



Nunez-Yanez, J. L., & Farhadi Beldachi, A. F. (2014). Run-time power and performance scaling in 28 nm FPGAs. *IET Computers and Digital Techniques*, 8(4). <https://doi.org/10.1049/iet-cdt.2013.0117>

Peer reviewed version

Link to published version (if available):  
[10.1049/iet-cdt.2013.0117](https://doi.org/10.1049/iet-cdt.2013.0117)

[Link to publication record in Explore Bristol Research](#)  
PDF-document

This paper is a postprint of a paper submitted to and accepted for publication in *IET Computers & Digital Techniques* and is subject to Institution of Engineering and Technology Copyright. The copy of record is available at IET Digital Library

## University of Bristol - Explore Bristol Research

### General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:  
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

# Run-time power and performance scaling in 28 nm FPGAs

Arash Farhadi Beldachi <sup>1</sup>, Jose L. Nunez-Yanez <sup>2</sup>

*Department of Electronic Engineering*

*University of Bristol, UK*

<sup>1</sup> arash.beldachi@bristol.ac.uk

<sup>2</sup> j.l.nunez-yanez@bristol.ac.uk

***Abstract***-The ability of scaling power and performance at run-time enables the creation of computing systems in which energy is consumed in proportion of the work to be done and the time available to do it. These systems favour active energy-efficient states in which useful computation is performed at low energy instead of using inactive energy savings modes that incur large latency and energy penalties to enter and exit modes in which the system is halted. This is particular useful in servers that spend most of their time at around 30% utilization and are rarely fully idle or at maximum utilization. A feature of an energy proportional computing system is that it must exhibit a wide dynamic range with multiple levels of energy and performance available. In this context this paper investigates how these levels can be obtained in commercially available state-of-the-art 28 nm FPGAs and characterizes its benefits. Adaptive voltage and frequency scaling is employed to deliver proportional performance and power in these FPGA devices. The results reveal that the available voltage and frequency margins create a large number of performance and energy states with scaling possible at run-time with low overheads. Power savings of up to 64.98% are possible maintaining the original performance at a lower voltage.

## 1. Introduction

Energy and power efficiency in FPGAs has been estimated to be up to one order of magnitude worse than in ASICs [1] and this limits their applicability in energy constrain applications. According to device vendors recent 28 nm FPGAs consume 50% lower power than previous generations [2] and this contributes to close this power gap. Additional power savings are possible if FPGAs can make use of techniques such as Adaptive Voltage Scaling (AVS) which results in significant reduction of the dynamic and static power by dynamically adjusting voltage and frequency in a closed-loop configuration. AVS is a popular power-saving technique that enables a device to regulate its own voltage and frequency based on workload, fabrication, and operating conditions and compares favourably with open-loop DVFS (Dynamic Voltage and Frequency Scaling). Our previous work [3] presented a novel design flow and IP library that enable the integration of closed-loop variation-aware adaptive voltage scaling in commercial FPGAs. This approach adapts the operational point over a wide range of voltage and frequency levels at run-time adapting to temperature, process and workload changes automatically. The investigation results were based on a 65nm Virtex-5 device and reveal that although the device has not been validated by the manufacture at below nominal voltage operational points; savings approaching one order of

magnitude are possible by exploiting the margins available in the chip. For this adaptive voltage scaling system to be beneficial there must be performance and voltage margins in the device that can be exploited. In this paper we investigate the presence of these margins in state-of-the-art FPGA devices manufactured in a 28 nm process maintaining other aspects of the system as described in our previous work [3]. The contributions of this work can be summarised as follows:

1. We introduce a low overhead IP Core that controls the system voltages using the PMBus (Power Manager BUS) standard and which can be employed in an adaptive voltage scaling system.
2. To the best of our knowledge, this is the first work which investigates the run-time power and performance scaling capabilities of 28 nm FPGAs and shows its benefits.

This work could be applied to high-performance computing systems based on FPGAs that do not require or cannot tolerate working at maximum levels of performance constantly. This could be similar to modern microprocessors that include a Turbo mode that must make sure that thermal limits are not exceeded. In this case this technology could use data from temperature sensors to locate frequency and voltage points that ensure safe and stable operation. The concept of trading performance and energy as demonstrated in this work can benefit many applications. For example, financial computing for low-latency trading requires responses of just fractions of a second and a configuration set at maximum voltage and frequency will be the most suitable in this scenario. Clock gating could be used to reduce temperatures when new operations are not required while transitions to active states possible in a single clock cycle. On the other hand, background calculations happening with a closed market or based on medium-frequency trading approaches will benefit from a different configuration points focused on energy efficiency at a reduce voltage and frequency.

The rest of the paper is structured as follows. Section 2 describes related work. Section 3 presents the voltage and frequency scaling IP cores and test platform architecture. Section 4 explores the performance and power margins available in 28 nm FPGAs. Finally, section 5 presents the final conclusions and future work.

## **2. Previous works**

In this section we review the related work in the area of FPGA power optimization. In order to identify ways of reducing the power consumption in FPGAs, some research has focused on developing new FPGA architectures implementing multi-threshold voltage techniques, multi-V<sub>dd</sub> techniques and power gating techniques [4-8]. Other strategies have proposed modifying the map and place&route algorithms to provide power aware implementations [9-11]. This related work is targeted towards FPGA manufacturers and tool designers to adopt in new platforms and design environments. On the other hand, a user level approach is proposed in [12]. A dynamic voltage scaling strategy for commercial FPGAs that aims to minimise power consumption for a given task is presented in their work. In this methodology, the voltage of the FPGA is controlled by a power supply that can vary the internal voltage of the FPGA. For a given task, the lowest supply voltage of operation is experimentally derived and at run-time, voltage is adjusted to operate at this critical point. A logic delay measurement circuit is used with an external computer as a feedback control input to adjust the internal voltage of the FPGA (VCCINT) at intervals of 200ms. With this approach, the authors demonstrate power savings from 4% to 54% from the VCCINT supply. The experiments are performed on the Xilinx Virtex 300E-8 device

fabricated on a 180nm process technology. The logic delay measurement circuit (LDCM) is an essential part of the system because it is used to measure the device and environmental variation of the critical path of the functionality implemented in the FPGA and it is therefore used to characterise the effects of voltage scaling and provide feedback to the control system. This work is mainly presented as a proof of concept of the power saving capabilities of dynamic voltage scaling on readily available commercial FPGAs and therefore does not focus on efficient implementation strategies to deliver energy and overheads minimisation. A comparable approach also based in delay lines is demonstrated, by the authors in [13]. A dynamic voltage scaling strategy is proposed to minimise energy consumption of an FPGA based processing element, by adjusting first the voltage, then searching for a suitable frequency at which to operate. Again, in this approach, first the critical path of the task under test is identified, and then a logic delay measurement circuit is used to track the critical point of operation as voltage and frequency are scaled. Significant savings in power and energy are measured as voltage is scaled from its nominal value of 1.0V down to its limit of 0.6V. Beyond this point, the system fails. Xilinx has also investigated the possibility of using lower voltage levels to save power in their latest family implementing a type of static voltage scaling in [14]. The voltage identification bit available in Virtex-7 allows some devices to operate at 0.9 V instead of the nominal 1 V maintaining nominal performance. During testing, devices that can maintain nominal performance at 0.9 V are programmed with the voltage identification bit set to 1. A board capable of using this feature can read the voltage identification bit and if active can lower the supply to 0.9 V reducing power by around 30%. This is a static configuration that maintains the original level of performance and takes place during boot time in contrast with the dynamic approach investigated in this paper.

