TuneFPGA: Post-Silicon Tuning of Dual-Vdd FPGAs

Stephen Bijansky
The University of Texas at Austin

Adnan Aziz
The University of Texas at Austin

ABSTRACT

Modern CMOS manufacturing processes have significant variability, which necessitates guard banding to achieve reasonable yield. We study an FPGA architecture with a dual voltage supply wherein the supply voltage for individual CLBs can be assigned after fabrication; this yields a mechanism for fixing chips that fail because of manufactured transistors being slower than designed. The fundamental advance our work makes is that we assign voltages based on manufactured data rather than designed values. The key contributions of our work are a CAD methodology and a detailed quantitative study using realistic data on the latest process technologies of the impact of post-manufacturing tuning on yield and power for dual-Vdd FPGAs. We find that, for a representative modern process, post-manufacturing tuning can increase the yield by up to $10\times$ compared with a conventional dual-Vdd design that selects the voltage supply pre-manufacturing, even with guard banding. Overall, the geometric mean of yield/power ratio is 27% greater using post-manufacturing tuning.

Categories and Subject Descriptors: B.6.1 Logic Design
General Terms: Design, Performance, Reliability
Keywords: Process Variation, Tuning, Yield, Delay, FPGA

Dedication

To the memory of Margarida Jacome, who inspired many of the ideas explored in this paper. She is missed.

1. INTRODUCTION

Process variations originate from many sources, including lithography, substrate doping, and chemical-mechanical polishing. Variations can be characterized as either systematic or random. Systematic variations are then broken down into wafer-to-wafer variations, die-to-die variations, and within-die variations. There have been many studies on the effects of variations. One representative study is Borkar *et al.* [1], which showed that for a batch of microprocessors all on the same wafer, variations in transistor channel length and variations in threshold voltage contributed to a 30% difference in chip frequencies. Borkar also reports that the standby leakage current varied by as much as $20\times$. The variation in leakage current is particularly significant since leakage continues to increase in importance in recent designs—in a dual core 65 nm Xeon processor, leakage accounted for 30% of the total power [2].

An increase in the number of variation sources has led to even more corner cases that need to be simulated for each design. Nassif [3] and the ITRS have estimated that the $3\sigma/\mu$ variation for 65 nm process technology is as high as 30% for transistor channel length and transistor threshold voltage. Future processes could have even larger amounts of variation.

DAC 2008, June 8–13, 2008, Anaheim, California, USA
Copyright 2008 ACM 978-1-60558-115-6/08/0006…5.00

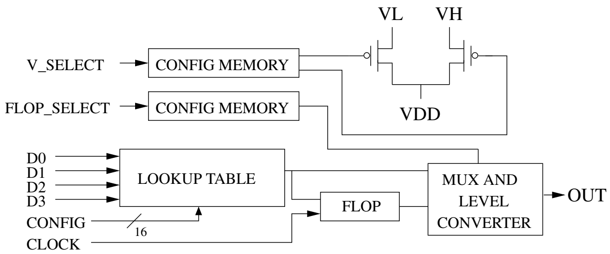

Figure 1: Tunable dual-Vdd CLB with configurable voltage supply PMOS transistors. The lookup table, flop, and multiplexer are connected to the local Vdd. VH is connected to the high supply voltage grid, and VL is connect to the low supply voltage grid. The multiplexer includes a level converter in the VL path only. Therefore, the CLB output is always high voltage.

Our work is embodied in the TuneFPGA system. TuneFPGA includes process models, transistor-level schematics for the custom dual-voltage CLB we designed, a voltage selection algorithm, and scripts using publicly available tools for performing yield-power tradeoffs. TuneFPGA is freely available at [4].

Broadly speaking, our approach differs from previous work in that we tune individual chips after fabrication. Therefore, this tunable CLB is able to respond to actual chip variations instead of estimated values. After the chip is fabricated, TuneFPGA tests each CLB's performance. Next, TuneFPGA computes CLB supply voltage assignments. Lastly, the chip is programmed with the CLB supply voltage assignments. Since TuneFPGA takes place after manufacturing, designers will still be able to use all of their current tools.

