Fault Characterization Through FPGA Undervolting

Behzad Salami*†, Osman Unsal*, and Adrian Cristal*†

*Barcelona Supercomputing Center (BSC), Barcelona, Spain. Emails: {behzad.salami, osman.unsal, and adrian.cristal}@bsc.es

†Universitat Politcnica de Catalunya (UPC), Barcelona, Spain.

***Abstract*—The power and energy efficiency of Field Programmable Gate Arrays (FPGAs) are estimated to be up to 20X less than Application Specific Integrated Circuits (ASICs). What is needed to close this gap is aggressive power/energy savings techniques. A such potentially effective approach is undervolting, which can directly deliver an order of magnitude static and dynamic power savings. However, aggressive undervolting, without accompanying frequency scaling leads to timing related faults, potentially undermining the power savings. Understanding the behavior of these faults and efficiently mitigate them can deliver further power and energy savings in low-voltage designs without performance degradation. In this paper, we conduct a detailed analysis of undervolting FPGA BRAM structures. Through experimental analysis, we found that lowering the supply voltage until a certain conservative level, V_{min}, does not introduce any observable fault. For the studied platforms, we measured this voltage guardband gap 39% of the nominal level ($V_{nom} = 1V$, $V_{min} = 0.61V$). Further undervolting corrupts some of the data bits stored in BRAMs; however, it also reduces the BRAMs power consumption a further 40%. When the voltage is lowered below V_{min}, the rate of these faults exponentially increases to 0.06%, by a fully non-uniform distribution over various BRAMs. This paper comprehensively analyzes the behavior of these faults.**

I. Introduction

Undervolting is a technique to decrease the supply voltage below the nominal level in order to save power and energy. Unlike Dynamic Voltage and Frequency Scaling (DVFS) [1] [2], the frequency is not scaled down in undervolting, so energy savings can be potentially significant. However, decreasing the voltage while keeping the frequency constant leads to timing related faults, which can cause applications to crash or terminate with wrong results. The severity of these faults depends on the fault rate, the fault location as well as on application characteristics. Therefore, characterization of these undervolting faults and understanding their behavior is critical to mitigate their impact. Although, there have been some previous undervolting works on CPUs [3], Graphic Processor Units (GPUs) [4], and Dynamic RAM (DRAM) memories [5], there are no ”deep-dive” undervolting fault characterization studies due to the relatively closed nature of these hardware substrates where the vendors expose few details. In comparison, the relatively open Field Programmable Gate Array (FPGA) architectures make it possible to conduct and report such detailed studies. However, to the best of our knowledge, such studies have not been thoroughly undertaken for FPGAs. Hence, the main contributions of this paper is extensively characterizing the behavior of faults, when commercial-FPGA BRAMs are aggressively undervolted. In this contribution, we target the FPGA on-chip Block RAM (BRAM) structures and report the most comprehensive experimental findings on undervolting using real hardware to date. We highlight three significant findings: *First*, we find that a significant voltage guardbanding of 39% exists below the nominal voltage level ($V_{nom} = 1V$, $V_{min} = 0.61V$) before faults start to occur for BRAM structures, which in turn leads to an order of magnitude power savings. Furthermore, we observe that the fault rate exponentially increases by further undervolting to a somewhat moderate 0.06% before the FPGA fails. *Second*, we find that the BRAM undervolting fault rate decreases at higher environmental temperatures; thus experimentally verifying the Inverse Temperature Dependence (ITD) [6]. ITD states that in undervolted nanometer technology nodes, the propagation delay is reduced in higher temperature environments that in turn, leads to a lower fault rate. This is significant since applying thermal stress would reduce the undervolting fault rate and thus also lowering the energy cost of fault mitigation.

This paper is organized as follows. In Section II we introduce the experimental setup. The overall behavior of BRAM undervolting is described in Section III. Fault characterization is discussed in Section IV. Finally, in Section V, we review the previous work.

II. Experimental Methodology

We perform our experiments on two FPGAs, i.e., XC7VX485T representing the Virtex7 family on a VC707 board and XC7K325T representing the Kintex7 series on a KC705 board. These FPGAs are respectively equipped with 2060 and 890 BRAMs, distributed over the chip with the size of 16 Kbits each. Each BRAM is a matrix of bitcells with 1024 rows and 16 columns. BRAMs can be either individually accessed or cascaded to build larger memories (with some overheads). This methodology provides flexibility for the FPGA designers to have single-cycle access to on-chip memories as per bandwidth or size needs. More details of our tested platforms are shown in Table I. Both platforms are fabricated with 28nm technology, and the standard nominal voltage of BRAMs is the same, $V_{nom} = 1V$. However, their difference is that XC7VX485T (Virtex7) is designed for performance while XC7K325T (Kintex7) is optimized for the power consumption. Hence, for a thorough evaluation, we selected these representative FPGAs.

