

Creating a Power-aware Structured ASIC

R. Reed Taylor
Carnegie Mellon University
Department of ECE
Pittsburgh, PA 15217
rt2i@ece.cmu.edu

Herman Schmit
Carnegie Mellon University
Department of ECE
Pittsburgh, PA 15217
herman@ece.cmu.edu

ABSTRACT

In an attempt to enable the cost-effective production of low- and mid-volume application-specific chips, researchers have proposed a number of so-called structured ASIC architectures. These architectures represent a departure from traditional standard-cell-based ASIC designs in favor of techniques which present more physical and structural regularity. This paper presents circuits which provide power-performance flexibility in this regular, structured ASIC environment. These circuits, which employ gate sizing and voltage scaling for energy efficiency, enable delay-constrained power optimization to be performed for structured ASIC designs.

Categories and Subject Descriptors: B.6.1 [Logic Design]: Design styles—Logic Arrays; B.6.3 [Logic Design]: Design aids—Optimization

General Terms: Performance, Design

Keywords: Structured ASIC, VPGA, Low-Power, Voltage Scaling, Gate Sizing, Power Optimization

1. INTRODUCTION

Integrated circuit manufacturing processes with feature sizes below 100nm pose a number of technological challenges to the ASIC designer. In particular, these new processes present increased manufacturing costs, substantial process variation, and exceedingly complex design rules compared to their technological predecessors.

In an attempt to mitigate these factors and to enable the cost-effective production of low- and mid-volume application-specific chips, researchers have proposed a number of so-called structured ASIC architectures. These architectures represent a departure from traditional standard-cell-based ASIC designs in favor of techniques which present more physical and structural regularity. This regularity can be leveraged in an effort to compensate for process variations, complex design rules, and rising manufacturing costs. This same regularity, however, strips structured ASICs of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED'04, August 9–11, 2004, Newport Beach, California, USA.
Copyright 2004 ACM 1-58113-929-2/04/0008 ...\$5.00.

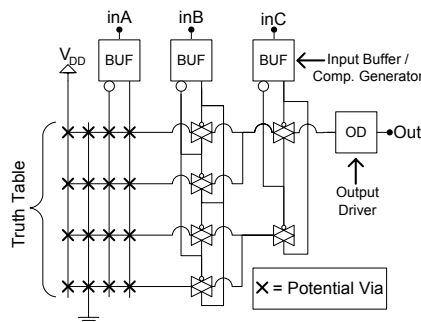


Figure 1: VPGA Logic Block

the power-performance flexibility which can be exploited (as in standard-cell ASIC design flows) in an effort to conserve power.

In this paper, we will introduce a number of circuits which restore power-performance flexibility to structured ASICs using gate sizing and voltage scaling techniques.

1.1 A Structured ASIC Family: the VPGA

While the range of structured ASIC architectures is still under active exploration, this paper will use the Via-Patterned Gate Array, or VPGA, as the prototypical structured ASIC architecture. The VPGA architecture is introduced in [4] and [7].

We will further assume that a single, 3-input lookup table (3-LUT) forms the heart of all logic cells. An array of these cells will make up the computational fabric of our VPGA. The findings relevant to this class of VPGAs should be extensible to other structured ASIC architectures, including those with multiple, heterogeneous logic units, regardless of their internal structure.

Architecturally, a VPGA resembles a traditional FPGA in a number of ways: they are both generally conceived as two-dimensional arrays of programmable logic units which can be selectively connected through the use of a fixed routing architecture with programmable switchboxes at possible junction points.

Unlike FPGAs, VPGAs are not field-programmable devices. VPGAs are “programmed,” or patterned, during latter stages of the manufacturing process by the selective placement of inter-layer vias. These vias form the required logical connections to complete and configure the device.

An example of via-patterning can be seen in Figure 1, where the truth-table for the logical behavior of the LUT

is determined entirely through the placement of vias on the potential locations, marked with “X”. The complement-generating input buffers are marked “BUF”, and the output driver is marked “OD”.

Programming through selective via placement makes VP-GAs much more dense, efficient, and high-performance than FPGAs, which are programmed using SRAM cells and pass-gates. In fact, VPGAs exhibit performance which is comparable to that of contemporary standard-cell ASICs.