2. DUAL VOLTAGE CLB DESIGN

We focus on tuning using a dual voltage supply because changes in the supply voltage have such a large impact on both the speed and power consumption of the chip. Recall that by selecting a higher supply voltage, the chip will operate faster but will use more power; a lower supply voltage leads to a slower chip that uses less power. The designer can use the lower voltage supply for logic on paths that easily meet their delay constraints. These non-critical paths can therefore run slower and use less power.

In a dual voltage design style, there are two voltage grids across the entire chip. Each CLB is connected to the appropriate supply grid through a pmos pass transistor. By having only one pmos pass transistor turned on at a time, each CLB can have one of two supply voltage choices.

Since there will be multiple voltage supplies on the chip, level converters are required at all junctions between low voltage outputs and high voltage inputs. A level converter is a buffer with additional transistors to help prevent short circuit current. TuneFPGA incorporates the Kulkarni *et al.* [5] STR6 level converter into the output multiplexer. By placing a level converter in the output multiplexer, TuneFPGA ensures that all CLB outputs will be high voltage. In order to preserve speed, the high-Vdd path through the multiplexer bypasses this level converter.

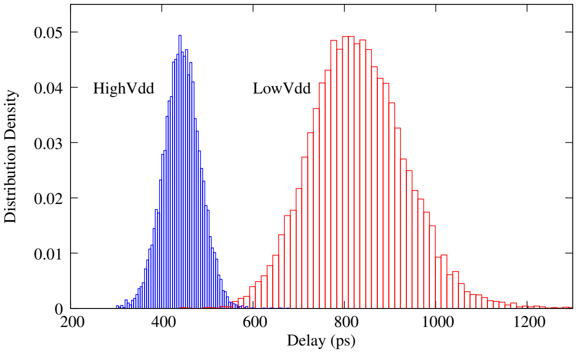

Figure 2: Dual-Vdd delay distribution for a single CLB

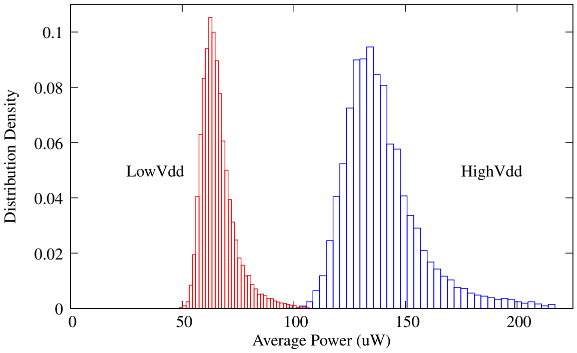

Figure 3: Dual-Vdd total power distribution for a single CLB

2.1 Tunable CLB

Figure 1 shows the CLB we designed. This CLB uses a LUT, a register, and a multiplexer to bypass the register. Compared with the Altera ALM and Xilinx CLB, the major addition to our CLB is that there are pass transistor connections to both a high-supply voltage grid and a low-supply voltage grid [6]. The LUT, flop, and multiplexer are all connected to a local Vdd for each CLB. The CLB can switch from one supply voltage grid to the other by turning on one of the large pmos pass transistors connected to the CLB's Vdd interconnect. Then, the local Vdd is used to connect to the CLB elements. The supply voltage configuration is stored in a register with complementary outputs so that only one pmos pass transistor can be turned on at a time. In this design, the supply voltage configuration is programmed before normal operation of the CLB begins.

2.2 Area Overhead

In order to size the pmos pass transistor, a SPICE simulation sweep was performed. The width was varied from 50× minimum size to 500× minimum size. The chosen pass transistor size of 133× of the minimum size resulted in less than a 1% increase in LUT delay compared to having no pass transistor. After layout, the area of one pass transistor was comparable to a standard cell gate. See [4] for details of this layout. In situations in which total area is more important than delay, a smaller pass transistor can be used.

The STR6 level converter [5] is composed of a buffer with 6 additional small transistors. These 6 additional transistor have about the same transistor width as a buffer. The overall area cost of the level converter is about the same as 2 buffers, which is small compared to the total area of a CLB cell.