Through the Power Management Bus (PMBUS) standard [7], it is possible to independently and dynamically regulate and monitor the supply voltage of such FPGA components as BRAMs (V_{CCBRAM})To modify V_{CCBRAM}, we use Texas

TABLE I: Specifications of Tested Platforms.

Hardware Platform (Board)	VC707	KC705
Device Family	Virtex 7	Kintex 7
Chip Model	XC7VX485T	XC7K325T
Number of BRAMs	2060	890
Basic Size of Each BRAM	1024*16-bits	1024*16-bits
Manufacturing Process Technology	28nm	28nm
Nominal V_{CCBRAM} (V_{nom})	1V	1V
Speed Grade	-2	-2

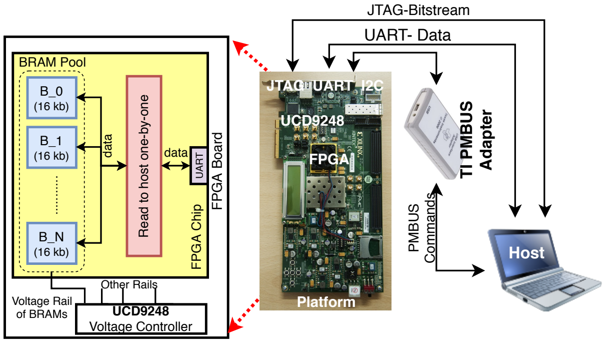


Fig. 1: Undervolting Experimental Setup in FPGA BRAMs.

List 1: Pseudo-code to Study Voltage Scaling on BRAMs, on the Experimental Setup of Fig. 1.

```
$V_{CCBRAM} = V_{min} = 0.61V$;
while($V_{CCBRAM} >= V_{crash}$) begin
    while(numRun <= 100) begin
        delay(1sec);
        Transfer content of BRAMs to the host;
        Analyse faulty data (rate and location);
        numRun++;
    end
    $V_{CCBRAM} -= 10(mV)$;
end
```

Instrument (TI) PMBUS USB Adapter, and the provided C-based Application Programming Interface (API), which facilitates accessing the on-board voltage controller through the host [8]. The experimental setup is shown in Fig. 1. It is composed of two distinct hardware and software components. The task of the hardware FPGA platform is to access BRAMs and transmit their content to the host, using a serial interface. On the other side, the host issues the required PMBUS commands to set a certain voltage to V_{CCBRAM}. Also, it analyzes potentially faulty data retrieved from BRAMs. Note that we verify and validate that the implemented serial interface is entirely reliable in any V_{CCBRAM} level.

On our setup, shown in List 1, first, we initialize V_{CCBRAM} with $V_{min} = 0.61V$. Then, we retrieve the content of BRAMs one-by-one and within each BRAM row-by-row, and transfer them to the host. In the host, we analyze the rate and location of faults. This process is repeated 100 times for each voltage level to obtain statistically significant results. The reported results in this paper are the median of these 100 tests. After a soft reset, we decrease V_{CCBRAM} by 10mV and repeat the process until the lowest voltage that our design operate, $V_{crash} = 0.54V$. Finally, to measure the power consumption with acceptable accuracy, we use a power meter, while to extract the power contribution of BRAMs in the nominal voltage level, we use Xilinx Power Estimation (XPE) tool. Note that experiments are performed on the default and fixed internal frequency of BRAMs, i.e., 555Mhz (1.8ns internal logic delay). On this setup, V_{CCBRAM} is gradually lowered, while reading contents of BRAMs until system crashes. For each voltage level, the fault rate and power consumption of BRAMs are recorded.