In-situ detectors located at the end of the critical paths remove the need for delay lines. This technology has been demonstrated in custom processor designs such as those based around ARM Razor [15]. Razor allows timing errors to occur in the main circuit which are detected and corrected reexecuting failed instructions. The latest incarnation of Razor uses an optimized flip-flop structure able to detect late transitions that could lead to errors in the flip-flops located in the critical paths. The voltage supply is lower from a nominal voltage of 1.2V (0.13 $\mu$ m CMOS) for a processor design based on the Alpha microarchitecture observing approximately 33% reduction in energy dissipation with a constant error rate of 0.04%. The Razor technology requires changes in the microarchitecture of the processor and it cannot be easily applied to other non-processor based designs. It also uses utilizes a specialized flip-flop. Our work in [3] presents the application of in-situ detectors to commercial FPGAs that deploy arbitrary user designs. The presented approach removes the need of delay lines as done previously by the authors in [13] increasing the system robustness and efficiency. Additionally, it only uses the technology primitives already available in the FPGA and it does not require chip fabrication or redesign.

In this paper we extend the work of [3] by presenting the additional blocks required to regulate voltage and frequency at run-time using state-of-the-art devices and leveraging the availability of the PMBus in off-the-shelf FPGA boards. In addition, we investigate the run-time power and performance scaling in 28nm devices and compare it with the work in [3] based on 65 nm FPGAs.

### **3. IP Cores and test platform architecture**

A key point in this research is that many modern FPGA boards include Power Management Bus (PMBus) Controllers. The PMBus is an open standard power management protocol that facilitates the communication

with power converters and other devices in a power system [16]. This technology means that software or hardware running in the device have access to a controllable power supply. This is the case with the latest evaluation kits (such as the KC705, VC707, ZC702) for Xilinx series 7 FPGAs that use the Texas Instrument (TI) UCD92xx PMBus controller. The TI UCD92xx series [17] are a family of digital power controller which supports a wide range of commands that allow an external host to configure, control, and monitor the controller through an I2C electrical interface using the PMBus command protocol.

These evaluation kits offer two methods to communicate with the PMBus controller [18]. The first method employs the Fusion Digital Power Designer software package provided by TI [19]. This software package has several tools that are able to communicate with the UCD92xx series of controllers from a Windows-based host computer. This software package requires the use of a USB Interface Adapter EVM [20] to connect the PMBus (I2C) interface of the UCD92xx controller and the USB port in the host computer. The second method consists in using the PMBus (I2C) interface which is available on the boards. This is a more complex method since it requires creating custom code on the device to read and write properly formatted PMBus and UCD92xx commands. TI UCD92xx PMBus Command Reference Manual and the industry standard PMBus Specification for UCD92xx command codes, data formatting, and PMBus protocol are available on [21] and [22], respectively to guide the designer in this task. We have selected the second method because we need to access the PMBus interface internally to scale the voltage dynamically and autonomously.

We have created two hardware units to have full control of the voltage and frequency in the system and these are described in the next two sections:

#### ***A. Dynamic Voltage Scaling unit***

Figure 1 shows the Dynamic Voltage Scaling (DVS) unit architecture. The DVS unit has three main components which are a MicroBlaze processor (MB); a register file implemented using a Dual-Port RAM (DPRAM) and an I2C IP core. These components are connected to a local AXI bus. The DVS unit has full configuration and monitoring capabilities of the power rails connected to the PMBus. The DPRAM is used to receive the commands from the system processor. The commands control and record power and voltage values. The MB is responsible for the execution of the commands, communicating with the PMBUS via the I2C IP core and writing the results to the DPRAM. The need for a MB processor is mainly due to the relatively complexity of I2C communications that means that a state machine implementation will be complex to design and maintain for different boards with slight PMBus implementation differences. Although using a simpler core such as a PicoBlaze could be an alternative, code size limitation could be a problem since it is possible to monitor and configure many parameters related to the main core in the processing subsystem, the FPGA fabric and the external DDR memories. The initialization, configuration and monitoring code is written in C and compiled into a .elf file using the standard Microblaze compiler. The DVS core is controlled with commands which are issued by system processor. A command has 32 bits and contains six parameters as it can be seen in Figure 2. Table 1 presents the details of the commands and parameters. Setting Action0 to 1 indicates that there is a new task to do for DVS IP Core. The Read/Write field indicates if the task is a monitoring or a voltage scaling task.

When the task is monitoring, Read (PL (programmable logic), MEM) and Read (V, I, P) determine which power supply (PL and Memory) and which parameter (Voltage, current and consumed power) are selected to monitor.

The reading voltage, current and power values will be recorded in address offsets 0x1, 0x2 of the DPRAM. The reading parameters and address offsets in the DPRAM can be changed or modified depending on the user requirements.

When the task is voltage scaling, the DVS IP Core scales the voltage to the value written in the Voltage value field. The scaling voltage range is from 650mV to 1V and from 1V down to 650 mV. The IP Core is designed to maintain the voltage in this range to avoid damaging or cutting off the power supply of the board. This means that the IP core will automatically reject commands that indicate a voltage value out of these ranges.

When a monitoring/ voltage scaling task completes, the MB will clear the command in the DPRAM and set the Action1 to 1 to inform that the task has finished to the system processor.

We have employed a Xilinx VC707 evaluation board in this work which uses a Xilinx Virtex 7 XC7VX485T device.

Table 2 shows the complexity of the DVS unit components after implementation in the XC7VX485T device. As it can be seen in this table, this unit is area efficient and it only consumes a small fraction of the available resources.

To help the debugging of the system five error report codes have been considered for the DVS unit. The list of the error codes can be seen in

Table 3. When one of the errors is detected, the MB will clear the command in the DPRAM and set the Action1 to the related error code in this table to inform that there is an error to the system processor.