3. CLB DELAY STATISTICS

We performed Monte Carlo simulations on the dual-Vdd CLB shown in Figure 1. The 65 nm Berkeley PTM model [7] was used for the SPICE transistor models. For each transistor in the CLB, both the transistor length and threshold voltage were modeled as uncorrelated Gaussian distributions with the $3\sigma/\mu$ variation chosen to be 20% [8]. The high voltage supply was set to 1.2 V and the low voltage supply was set to 0.8 V [9]. The simulation temperature was set to 85 °C.

We measured the performance of our CLB in 100,000 Monte Carlo simulations using HSPICE. The LUT was programmed for the worse-case delay configuration, which was fifteen logic-zeros and one logic-one. This leads to a slow rise time for LUT output, which in turns leads to a slow fall time for the CLB output because of the inverting multiplexer. Delay was measured from the 50% crossing of the input signal to the 50% crossing of the output signal.

In Figure 2, the worse-case falling output transition delay distribution is shown for all 100,000 trials. In this figure, the bars on the left represent the delay when all the CLB elements are connected to the high voltage supply through the pmos pass transistor. Then, the CLB is disconnected from the high voltage supply and connected to the low voltage supply through the other pmos pass transistor. The bars on the right represent the delay when all of the CLB elements are connected to the low voltage supply. In order to present an equal comparison, the number of bars on the left is equal to the number of bars on the right. This graph shows that there is a large spread of delay values for this CLB when intra-chip variation is taken into account. Moreover, the low-Vdd distribution has a larger amount of variation.

The average power was measured from when the input changes by 1% until the output reaches 99% of its final value. The power includes both switching and leakage. Figure 3 show the total power values; the power spreads are similar to the delay spreads. There are also similar graphs for only leakage power.

Taken together, these graphs show that there is potential to reduce both dynamic power and leakage power. The following summarizes the potential power savings for a single CLB when using a dual-Vdd approach:

1. The delay increases from a mean of 444 ps using high-Vdd to 831 ps using low-Vdd.
2. The power decreases from a mean of 140 μw using high-Vdd to 70 μw using low-Vdd.
3. The leakage power decreases from a mean of 25.7 μw using high-Vdd to 15.7 μw using low-Vdd.

If the designer is able to decrease the speed on non-critical path CLBs by switching those CLBs to a low voltage supply, power can be reduced. Even with CLBs on the critical path, if there is enough slack, some of those CLBs could use a low supply voltage while the other CLBs use a high supply voltage. Overall, there is potential to reduce total power by almost 50%.

4. POST-MANUFACTURING DELAY MEASUREMENT

TuneFPGA uses post-manufacturing CLB delay measurement, which is the subject of ongoing research. Nabaa *et al.* [10] present a tunable body bias FPGA that includes a CLB characterizer. Nabaa's characterizer uses a phase detector to measure delay. The characterizer sends a clock signal to the CLB under test. The output of the CLB under test is sent back to the characterizer. The phase detector compares the initial clock signal to the CLB output. Using this method, the characterizer measures the delay of both the CLB and any intermediate routing resources. The characterizer starts with CLBs that are adjacent to the characterizer. Then, the characterizer proceeds to sequentially test CLBs further away while compensating for the already measured delay through intervening CLBs and interconnects. Nabaa has estimated that the area of characterizer is about nine FPGA tiles.

Another promising approach would be to implement delay characterization based on Razor by Ernst *et al.* [11]. By using a shadow latch and comparator logic, Razor has mechanisms to monitor when a delay error has taken place. For TuneFPGA, a test input could run through the CLB in successively faster clock cycles until there is a delay error. Additionally, neighboring CLBs could perform the shadow latching and comparator logic needed for Razor testing using existing CLB resources.

Other delay characterization methods include the work by Dhar *et al.* [12], which introduces an adaptive voltage scaling controller that uses an inexpensive ring oscillator to measure speed. By placing multiple ring oscillators throughout the design, CLB delay can be approximated based on the delay of the nearest ring oscillator.

5. CAD FLOW

TuneFPGA integrates existing design tools with a custom C++ program we wrote that implements the voltage assignment algorithm described below. The inputs to TuneFPGA are a manufactured design with CLB delay measurements, a logic netlist to use for static timing analysis, and a target delay. The output is a voltage assignment for each CLB. Since TuneFPGA is based on static timing analysis, it is very fast.