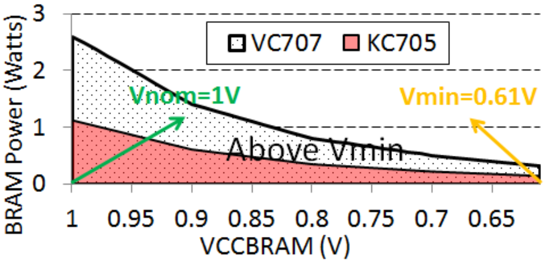
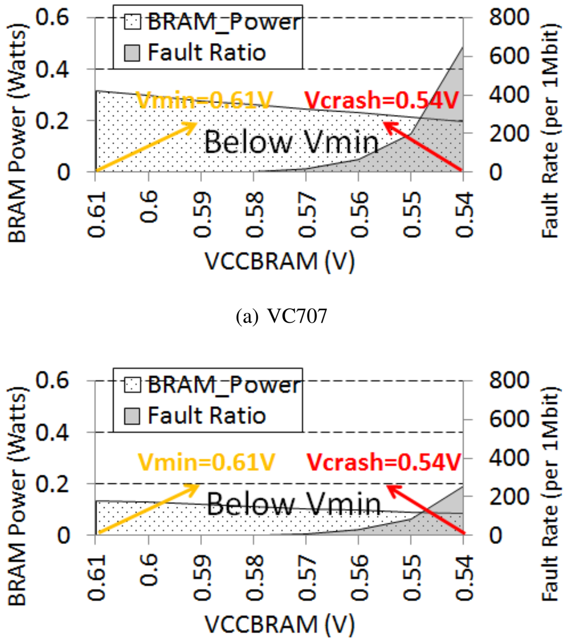


Fig. 2: BRAM Power above $V_{min} = 0.61V$ in the voltage guardband region (no observable fault).

III. Overall Behavior on BRAM Undervolting

Our experiments on undervolting BRAMs below nominal level, $V_{nom} = 1V$, demonstrate two thresholds. *First*, a voltage guardband, V_{min}, that separates the fault-free and faulty regions. *Second*, V_{crash} that is the lowest level of the voltage that our design practically operates, below that FPGA fails. In our test environment and for both tested platforms, $V_{nom} = 1V$ due to the factory settings, whereas $V_{min} = 0.61V$ and $V_{crash} = 0.54V$, obtained through our experiments. Note that repeating these tests in more noisy and harsh environments, worst case scenarios, can cause observable faults above 0.61V, as well; however, our tests are performed under normal environmental conditions. Also, due to our experiments with various Xilinx FPGAs at 28nm technology, we posit that $V_{crash} = 0.54V$ is strictly set by the factory to prevent device damage in extremely low voltages.

When $V_{CCBRAM} >= V_{min}$, no observable faults occur. When $V_{CCBRAM} = V_{min} = 0.61V$ we observed significant BRAM power savings over $V_{nom} = 1V$, more than an order of magnitude for both platforms including the sum of static and dynamic power, without incurring any reliability degradation, as shown in Fig. 2. Further lowering V_{CCBRAM} below V_{min} the fault rate exponentially increases, while the power consumption is reduced, as summarized in Fig. 3. As can be seen, both power consumption and reduction are less in KC705 than VC707, which is the consequence of having relatively less BRAMs and also the inherent power optimizations adopted for KC705 by the vendor.



(a) VC707

(b) KC705

Fig. 3: BRAM Power and Fault Rate Behavior, undervolting below $V_{min} = 0.61V$ until $V_{crash} = 0.54V$.

TABLE II: Fault Rate Stability Over Time. Fault Rate Analysis of 100 runs at $V_{crash} = 0.54V$ with pattern=16'hFFFF.

Parameter	VC707	KC705
AVERAGE fault rate (per 1 Mbit)	652	254
MINIMUM fault rate (per 1 Mbit)	630	240
MAXIMUM fault rate (per 1 Mbit)	669	264
STANDARD DEVIATION of fault rates	7.3	4.8

IV. Fault Characterization

In this section, we comprehensively characterize the behavior of faults, where V_{CCBRAM} is underscaled from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$. As can be seen in Fig. 3a and 3b, the fault rate in VC707 is relatively more than KC705, by 47.4% on average. This difference can be the consequence of the architectural and technological differences adopted to optimize performance and power in VC707 and KC705, respectively.

A. Fault Stability Over Time

As earlier mentioned, we repeat each test 100 times to get statistically significant results. We did not observe a significant difference among runs, since the standard deviation of the fault rate and locations among these runs is negligible. More details about these 100 runs are summarized in Table. II.

B. Fault Variability Among BRAMs

By statistically analyzing experimental results, we observed that the fault rates in aggressively low-voltage regions below $V_{min} = 0.61V$ considerably varies among BRAMs. For instance, experimenting on VC707 at $V_{crash} = 0.54V$, the maximum, minimum, and average fault rate within BRAMs are 2.84%, 0%, and 0.04%, respectively. Also, 38.9% of BRAMs in VC707 and 45.2% in KC705 has no observable faults. As a further analysis, we clustered this statistical information in low-, mid-, and high-vulnerable classes of BRAMs, using the k-means clustering algorithm. As can be seen in Table. III, for instance, the vast majority of BRAMs in VC707, 88.6%, are recognized as low-vulnerable with an average fault rate of 0.02%, ~ 3.4 faults within an individual BRAM with the size of 1024*16-bits.