Table 1 - DVS control commands

<b>Parameter</b>	<b>Related operation</b>	<b>Parameter value</b>	<b>Description</b>
Read/Write	Read/Write	Read/Write=0	The IP core will read the voltage/ current /consumed power of the PL/ Memory
		Read/Write=1	The IP core will scale the PL voltage (VCCint )
PL, MEM	Read	Read(PL, MEM)=0	The PL is selected to monitor its voltage/ current / power
		Read(PL, MEM)=1	The Memory is selected to monitor its voltage/ current / power
V, I, P	Read	Read(V, I, P)=0	The voltage of the PL/PS/Mem is selected to monitor
		Read(V, I, P)=1	The current of the PL/PS/Mem is selected to monitor
		Read(V, I, P)=2	The power of the PL/PS/Mem is selected to monitor
Voltage value	Write	650 mV-1V	The target voltage value of the scaling

Table 2 - Complexity of the DVS unit components

Resource	FF	Utilization	LUT	Utilization
Microblaze processor	972	0.16%	631	0.21%
I2C Controller	343	0.06%	468	0.15%

Table 3 - Error codes

Error name	Error code
User command error	0x0002
PMBus initialisation error	0x0003
PMBus page writing error	0x0004
Writing to PMBus error	0x0005
Reading from PMBus error	0x0006

### ***B. Dynamic Frequency Scaling unit and testing platform***

The Dynamic Frequency Scaling (DFS) unit is based on a PicoBlaze™ [23] 8-bit microcontroller. This microcontroller is area-efficient and occupies only 26 Slices and 2 BRAMs. We have employed a reference design [14] built around the Picoblaze to scale frequencies and test the system. The reference design contains all the necessary routines to communicate with the off-chip Silicon Labs Si570 programmable oscillator to scale the frequency. The programmable oscillator available on the board operates with a frequency range of 10MHz to 945MHz. The Picoblaze receives the commands from the system processor in this scenario to scale the frequency and to inform the system processor when there is a timing failure.

The DFS unit and the DVS IP core occupy a small portion of the device. The same as in the Xilinx reference design [14]; we have employed a chain of Power Consuming and Speed Testing Modules (PCASTMs) with a variable number of modules to occupy different percentages of the device. Figure 3 displays the overview of the design. Each PCASTM module contains an additional KCPSM6 processor (i.e. Picoblaze) with three additional power consuming peripherals and a UART forming a communication pathway through the chain. The peripherals used in the PCASTM are as follows:

- 16 Toggle Flip-Flops: 16 flip-flops that toggle between "0101 0101 0101 0101" and "1010 1010 1010 1010".
- 16-Bit LFSR Counter: a maximal length Linear Feedback Shift Register (LFSR).
- 16-Bit Accumulator: connected to the 16-bit LFSR counter.

In addition, each PCASTM includes a simple 'speed test' (ST) circuit to evaluate the performance of the chain. The ST circuit of each module has an 8-bit LFSR counter and an 8-bit comparator. Each module is connected to its neighbours in the chain of PCASTM and compares the value of its own counter with the value of the counter in the previous module. Failure will be detected and reported as soon as any pair of counter values do not match.

We have implemented the test systems with an initial 100MHz clock frequency and the Picoblaze increases the frequency to detect the maximum operational frequency and performance. We have measured the latencies between the issuing of a monitoring command and when its execution completes at 1.63 ms. Also, commands that request a voltage scaling operation need approximately 8.64 ms to complete. These read and write latencies should be taken into account when developing energy proportional systems based on these devices and boards. We have also measured that the minimum safe voltage is 700 mV.

#### 4. Power and performance analysis

In this section we have implemented different test systems with a varying number of test modules to evaluate the run-time power and performance scaling of the systems.

##### 4-1-Area

Table 4 shows the number of LUTs and BRAMs which are occupied by different number of PCASTMs .As it can be seen in this table; we have used different portions of the device up to 66.42% and 97.28% of the LUTs and BRAMs, respectively.

Table 4 - Occupied area of the test system with different number of PCASTMs

Number of PCASTMs	Slice LUTs	Utilization (Slice LUTs)	BRAM count	Utilization (BRAMs)
50	12213	4.02%	73	5.049%
100	22151	7.30%	123	9.90%
150	32112	10.58%	173	14.76%
200	42233	13.91%	223	19.61%
250	52009	17.13%	273	24.47%
300	62062	20.44%	323	29.32%
350	72133	23.76%	373	34.17%
400	82036	27.02%	423	39.03%
450	92057	30.32%	473	43.88%
500	102019	33.60%	523	48.74%
550	111981	36.88%	573	53.59%
600	121943	40.17%	623	58.45%
650	131905	43.45%	673	63.30%
700	141868	46.73%	723	68.16%
750	151830	50.01%	773	73.01%
800	161792	53.29%	823	77.86%



850	171754	56.57%	873	82.72%
900	181716	59.85%	923	87.57%
950	191678	63.14%	973	92.43%
1000	201640	66.42%	1023	97.28%

#### 4-2- Analysis at a fixed frequency of 100 MHz

Figure 4 displays the monitored voltage for the test systems with different numbers of modules. The legend shows the requested voltages and VCCint shows the monitored voltage. This figure shows that the offset between requested and monitored voltage is maximum 1%.

Figure 5 displays the monitored power consumption for different test modules with different power supply voltages . This figure reveals that there is a linear relationship between occupied area and consumed power which is reasonable to expect. In addition, scaling voltage reduces the consumed power from 45.14% for the smallest configuration with 50 PCASTM modules up to 64.98% for the most complex configuration with 1000 PCASTM modules.

Figure 6 shows the monitored power consumption at the nominal Voltage (i.e. 1V) compared to the estimated power from the Xilinx power tool (Xpower Analyzer) for different test modules. Figure 6 shows that the measured power is approximately 30% higher than the values estimated by the Xpower Analyzer.

Figure 7 displays the temperatures reached by each of the configurations. As expected, more complex configuration increase the temperatures measured in the device but in all the cases, the temperatures remains below dangerous levels.

#### 4-3-Analysis at the maximum frequency

We have increased the clock frequency with the DFS IP core to investigate the maximum clock frequency for each configuration as well as measuring the power consumption and temperature at the maximum frequency.

Figure 8 presents the voltage, frequency and complexity analysis. This figure shows that the modules can clock from 800 MHz for the simplest configuration with 50 modules down to 650 MHz for the most complex with 1000 modules. Frequency reduces to a range of 350 MHz to 200 MHz for the 0.7 Voltage. Figure 8 also shows a drop for the configuration with 100 modules which can be considered out an outlier probably due to some place&route effect.

Figure 9 shows the total power for each of these configurations for the maximum frequency supported by each voltage. A large dynamic range of power values is possible ranging from less than 1 W to up to 9 W. This shows that energy proportional computing is possible and that different levels of performance and power can be achieved varying the complexity, voltage and frequency of the user design. For example, the lowest power corresponds to 50 modules, 339 MHz and 0.7 V at 0.29 W while the highest power corresponds to 1000 modules, 639 MHz and 1 V at 8.51 W.