First, the logic netlist, in blif format, is converted into structural Verilog using PERL. Synopsys PrimeTime then generates static timing paths that are using for timing analysis. Next, TuneFPGA assigns the supplied delay measurement for each CLB in the design.

TuneFPGA is now ready for the voltage assignment algorithm (VAA). TuneFPGA-VAA starts by assigning all CLBs to use a low-Vdd supply. Static timing analysis is done for the entire design using the measured CLB delay values. CLBs on a path that does not meet the design delay are marked as failing CLBs. A greedy selection algorithm then assigns a high-Vdd supply to the failing CLB that most improves the worst negative slack (WNS). Static timing analysis is repeated for the entire design and the greedy selection algorithm is run again. This iterative process continues until either all of the paths meet the target delay or all of the failing CLBs have been switched to a high-Vdd voltage supply. The average runtime of TuneFPGA-VAA on the benchmark with the largest number of CLBs (C5315) was 0.4 seconds on a 3.2 GHz Intel Xeon.

6. EXPERIMENTS

We use the ISCAS-85 combinational logic benchmarks to study the advantages of post-silicon tunable logic. The benchmarks include both control flow and datapath logic. The benchmarks were analyzed using the TuneFGPA methodology described in Section 5. All of the benchmarks were synthesized, optimized, and then mapped to 4-input LUTs using the RASP version of SIS and FlowMap [13]. Individual CLB delay and power values were randomly assigned using the CLBs simulated in Section 3. These benchmarks have a logic depth from 4 CLBs (C499) to 13 CLBs (C3540). See [4] for results on additional benchmarks.

6.1 Setup for Experiments

Results are summarized in Table 1. Key details about the experimental setup are as follows:

- Our implementation of Li's [6] pre-manufacturing voltage assignment has a 10% guard band. As discussed in more detail in Section 6.3, this was the best guard band that we found.
- Yield, power, and the percentage of high-Vdd supply CLBs are calculated for each target delay. The range of target delays was chosen such that the all high-Vdd has a yield of 50% at the fastest target delay and the all low-Vdd has a yield of 50% at the slowest target delay.
- The figure of merit (FOM) for each voltage assignment is the yield/power ratio, which is then normalized to the TuneFPGA FOM value for each target delay.

6.2 Experimental Results

The key take-aways from the experimental results in Table 1 are as follows:

- Compared to High, TuneFPGA had exactly the same yield; this is to be expected since TuneFPGA can set all CLBs to high-voltage if needed. TuneFPGA improved the geometric mean of the power across all benchmarks by 40%.
- Compared to Li [6], TuneFPGA had better yield and used less power. TuneFPGA improved the geometric mean of the yield/power ratio across all benchmarks by 27%.

Further observations from analyzing the results are as follows:

- Designs with mostly short paths show a greater impact from variations. For the C499 benchmark, which has a logic depth of 4 CLBs, selecting voltages post-manufacturing results in yield increases of up to 10× compared with selecting voltages pre-manufacturing with the same delay target.
- Li [6] pre-manufacturing assignment yields are not always monotonically increasing because each target delay has a custom high-Vdd map. Therefore, some target delays will have a better yield for a given assignment map than other target delays.
- Since the Li [6] pre-manufacturing assignment has a 10% guard band, power values for a given target delay are usually more closely matched to a TuneFPGA assignment with a 10% guard band as well. C3540 has a pre-manufacturing power value of 42.2 mW at 7.99 ns, while the TuneFPGA power is 41.6 mW at 7.05 ns.
- High has a large jump in yield between the fastest target and the second fastest target. For C499 with High, a 2.00 ns target has a yield of 50% and a 2.32 ns target has a yield of 100%. Figure 2 shows that high-Vdd CLBs have delay variations of about 0.20 ns. Therefore, small changes in target delay effectively provide margin for high-Vdd CLB delay variations.