2. CIRCUIT STRUCTURES FOR POWER OPTIMIZATION

The variety of gate sizes available in a standard-cell library provides multiple power-delay points for each gate. A gate sizing tool selects from this group the appropriate size for each gate instance, according to the performance needs of the delay paths that pass through that instance. A comparison of some recent approaches to this problem is presented in [2].

In a “power-unaware” structured ASIC, the single, repeated logic block will provide only one point on the power-delay curve for each logic function. Hence, without adding some circuit structures which impart power-performance flexibility, it will be impossible to perform in-place optimization techniques for power minimization resembling those used in SC-ASICs.

One possible workaround for this inflexibility would be to create a heterogeneous structured ASIC, where, for example, one in four logic blocks includes a larger driver. It would then become the job of the automatic placement algorithm to map critical cells to high-drive locations. Algorithms for this type of mapping can be found in [1].

This paper will instead focus on circuit structures which can be used to provide flexibility even in a homogeneous VPGA by using only via-patterning techniques to select between power-performance modes. No modifications to the underlying circuit structure (e.g. transistor sizes) will be possible in a VPGA once the architecture is fixed; hence, just as the logic function implemented by the LUT in each VPGA cell is “programmed” through via placement (as seen in Figure 1), so must the mode of power operation be “via-programmable” for each cell.

2.1 Gate Sizing for Power Optimization

One method for creating flexibility in power-performance for a structured ASIC logic block is to emulate the same method of gate sizing which is frequently used in standard-cell design flows. This can be accomplished with a VPGA logic cell that provides the ability to choose from a range of gate sizes on its output driver. A VPGA output driver circuit (labeled “OD” in Figure 1) which provides this functionality is detailed in Figure 2.

Using selective via placement, this circuit can be configured to use either one or two parallel output drivers. Transistors M1 and M2 serve as the primary output driver. When vias are added at the three potential locations marked with “H”, the transistors M3 and M4 form an auxiliary, parallel output driver. This will effectively increase the drive strength of the logic block.

These two drivers can be sized to match the expected needs of the VPGA architecture in question. Depending on the needs of the architecture, the fundamental driver

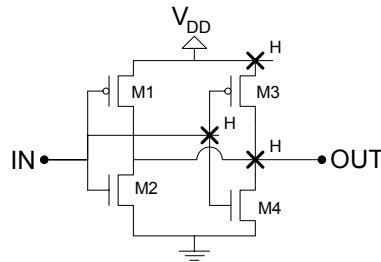


Figure 2: VPGA Output Driver for Gate Sizing

is not necessarily a minimum-sized inverter, and that the auxiliary driver does not need to be the same size as the primary driver. Indeed, it is easy to envision a similar circuit which contains $N - 1$ separately selectable auxiliary drivers, conveying 2^N possible driver configurations. For the analysis in this paper, the primary and auxiliary output drivers are assumed to be the same size, and we will refer to the two resultant modes of operation as offering “low” and “high” drive strengths.

This gate-sizing circuit meets our requirement that it be configurable solely through the placement of vias. When the vias marked “H” are absent, the unused auxiliary driver is completely severed from the circuit and the power supply, eliminating any leakage current or unnecessary load capacitance.

Unlike its standard cell counterpart, this via-patterned gate sizing approach does not permit any area recovery through gate selection – the auxiliary driver remains on the chip, whether it is used or not. Still, this approach is quite simple and straightforward, and is not area intensive.

2.2 Voltage Scaling for Power Optimization

It is well-known that reductions in the power supply voltage V_{DD} of a CMOS circuit convey quadratic savings in overall dynamic power dissipation. Unfortunately, reductions in the power supply voltage without modifications to the transistor threshold voltages V_{TN} and V_{TP} will also cause the logic performance to degrade. As a result, chip-wide reductions in supply voltage will tend to produce unacceptable timing violations on critical paths through the circuit.

To remedy this, we will employ a technique known as *selective voltage scaling*, where the supply voltage is reduced only for components with substantial positive timing slack. In this manner, any performance degradation will be confined to non-critical paths, allowing the chip-wide clock frequency (and overall system performance) to be preserved.