The maximum allowed operating temperature for the device is 85°C according to Virtex-7 T and XT FPGAs data sheet [24]. In all these experiments an FPGA cooling fan is active at a constant rate. Figure 10 displays the

temperature of the device when it operates at the maximum frequency. Although a higher temperature at the maximum frequency is expected, the FPGA cooling fan keeps the temperature close to that of the 100 MHz case shown in Figure 7 and it stays well below the recommended 85°C value.

#### **4-4-Static Power**

We have implemented the systems with different complexities to measure the static power. The clock generator is stopped so that only static power remains using a user switch available on the board. We changed the monitoring method to the TI monitoring tool to measure the static power since the DVS core does not operate without clocks. Figure 11 shows the static power and voltage analysis. As it can be seen in this figure, the static power reduces up to 76.9% by scaling voltage from the nominal voltage to 0.7 V.

Figure 12 - Figure 15 compare the percentage of the static and dynamic power for the different numbers of test modules at the nominal, 0.9, 0.8, and 0.7 voltages. These figures show that percentage of static power decreases with more complex configurations. This is reasonable in FPGA devices since unused logic cells will still have leakage although they do not participate in the active computation. As it can be seen in these figures, down scaling the voltage reduce the percentage of the static power compared with dynamic power because the reduction in static power is determined by a higher order polynomial than in the case of dynamic power as seen in [13]. The percentage of the static power reduces by 19.25% and 9.43% for most and lowest complex configurations respectively when the voltage scales from nominal to 0.7 V.

#### **4-5-Margins analysis**

We have created timing constraints to analysis the maximum frequency of a single PCASTM module for each configuration with varying numbers of modules using the Xilinx timing analyzer software, which is available in the ISE package, and compare these frequencies with the maximum achieved frequencies in the physical prototype at nominal voltage to investigate the existing margins.

Figure 16 displays the software reports and achieved maximum frequencies. This figure shows that the static timing analysis reports a maximum frequency of around 200 MHz which is consistent with the value reported by the manufacturer in [23]. The figure also shows that there is a large margin compared with the measured performance. We have verified that the test circuits exercise the critical paths in the design validating this result.

### **5. Conclusion and future work**

Our previous work in [3] investigated the capability of standard FPGA devices to operate out of their nominal ranges with over and under scaling of frequency and voltage. The work presented in [3] was based on older Virtex-5 devices fabricated using a 65 nm process. In this paper we investigate if these margins are still present in modern 28 nm FPGAs that have the same nominal voltage of 1 V. The device considered belongs to the series 7 family. We propose a DVS unit that exploits the presence of controllable voltage regulators via the PMBus protocol to change voltages at run time while the DFS is based on a low overhead Picoblaze controller that communicates and programs the external oscillator available in the boards. The results reveal that although these FPGAs have not been validated by the manufacturer at below nominal voltage operational points, the margins

available make these chips a good platform for energy proportional computing. Future work involves further validation of the power adaptive architecture in a commercial application involving software and hardware components to accurately measure adaptability speed and energy savings in addition to the power.

## References

- [1] Kuon, I. and Rose, J. 2007. Measuring the gap between fpgas and asics. *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on 26, 2, 203 –215
- [2] [http://www.xilinx.com/support/documentation/data\\_sheets/ds180\\_7Series\\_Overview.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf)
- [3] Nunez-Yanez, J., "Adaptive Voltage Scaling with in-situ Detectors in Commercial FPGAs," *Computers, IEEE Transactions on*, vol.PP, no.99, pp.1,1, 0
- [4] Rahman, A., Das., Tuan T., and Rahut, A. 2005. Heterogeneous routing architecture for low-power FPGA fabric. In *Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005*. pp. 183 – 186.
- [5] Ryan, J. and Calhoun, B. 2010. A sub-threshold fpga with low-swing dual-vdd interconnect in 90nm cmos. In *Custom Integrated Circuits Conference (CICC), 2010 IEEE*. pp. 1 –4.
- [6] Li, F., Lin, Y., and He, L. 2004. Vdd programmability to reduce fpga interconnect power. In *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*. pp. 760 – 765.
- [7] Li, F., Lin, Y., He, L., and Cong, J. 2004. Low-power fpga using pre-defined dual-vdd/dual-vt fabrics. In *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. FPGA '04*. ACM, New York, NY, USA, 42–50.
- [8] Raham A. and Polavarapuv, V. 2004. Evaluation of lowleakage design techniques for field programmable gate arrays. In *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. FPGA '04*. ACM, New York, NY, USA, 23–30.
- [9] Lamoureux, J. and Wilton, S. . On the interaction between power-aware fpga cad algorithms. In *Computer Aided Design, 2003. ICCAD-2003. International Conference on*. 701 – 708.
- [10] Lamoureux, J. and Wilton, S. 2007. Clock-aware placement for FPGAs. In *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*. 124 –131.
- [11] Gayasen, A., Tsai, Y., Vijaykrishnan, N., Kandemir, M., Irwin, M. J., and Tuan, T. 2004. Reducing leakage energy in fpgas using region constrained placement. In *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. FPGA '04*. ACM, New York, NY, USA, 51–58.
- [12] Chow, C., Tsui, L., Leong, P., Luk, W., and Wilton, S. Dynamic voltage scaling for commercial FPGAs. In *Field-Programmable Technology, 2005. Proceedings. 2005 IEEE International Conference on*. 173 –180.
- [13] Atukem, N. Nunez-Yanez, J... Adaptive Voltage Scaling in a Dynamically Reconfigurable FPGA-Based Platform. *ACM Trans. Reconfigurable Technol. Syst.* 5, 4, Article 20 (December 2012)
- [14] Information available at [http://www.xilinx.com/support/documentation/application\\_notes/xapp555-Lowering-Power-Using-VID-Bit.pdf](http://www.xilinx.com/support/documentation/application_notes/xapp555-Lowering-Power-Using-VID-Bit.pdf)
- [15] S. Das, et al., *Razor II*, *IEEE J. Solid-State Circuits*, pp.32--48, Jan. 2009.
- [16] Information available at <http://pmbus.org/index.php>
- [17] Information available at <http://www.ti.com/lit/ug/sl00490/sl00490.pdf>

[18] Information available at <http://www.xilinx.com/support/answers/37561.html>

[19] Information available at [http://focus.ti.com/docs/toolsw/folders/print/fusion\\_digital\\_power\\_designer.html](http://focus.ti.com/docs/toolsw/folders/print/fusion_digital_power_designer.html)

[20] Information available at <http://focus.ti.com/docs/toolsw/folders/print/usb-to-gpio.html>

[21] Information available at <http://focus.ti.com/lit/ug/slou337/slou337.pdf>

[22] Information available at <http://pmbus.org/specs.html>

[23] Information available at <http://www.xilinx.com/products/intellectual-property/picoblaze.htm>

[24] Information available at

[http://www.xilinx.com/support/documentation/data\\_sheets/ds183\\_Virtex\\_7\\_Data\\_Sheet.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds183_Virtex_7_Data_Sheet.pdf)