6.3 Guard Banding Voltage Selection

As a comparison to TuneFPGA, we also experimented with a pre-manufacturing dual-Vdd voltage assignment based on the work of Li *et al.* [6]. Li's work performed voltage assignment based on power sensitivity. Our implementation of Li's voltage assignment uses the same methodology as TuneFPGA except that the CLB delays use nominal transistor parameter values. The list of CLBs that use a high-Vdd supply is then saved for individual chip configuration.

In order to improve the yield of the pre-manufacturing voltage assignment, the target delay was given a 10% guard band. For example, if the target delay after manufacturing was 5 ns, then the pre-manufacturing voltage assignment would switch enough cells so that the design would have at most 4.5 ns of delay when using nominal transistor values. Since the cell delays are discrete values, the actual target delay would most likely be less than 4.5 ns. One example is that C499 has the same voltage assignment for both 3.29 ns and 3.61 ns target delays; yet, target delay 3.29 ns has a yield of 10%, while target delay 3.61 ns has a yield of 90%. Guard band values from 0% to 20% were used, with 10% resulting in the best yield/power ratio. Tables with results of these other guard band values can be found at [4].

It is noteworthy that designs with no guard band led to abysmal yield. For C5315, using *nominal* transistor values, a target delay of 4.90 ns is always achievable using pre-manufacturing assignment. However, with process variation, the yield drops. A guard band of 10% results in a pre-manufacturing yield of 87% [4]. A 5% guard band has a yield of 64%, while no guard band has just a 3% yield.

Table 1: Benchmark results using 4 different voltage assignments. Each benchmark was instantiated on 1000 chips, each of which have independent variations. TuneFPGA: post-manufacturing voltage assignment using TuneFPGA. Li [6]: our implementation of Li's pre-manufacturing dual-Vdd voltage assignment. All High: every CLB is connected to the high-Vdd supply. All Low: every CLB is connected to the low-Vdd supply.

(a) C499: Implemented with 74 CLBs.

Target Delay (ns)	Yield (%)				Average Power (mW)				Average High-Vdd CLBs (% of Total)				Normalized Yield/Power Ratio			
	Tune FPGA	Li [6]	All High	All Low	Tune FPGA	Li [6]	All High	All Low	Tune FPGA	Li [6]	All High	All Low	Tune FPGA	Li [6]	All High	All Low
2.00	50.6	50.6	50.6	0	10.4	10.4	10.4	n/a	99.9	100	100	n/a	1	1	1	n/a
2.32	100	100	100	0	9.8	10.4	10.4	n/a	89.1	100	100	n/a	1	0.95	0.95	n/a
2.64	100	95.1	100	0	9.0	9.7	10.4	n/a	72.9	86.5	100	n/a	1	0.88	0.86	n/a
2.97	100	60.5	100	0	8.0	9.1	10.4	n/a	54.6	75.7	100	n/a	1	0.53	0.77	n/a
3.29	100	9.9	100	0	6.9	7.4	10.4	n/a	33.5	43.2	100	n/a	1	0.09	0.67	n/a
3.61	100	89.8	100	0.1	5.7	7.4	10.4	5.0	9.7	43.2	100	0	1	0.69	0.55	0
3.93	100	50.1	100	50.1	5.2	5.2	10.4	5.2	0.9	0	100	0	1	0.50	0.50	0.50

(b) C3540: Implemented with 509 CLBs.

Target Delay (ns)	Yield (%)				Average Power (mW)				Average High-Vdd CLBs (% of Total)				Normalized Yield/Power Ratio			
	Tune FPGA	Li [6]	All High	All Low	Tune FPGA	Li [6]	All High	All Low	Tune FPGA	Li [6]	All High	All Low	Tune FPGA	Li [6]	All High	All Low
6.12	50.0	50.0	50.0	0	45.3	45.8	71.3	n/a	27.3	28.7	100	n/a	1	0.99	0.63	n/a
7.05	100	99.8	100	0	41.6	45.0	71.3	n/a	17.0	26.5	100	n/a	1	0.92	0.58	n/a
7.99	100	99.7	100	0	39.1	42.2	71.3	n/a	10.0	18.7	100	n/a	1	0.92	0.55	n/a
8.92	100	99.6	100	0	37.5	40.7	71.3	n/a	5.6	14.5	100	n/a	1	0.92	0.53	n/a
9.85	100	99.6	100	0	36.5	39.9	71.3	n/a	2.7	12.2	100	n/a	1	0.91	0.51	n/a
10.78	100	91.5	100	0	35.7	37.8	71.3	n/a	0.7	6.3	100	n/a	1	0.87	0.50	n/a
11.71	100	90.5	100	50.0	35.5	35.6	71.3	35.5	0.1	0.2	100	0	1	0.90	0.50	0.50

