviation of the LUT versus voltage and temperature, the design presented in Fig. 8 has been implemented in the FPGA. As the CLBs inside the FPGA on the board are clustered into 1395 groups, the test design consists in a square wave feeding a chain of 1394 LUTs implementing inverting logic gates. The delay is measured by another LUT implementing a XOR gate which will have its output high as long as the chain output logic level differs from

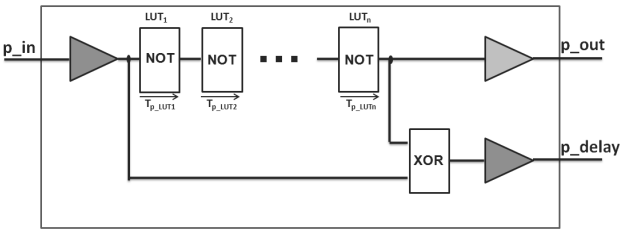


Figure 8. FPGA design used to measure the propagation delay variation.

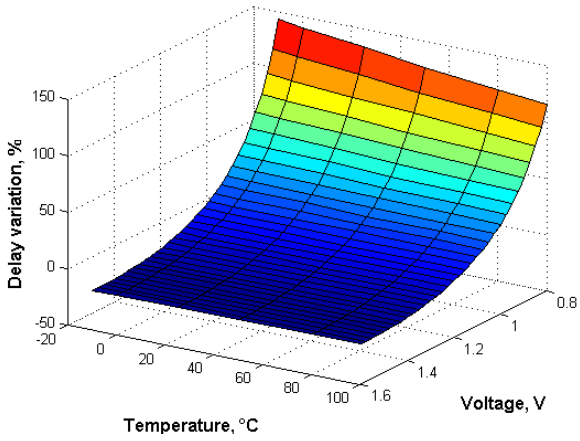


Figure 9. LUT propagation delay variation as a function of voltage and temperature.

the input one. Each LUT is mapped to a CLB cluster, resulting in the chain being uniformly distributed throughout the FPGA surface to take into account the on-chip variation effects.

The nominal power supply voltage of the FPGA under test is rated at 1.2 V and an internal mechanism monitors this voltage. As soon as it goes below a trip point, the FPGA freezes until the voltage goes back to an acceptable level and then restarts. That trip point has been measured at approximately 0.8 V. Hence, the measurements have been done from 0.8 V to 1.6 V by step of 25 mV to have a symmetrical variation around the nominal voltage and a good resolution. The dwell time has been fixed at one second.

Measurements have been done at temperature set to -20°, 0°, 25°, 50° and 100°. Fig.9 show the results of these measurements where the represented delay variation is the variation of the measured delay with respect to the one measured at nominal voltage for each temperature. It indicates how the nominal delay is impacted due to voltage and temperature. It also clearly exhibits the previously explained dependencies of the propagation delay to voltage and temperature. The propagation delay is inversely proportional to the voltage and proportional to the temperature.

D. Measurement of Multiplier deviation versus voltage and temperature

The design used to measure the variation of the multipliers propagation delay is shown in Fig.10 and is based on the

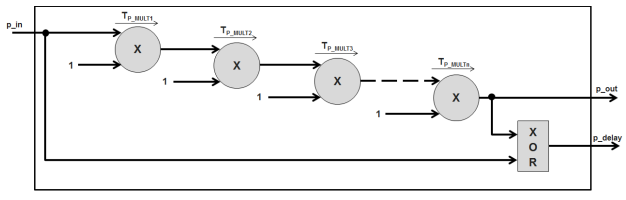


Figure 10. FPGA design used to measure the propagation delay variation of FPGA multipliers.

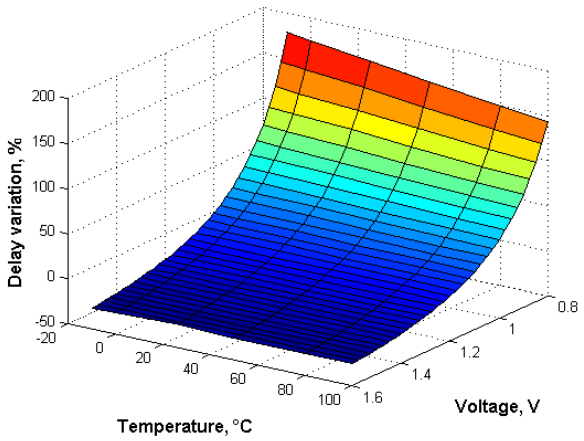


Figure 11. Multipliers propagation delay variation as a function of voltage and temperature.

same principle previously used to measure the LUT variation. A square wave signal feeds a chain of multipliers which all have one input operand set to 1. Hence, The result of the multiplication is always the value of the input square wave, just delayed by the multiplier propagation delay. A XOR gate compares the input and output of the chain with the time at logical one representing the multiplier chain propagation delay, which is then measured by the oscilloscope. The totality of the FPGA multipliers are used in the chain. The same voltage and temperature sweeps as the one used to measure the LUT variation have been applied and the results are given in Fig.11.