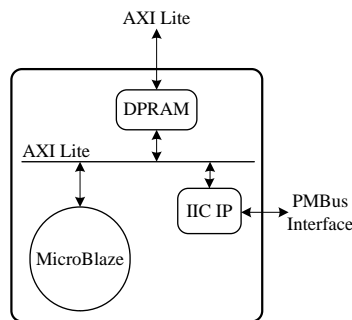


Figure 1 - DVS unit architecture

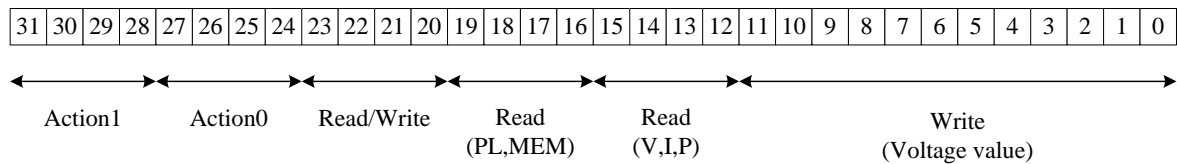


Figure 2 - Command parameters.

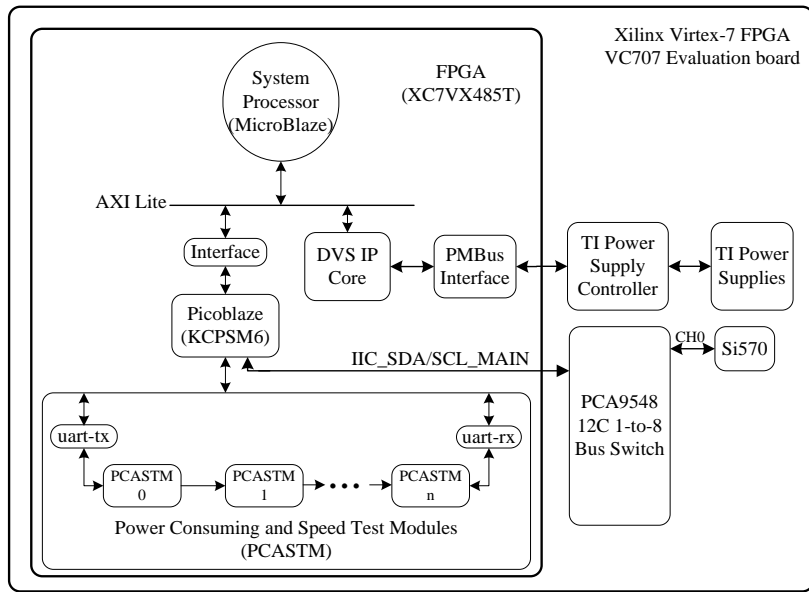


Figure 3 - Overview of the design

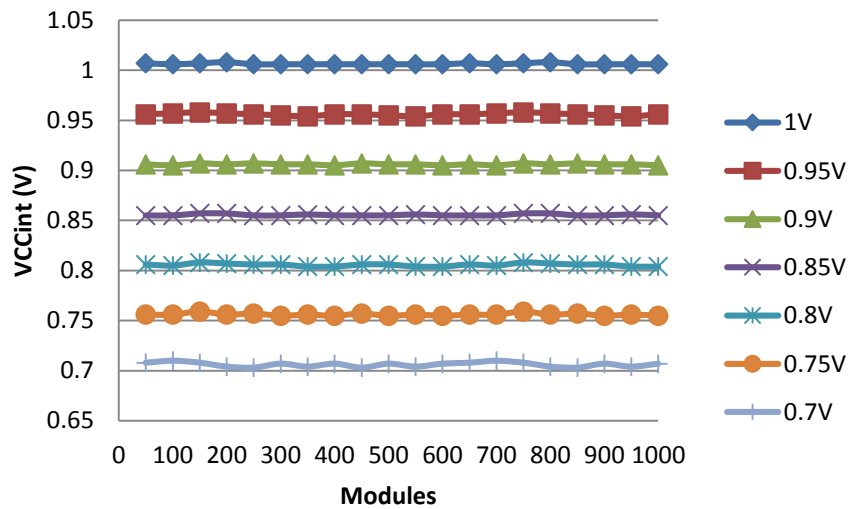


Figure 4 - Voltage scaling accuracy analysis

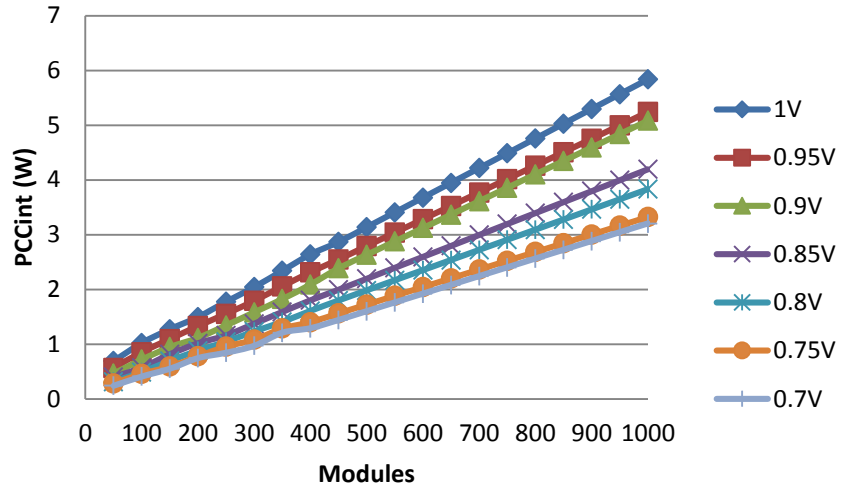


Figure 5 - Power and Voltage analysis

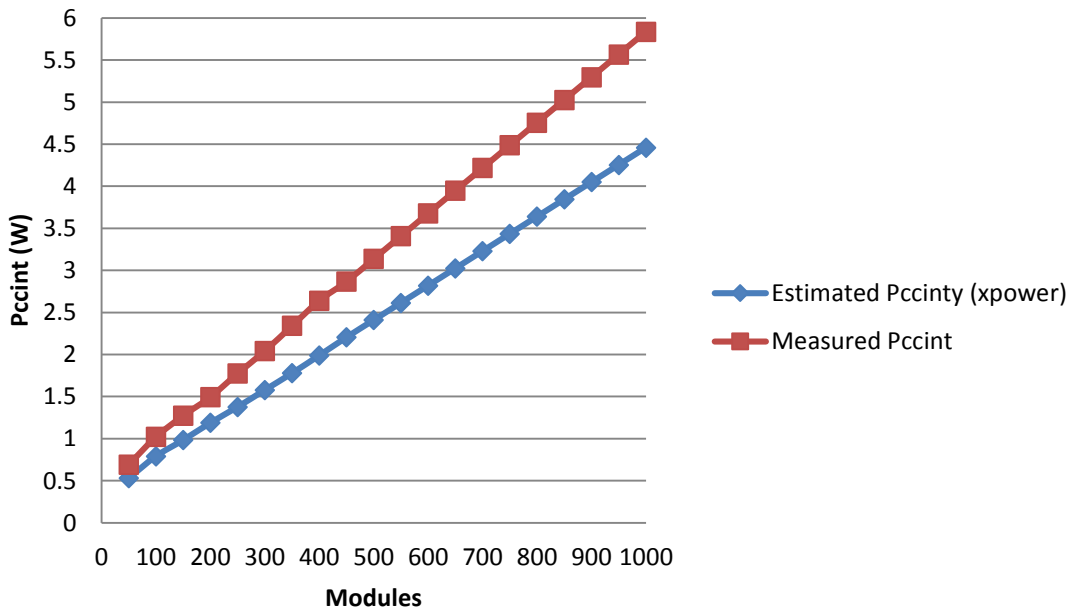