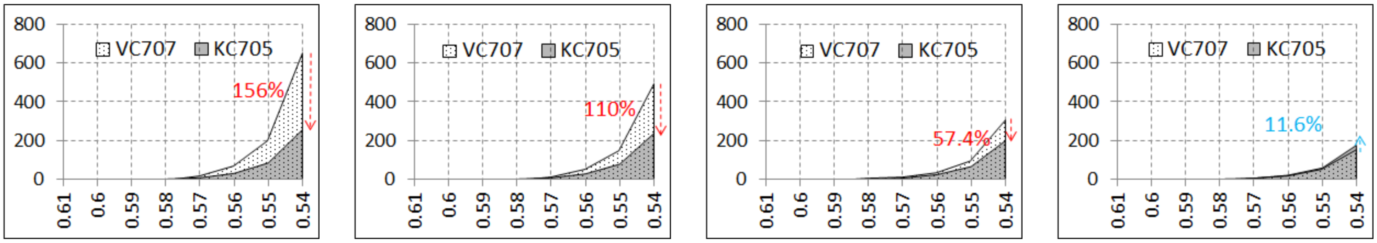
TABLE III: Clustering BRAMs due to Vulnerability Features (Reported numbers are at $V_{crash} = 0.54V$).

		VC707	**KC705**
low-vulnerable	%BRAMs	88.6%	93.4%
	Average Fault Rate (%)	0.02%	0.01%
mid-vulnerable	%BRAMs	9.4%	5.7%
	Average Fault Rate (%)	0.24%	0.17%
high-vulnerable	%BRAMs	1.8%	0.9%
	Average Fault Rate (%)	0.86%	0.74%

This significant fault variability among BRAMs can be due to either the chip-dependent process variation or design tools for place and route. We verify this argument by performing the following test; for our test design, we extracted the fault rate of BRAMs with several place-and-route compiles. Repeating the voltage lowering operation on these various bitstreams, we observed almost an identical fault rate in the corresponding physical locations of BRAMs. Hence, we conclude that this fault rate variability among BRAMs is the result of the process variation.

C. Impact of the Environmental Temperature

We perform an experiment to study the effect of the environmental temperature on the behavior of faults when V_{CCBRAM} is dropped below V_{min}. Toward this goal, we repeat the original design described in Section II, while the hardware platform is placed inside a chamber that its temperature can be regulated by a heater. In this setup, we underscale the V_{CCBRAM} and retrieve the content of BRAMs to extract the fault rate, while controlling and monitoring the on-board temperature in the host, through an on-chip sensor. The experimental results are shown in Fig. 4 under various on-board temperatures, 50°C (default temperature), 60°C, 70°C, and 80°C. As can be seen, with heating up, the fault rate is constantly reduced. For instance, the fault rate reduces by more than 3X for the VC707, by going from the default on-board temperature, 50°C to 80°C. This observation is the consequence of the Inverse Thermal Independence (ITD) property [6]. ITD is a thermal property of devices with nanoscale technology nodes. It states that in contrast with the traditional CMOS technologies, in the nanoscale technology nodes at low-operating voltages the circuit delay reduces. The reason is that as the technology scales down, the supply approaches the threshold voltage. Hence, at low-voltage regimes, increasing the temperature reduces the threshold voltage and allows the device to switch faster at higher temperatures. With the circuit delay decrease,



(a) On-board Temperature= 50°C (b) On-board Temperature= 60°C (c) On-board Temperature= 70°C (d) On-board Temperature= 80°C

Fig. 4: The correlation among on-board temperature, supply voltage of BRAMs, technology (VC707 vs. KC705), and fault rate. (x-axis: V_{CCBRAM} from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$, y-axis: the fault rate (per 1Mbit))

subsequently, the number of critical paths, and in turn, the rate of timing faults in the critical paths are reduced, as we experimental verified in our case. Also, as can be seen, the fault rate in VC707 is reduced more aggressively than KC705. A relatively 156% more fault rate in 50°C is reduced to 11.6% less ratio in 80°C, for VC707 vs. KC705. The architectural and technological difference between these platforms can be the reason, since their design goal is different; performance (VC707) and power (KC705).