IV. Application

To validate our contribution, we use the design presented in Fig12. It consists in an input signal toggling each clock cycle and feeding a 1394-buffer chain before being captured by the output register. From the standard STA results, the computed maximum frequencies are 1.25 MHz and 1.34 MHz for worst case at respectively 125°and -40°and 2.75 MHz for the best case.

With our proposed method, we were able to compute the maximal frequency in several operating conditions, not limited to the subset used by standard STA tool. The results obtained and the comparison with the measured maximal frequencies are given in Fig.13 and show the different values of the

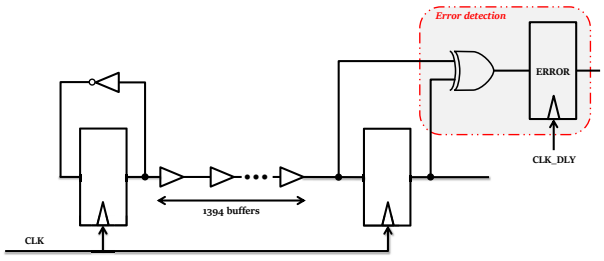


Figure 12. Validation circuit.

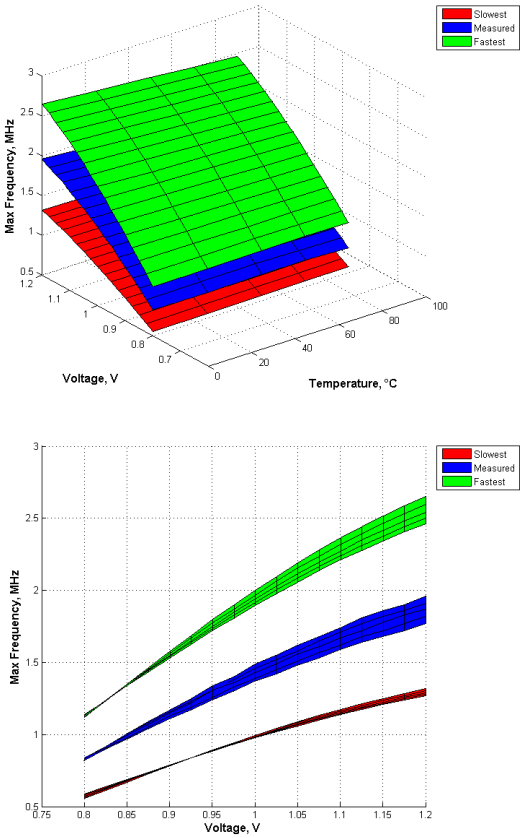


Figure 13. Enhanced STA and measured maximal frequency as a function of voltage and temperature. Top) 3D View - Bottom) Max. Frequency vs voltage view.

maximal frequency as a function of the operating conditions. To perform the measurement of the real maximal frequencies, the clock frequency has been varied until an error is detected with the last frequency before the error being the maximal frequency. The timing error detector described in [8] has been used for error detection and corresponds to the error detection block in Fig.12. The green surface corresponds to the timing analysis for the fastest chip while the red one represents the slowest chip. The blue surface is the measured maximal frequency on an FPGA chip and is between the fast and slow timing analysis. It shows the importance of using the STA results to take into account inter-die variations in order to cover all FPGA chips of the family and speed-grade.

Then, as a concrete example of use of our technique, we perform an enhanced timing analysis for the application presented in Fig.14 which represents a subset of a motor control application. It is a function used to apply pulse-width modulation (PWM) signals to an electric motor and is composed of two blocks. The first one generates the PWM signal according to the ratio input which is the duty cycle. The second block is used to shape the PWM signals in order to be applied on the two power switches of the half-bridge. The shaping operation consists in adding a dead-time between

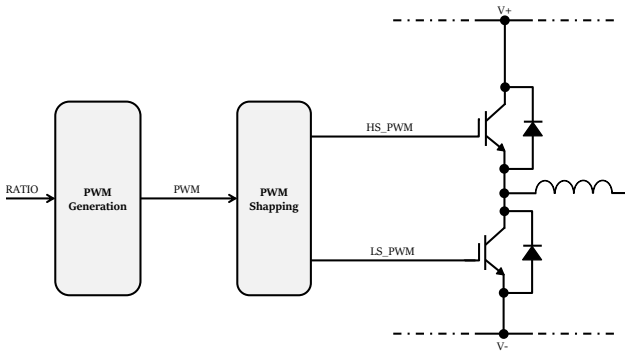


Figure 14. PWM block used to apply the motor control signals.

the top and the bottom PWM signals to avoid cross-conduction resulting in a short-circuit of the half-bridge which can results in critical damages.