Figure 6 - Monitored power consumption compared to the Xilinx tool estimated power

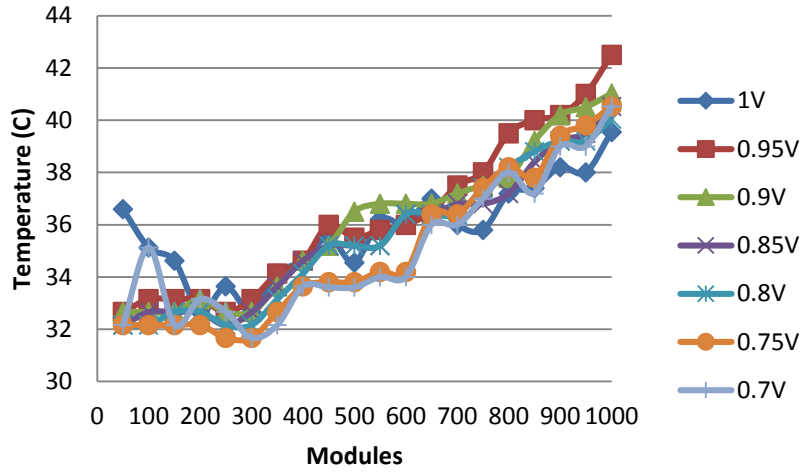


Figure 7 - Temperature analysis

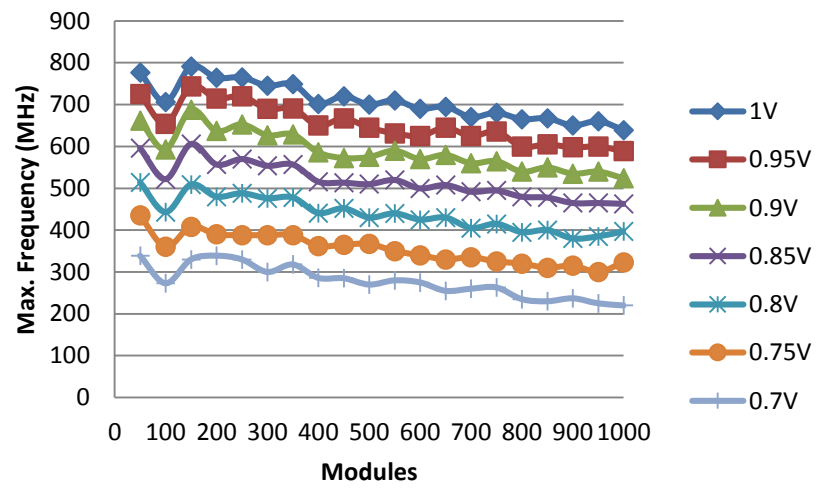


Figure 8 - Voltage, frequency and complexity analysis

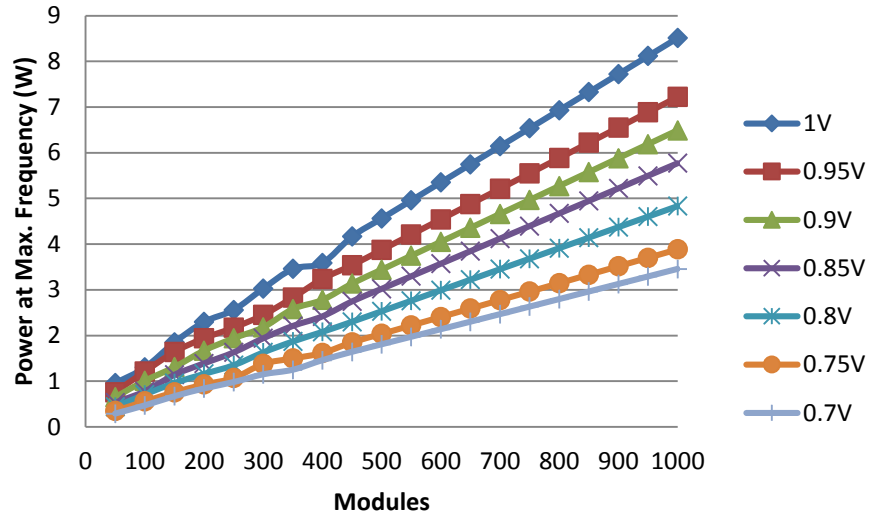


Figure 9 - Power and Voltage analysis at maximum frequency

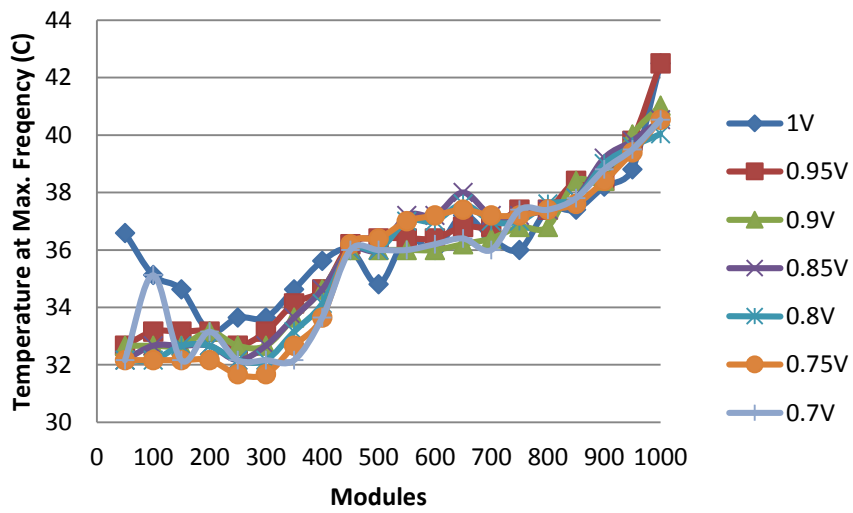


Figure 10 - Temperature analysis of the device when it operates in the maximum frequency