V. Related Work

Most commercial devices are designed with a voltage guardband below the standard minimum nominal supply voltage to ensure the correct functionality in the worst case environmental and process variations. This voltage guardband is fully vendor- and system-dependent; for instance, it was measured to be 20% in GPUs [9] and 16% in DRAMs [5]. We experimentally determined the voltage guardband for Xilinx FPGA BRAMs to be 39%, in which delivers more than an order of magnitude power savings.

Tackling with the increased delay in low-voltage regions below V_{min}, accompanying frequency underscaling is a promising solution [1]; however, with the cost of performance degradation. A more aggressive approach is to allow designs to experience timing faults and in turn, tolerating these faults. Characterizing these faults can allow better power and reliability trade-offs, without performance degradation, as is for DVFS approach. Among the real hardware devices, this approach is extensively studied for modern processors [10], [11], [12], [13]; however, there are several recent efforts on other hardware devices, as well, i.e., GPUs [4], ASICs [14], [15], and memory systems [5], [16]. In parallel, several simulation-based framework [17] or design optimization [18], [19] are also proposed to study undervolting through nano-meter technology parameters; however, it is evident that this approach lacks the exact information of the fault model under very low-voltage operations and their validation on the silicon remains a key question. Our paper studies aggressive undervolting for the first time in commercial FPGAs, emphasized on on-chip BRAMs.

Acknowledgment

We thanks Pradip Bose, Alper Buyuktosunoglu, and Augusto Vega from IBM Watson for their contribution to this work. The research leading to these results has received funding from the European Union's Horizon 2020 Programme under the LEGaTO Project (www.legato-project.eu), grant agreement n° 780681.

References

[1] Nunez-Yanez, et al. "Energy optimization in commercial FPGAs with voltage, frequency and logic scaling", in *IEEE Transactions on Computers*, 2016.
[2] G. Semeraro, et al. "Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling", in *HPCA*, 2002.
[3] R. Bertran, et al. "Very Low Voltage (VLV) Design", in *ICCD*, 2017.
[4] J. Tan, et al. "Combating the reliability challenge of GPU register file at low supply voltage", in *PACT*, 2016.
[5] K. K. Chang, et al. "Understanding reduced-voltage operation in modern DRAM devices: Experimental characterization, analysis, and mechanisms", in *Measurement and Analysis of Computing Systems*, 2017.
[6] Neshatpour, K., et al. "Enhancing Power, Performance, and Energy Efficiency in Chip Multiprocessors Exploiting Inverse Thermal Dependence", in *IEEE TVLSI*, 2018.
[7] "Power Management Bus (PMBUS)." http://pmbus.org
[8] Texas Instruments (TI), "Fusion Digital Power Designer". http://www.ti.com/tool/FUSION_DIGITAL_POWER_DESIGNER
[9] J. Leng, et al. "Safe limits on voltage reduction efficiency in GPUs: a direct measurement approach", in *MICRO*, 2015.
[10] A. B. Kahng, et al. "Designing a processor from the ground up to allow voltage/reliability tradeoffs", in *HPCA*, 2010.
[11] K. Swaminathan, et al. "Bravo: Balanced reliability-aware voltage optimization", in *HPCA*, 2017.
[12] G. Papadimitriou, et al. "Harnessing voltage margins for energy efficiency in multicore CPUs", in *MICRO*, 2017.
[13] A. Bacha, et al. "Dynamic reduction of voltage margins by leveraging on-chip ECC in Itanium II processors", in *ISCA*, 2013.
[14] G. Tziantzioulis, et al. "b-HiVE: A bit-level history-based error model with value correlation for voltage-scaled integer and floating point units", in *DAC*, 2015.
[15] P. N. Whatmough, et al. "14.3 a 28nm SOC with a 1.2 ghz 568nj/prediction sparse Deep-Neural-Network engine with¿ 0.1 timing error rate tolerance for IOT applications", in *ISSCC*, 2017.
[16] L. Yang, et al. "SRAM voltage scaling for energy-efficient convolutional neural networks", in *ISQED*, 2017.
[17] J. Zhang, et al. "ThUnderVolt: Enabling Aggressive Voltage Underscaling and Timing Error Resilience for Energy Efficient Deep Neural Network Accelerators", *arXiv:1802.03806*, 2018.
[18] B. Reagen, et al. "Minerva: Enabling low-power, highly-accurate deep neural network accelerators", in *ACM SIGARCH Computer Architecture News*, 2016.
[19] G. Yalcin, et al. "Exploring Energy Reduction in Future Technology Nodes via Voltage Scaling with Application to 10nm", in *PDP*, 2016.