The enhancement timing analysis results are shown in Fig.15 where the green surface corresponds to the fastest chip and the red one to the slowest chip. For example, while the maximum frequency is at least 200 MHz at nominal voltage, it is reduced to around 180 MHz when the voltage drops by 100 mV. Should the operating conditions generates 100 mV drop in the FPGA voltage, running the design at 200 MHz as recommended by the STA tool will lead to timing errors in this case. By performing our enhanced timing analysis, the designer would be aware of this situation and could take the necessary actions.

V. Conclusion

In this paper, a timing analysis enhancement technique has been presented. Contrary to standard timing analysis based on corners, thus limiting the timing analysis to a subset of operating conditions, it gives the possibility to perform the timing analysis depending on the real operating conditions that the FPGA will encounter. It is important to be able to perform that timing analysis using the real operating conditions as highly aggressive environment can invalidate the standard timing analysis results, leading potentially to a design not working in the real operating conditions. For that, it is necessary to measure and model the delay variation of each FPGA blocks regarding the applied voltage and temperature. We proposed a method starting with the measurement of these deviations, to their use in the enhancement of the timing analysis. It required a few steps and is thus, integrable into an industrial FPGA development process for applications intended to be used in such aggressive environment.

Acknowledgment

The authors acknowledge the support of General Council of Val dOise (CGVO) and the urban community (CA) of Cergy-Pontoise, under Grant Agreement number 14RETID382 (EXELA-VO project).

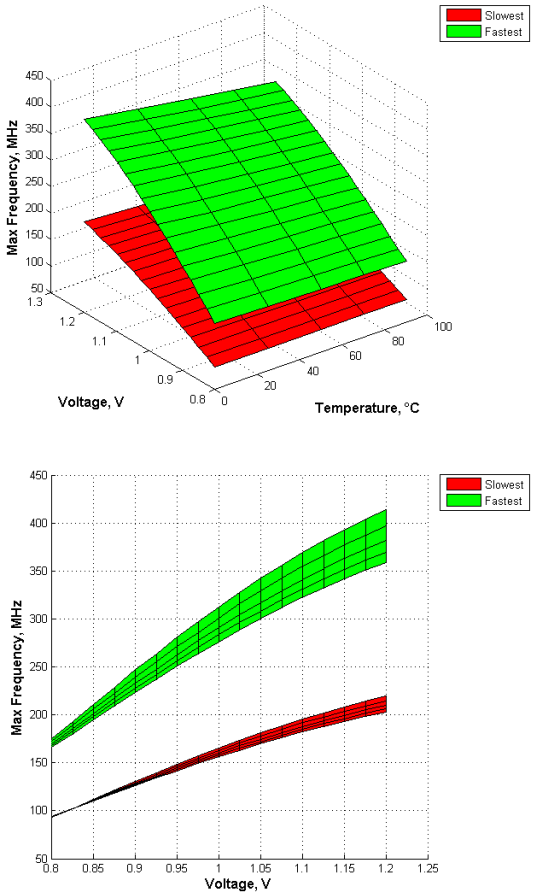


Figure 15. Enhanced timing analysis for an automotive application. Top) 3D View - Bottom) Max. Frequency vs voltage view.

References

[1] *Guaranteeing Silicon Performance with FPGA Timing Models*, Wp-01139-1.0 ed., Altera, August 2010.

[2] *System Design with Advance FPGA Timing Models*, Wp-01213-1.0 ed., Altera, February 2014.

[3] *Xilinx Constraints Guide*, Ug625 (v. 14.5) ed., Xilinx, April 2013.

[4] *Vivado Design Suite User Guide - Design Analysis and Closure Techniques*, Ug906 (v2017.1) ed., Xilinx, April 2017.

[5] *Timing Closure*, Lattice Semiconductor, October 2013.

[6] A. Sassone, A. Calimera, A. Macii, E. Macii, M. Poncino, R. Goldman, V. Melikyan, E. Babayan, and S. Rinaudo, "Investigating the effects of inverted temperature dependence (itd) on clock distribution networks," in *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2012, pp. 165–166.

[7] M. A. Kacou, F. Ghaffari, O. Romain, and B. Condamin, "Influence of high-power electric motor on an fpga used in the drive system of electric car," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 4796–4801.

[8] S. Valadimas, A. Floros, Y. Tsiatouhas, A. Arapoyanni, and X. Kavousianos, "The time dilation technique for timing error tolerance," *IEEE Transactions on Computers*, vol. 63, no. 5, pp. 1277–1286, May 2014.