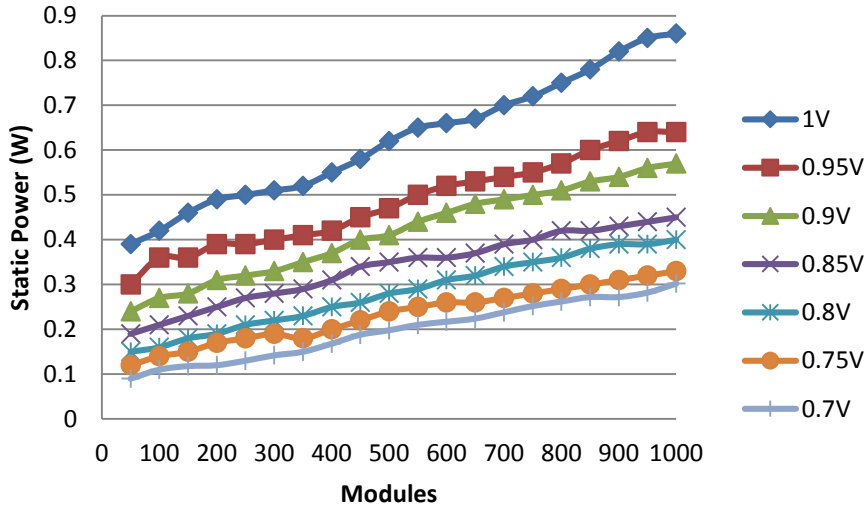


Figure 11 - Static Power and Voltage analysis

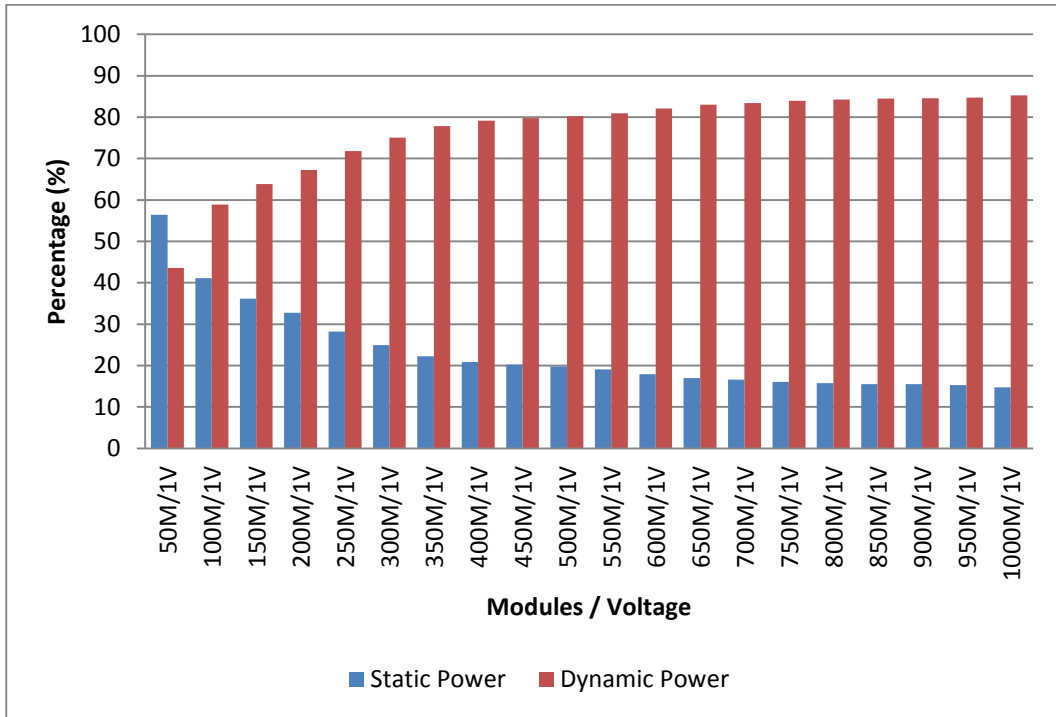


Figure 12 - Static Power and Dynamic Power at nominal voltage

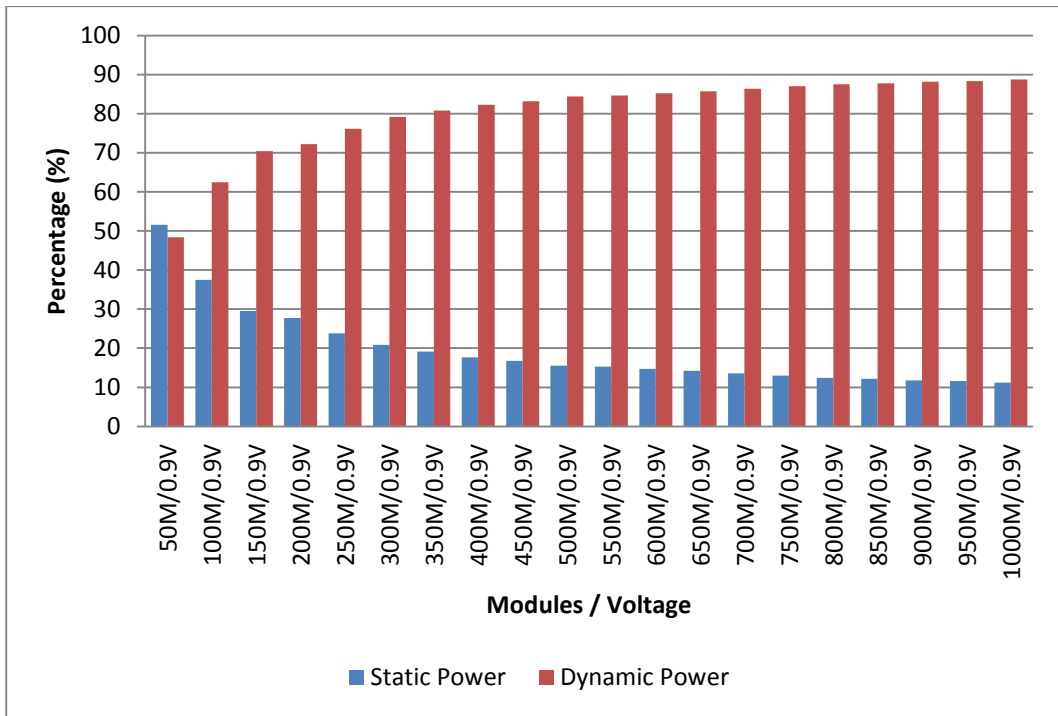


Figure 13 - Static Power and Dynamic Power at 0.9 V