7. RELATED WORK

Besides the pre-manufacturing guard banding work described in Section 6.3, other related work includes adding adaptive body bias for each CLB [10], performing custom placement and routing for each chip [14], or having multiple pre-defined placements for each chip [15]. Adaptive body biasing addresses leakage power primarily, whereas our supply voltage selection technique reduces both switching and leakage power. Additionally, because our method tunes individual chips, TuneFPGA only requires only one time consuming placement and routing configuration for all of the chips.

8. CONCLUSION AND FUTURE WORK

TuneFPGA makes it possible to more tightly meet the design goals while achieving *additional* yield increases and power decreases. TuneFPGA is dependent only on a post-silicon delay map; in particular, it is independent of the delay distribution and correlations between CLB delays. While FPGA interconnect is not addressed in the current work, we realize that interconnect contributes a large portion of the total FPGA delay and power [16]. We anticipate that these TuneFPGA techniques will transfer to interconnect in a similar manner as CLBs. Future work will also investigate whether there are any additional gains from modeling correlated inter-chip variation in addition to the presented uncorrelated intra-chip variation.

9. REFERENCES

[1] S. Borkar *et al.*, "Parameter Variations and Impact on Circuits and Microarchitecture," *Design Automation Conference*, 2003.

[2] S. Rusu *et al.*, "A 65-nm Dual-Core Multithreaded Xeon Processor with 16-MB L3 Cache," *IEEE J. Solid-State Circuits*, 2007.

[3] S. Nassif, "Modeling and Analysis of Manufacturing Variations," *IEEE Custom Integrated Circuits Conference*, 2001.

[4] `http://TuneFPGA.googlepages.com`.

[5] S. Kulkarni *et al.*, "High performance level conversion for Dual Vdd esign," *IEEE Trans. VLSI Syst.*, 2004.

[6] F. Li *et al.*, "Field Programmability of Supply Voltages for FPGA Power Reduction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2007.

[7] Y. Cao *et al.*, "New Paradigm of Predictive MOSFET and Interconnect Modeling for Early Circuit Simulation," *IEEE Custom Integrated Circuits Conference*, 2000.

[8] A. Srivastava *et al.*, "Accurate and Efficient Gate-Level Parametric Yield Estimation Considering Correlated Variations in Leakage Power and Performance," *Design Automation Conference*, 2005.

[9] J. Dorsey *et al.*, "An Integrated Quad-Core Opteron Processor," *IEEE International Solid-State Circuits Conference*, 2007.

[10] G. Nabaa *et al.*, "An adaptive FPGA architecture with process variation compensation and reduced leakage," *Design Automation Conference*, 2006.

[11] D. Ernst *et al.*, "Razor: Circuit-Level Correction of Timing Errors for Low-Power Operation," *IEEE Micro*, 2004.

[12] S. Dhar *et al.*, "Closed-loop adaptive voltage scaling controller for standard-cell ASICs," *International Symposium on Low Power Electronics and Design*, 2002.

[13] J. Cong *et al.*, "RASP: A General Logic Synthesis System for SRAM-Based FPGAs," *International Symposium on Field Programmable Gate Arrays*, 1996.

[14] K. Katsuki *et al.*, "A Yield and Speed Enhancement Scheme under Within-Die Variations on 90nm LUT Array," *IEEE Custom Integrated Circuits Conference*, 2005.

[15] Y. Matsumoto *et al.*, "Performance and Yield Enhancement of FPGAs with Within-Die Variation using Multiple Configurations," *International Symposium on Field Programmable Gate Arrays*, 2007.

[16] T. Tuan *et al.*, "A 90nm low-power FPGA for battery-powered applications," *International Symposium on Field Programmable Gate Arrays*, 2006.