Power-saving techniques using selective voltage scaling have been proposed in a number of previous contexts. One widely-known application of selective voltage scaling is called Clustered Voltage Scaling, or CVS. Introduced in [8], CVS implements selective voltage scaling specifically in the context of SC-ASICs. POGA, or *Power Optimization for Gate Arrays*, is an approach to selective voltage scaling specifically designed for application to VPGAs. This approach is detailed in [6].

2.3 Dual-supply Mixed Swing VPGA

The use of selective voltage scaling will typically create many points in a circuit where signals with differing voltage swings will interact. At these points, there is a risk that

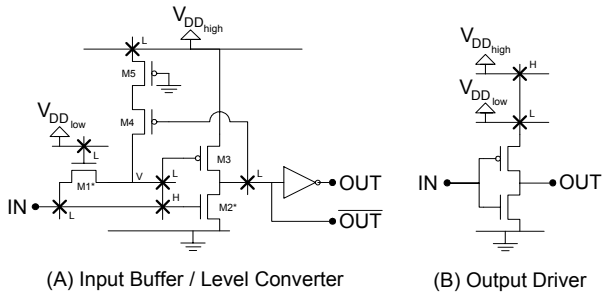


Figure 3: Dual-Supply Level Converter and Output Driver

under-driven PMOS transistors may not be able to fully enter cut-off, and static currents may flow through these logic gates. This static current flow would rapidly undermine any achieved power savings, and must therefore be prevented through the use of *voltage level converters*.

The needed ingredients, then, for enabling selective voltage scaling in a VPGA are: a circuit which will produce buffered, complementary inputs, and which can perform level conversion if needed (this circuit is labeled “BUF” in Figure 1); and an output driver which can drive the logic block output in either high- or low-swing mode (labeled “OD” in Figure 1).

In the past, level converters were typically constructed from DCVS logic; however, in [5], some newer and more sophisticated circuit structures for voltage level conversion were introduced. These newer level converters were shown to be faster and more efficient than their DCVSL counterparts. One such circuit, modified and optimized for use in a VPGA environment, can be seen in Figure 3(A).

This level converter is based on the dual-supply level converter “STR1” presented in [5], augmented to allow for optimized mixed-swing VPGA operation. When vias are placed just at the locations marked “L”, the transistors M1, M4, and M5 all work to perform level conversion on the input. This restores node V to full-swing operation, preventing static current flow through transistors M2 and M3.

The performance of the input buffer in this mode is acceptable; however, in a mixed-swing environment, it will often be the case that an input to a logic block will, in fact, be a full-swing signal. In this case, we would like to use via patterning to eliminate the performance penalty associated with level conversion, to the extent possible.

This can be accomplished by instead placing a via at the location marked “H”. In this case, the transistors M1, M4, and M5 will all be disconnected from our circuit and from the voltage supply rails; therefore, they will have no detrimental impact on performance. Configured in this manner, the input buffer looks like a simple series of two inverters, both powered by V_{DDH} .

The output driver shown in Figure 3(B) is very simple. By placing a via at the locations marked “H” or “L”, it can be configured to use either V_{DDH} or V_{DDL} as its supply, producing a corresponding high- or low-swing output.

In cases where the inputs and the output are in low-swing mode, it is straightforward to enable a third, highly energy-efficient configuration of the input buffer: it can function without the use of the level-converter (transistors M1, M4, and M5), yet be powered entirely by V_{DDL} . This eliminates

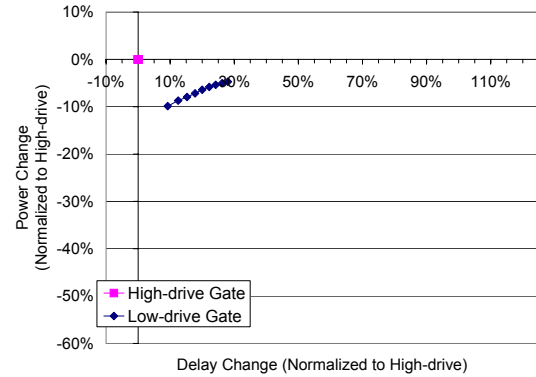


Figure 4: Power-Delay Plot: Gate Sizing

the situation where an input is needlessly converted to high-swing operation for transmission through the LUT, only to have the output driver return the signal to low-swing mode. (The potential vias which enable this mode are not shown in the figure.)