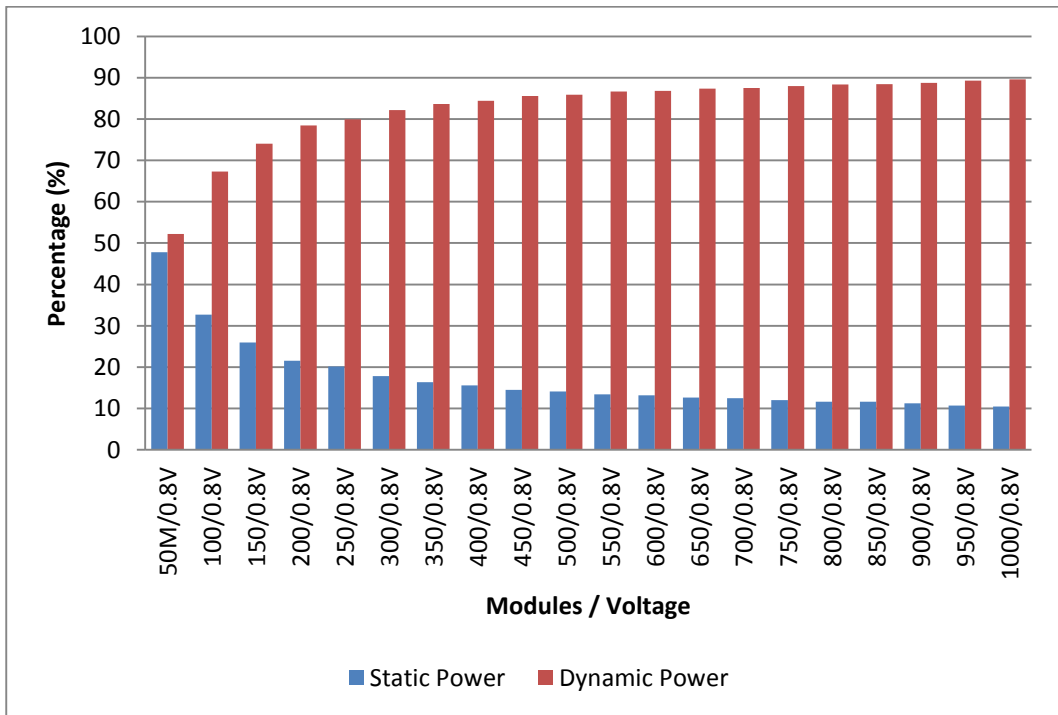


Figure 14 - Static Power and Dynamic Power at 0.8 V

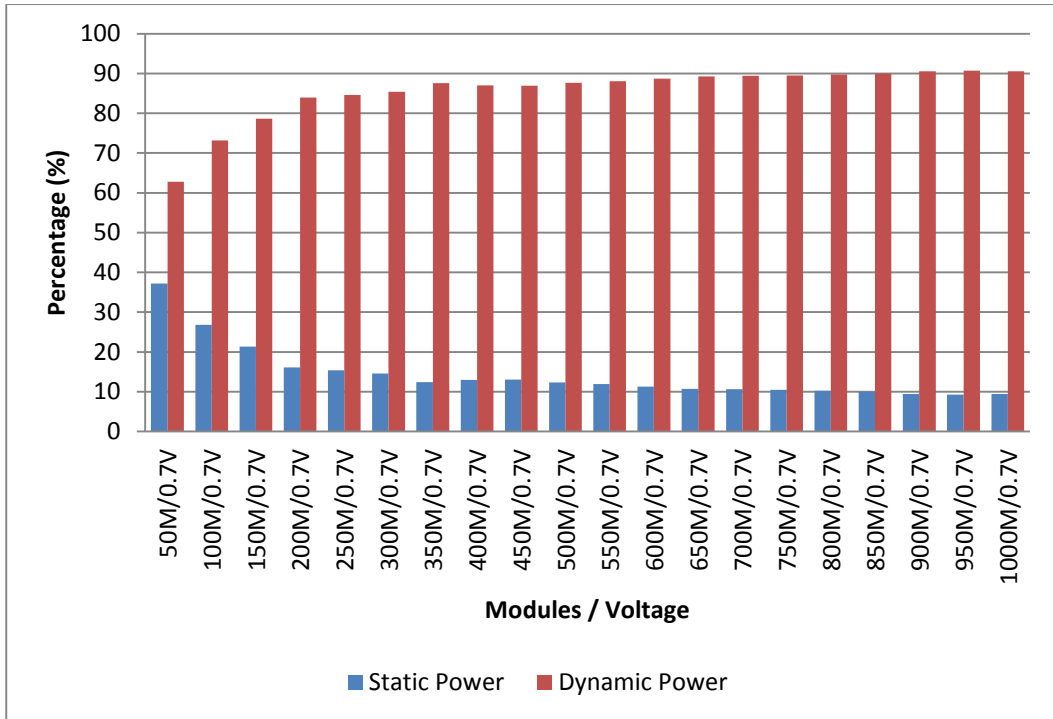


Figure 15 - Static Power and Dynamic Power at 0.7 V

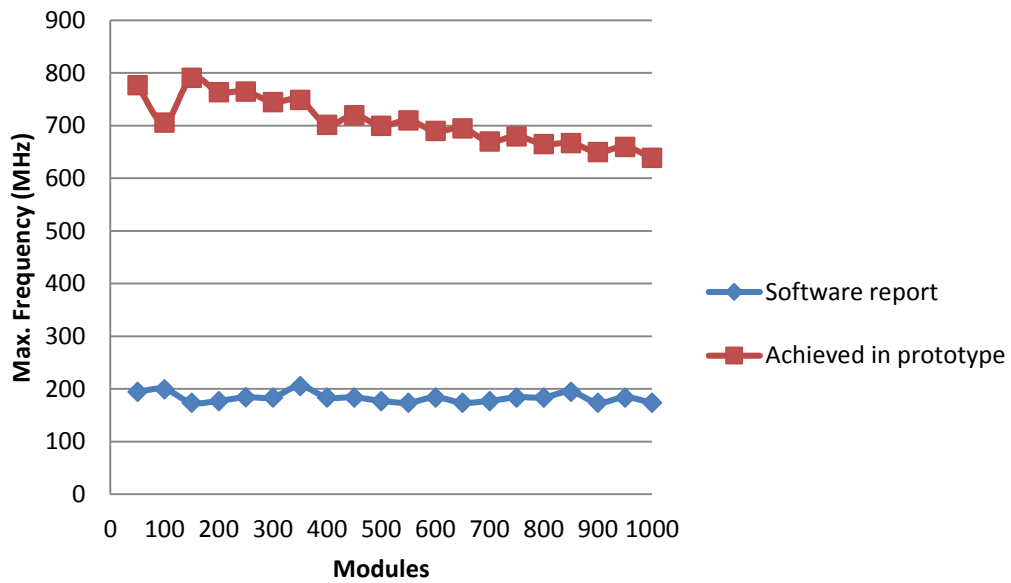


Figure 16 - Frequency analysis