At the circuit level, the externally-regulated level-converting buffers and output driver are relatively compact, consisting of a handful of additional transistors. The primary source of area overhead for the dual-supply mixed swing logic comes from the distribution of V_{DDL} , the reduced power supply.

3. RESULTS

The goal of this work is to produce VPGA-compatible circuit structures which lend exploitable power-performance flexibility to the basic logic block. In order to directly evaluate this flexibility, we will examine various power-performance points for each configuration of each presented technique, over a range of load capacitances.

3.1 Circuit Analysis

Circuits were evaluated (using hspice) over a series of single-input transitions, and for a range of load capacitances. From these simulations, we obtained timing and power data including worst-case propagation delays, and total power consumed over a series of transitions.

To compare the flexibility provided by each technique, we will look at normalized power-delay difference plots, such as the one seen in Figure 4. In these plots, the X-axis represents the normalized change in delay: a configuration which is slower than the baseline will appear to the right of the origin, and one which is faster than the baseline will appear to the left of the origin. The Y-axis represents the normalized change in power dissipation: a configuration which uses less power than the baseline will appear below the origin, and one which uses more power than the baseline will appear above the origin. In these plots, each point represents a measurement for a particular load capacitance. The power and delay values for each such point are relative to the baseline design when driving that same load. These points representing the baseline design are visible as a square centered exactly on the origin.

Our VPGA-compatible gate sizing approach (described in Section 2.1) is analyzed in Figure 4. The “high-drive” configuration forms the baseline configuration for this plot. Power-delay points for the “low-drive” configuration are in

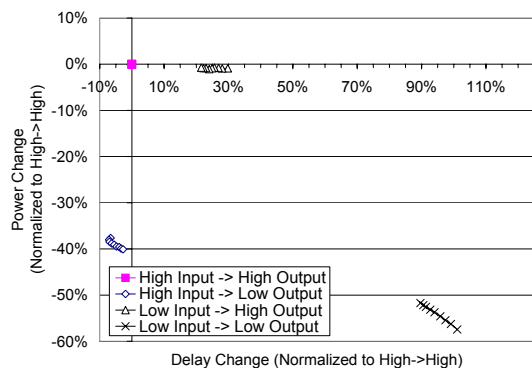


Figure 5: Power-Delay Plot: Dual-Supply Mixed Swing

the bottom-right quadrant. From this, we can tell that the weak driver configuration will use between 5% and 10% less power than the high-drive configuration, while taking 10-30% longer per transition. Again, each data point shown corresponds to a particular load capacitance.

The results seen in this plot demonstrate exactly the type of flexibility we were hoping to obtain. Generally speaking, we would like for any low-power configuration to appear in the lower-right quadrant of this plot, meaning that the configuration saves power in exchange for some sacrificed performance.

A power optimization tool, with the characterized high- and low-drive configurations made available to it, could now choose between two power-delay points for any given load capacitance: the high-drive baseline point, or the corresponding low-drive point. Any traditional algorithm for gate sizing, such as those in [2], should be able to handle this task in a straightforward manner, as will the VPGA-specific algorithm presented in [6].

In Figure 5, we see the same type of results, generated for our dual-supply mixed-swing circuit, presented in Section 2.3. This plot presents four sets of data points, corresponding to the possible low- and high-swing configurations of the gate inputs and outputs. The high-input, high-output configuration serves as the baseline. (For clarity, our results assume the inputs to be either all low- or all high-swing; however, it would be simple to perform the same analysis for all 16 possible input/output combinations.)

The high-to-low configuration uses 35 - 40% less power, and is approximately 5% faster than the baseline. The low-to-low configuration uses approximately 51 - 57% less power, with 90% to 100% delay penalty. Finally, the low-to-high configuration uses about 1% less power, with roughly 20 - 30% delay penalty.

Clearly, the use of a mixed-swing structure to reduce power consumption is not as straightforward as the use of the gate sizing structure. For example, if an optimization tool migrates a cell from a high-to-high to a high-to-low configuration, it will inadvertently force other gates in the design to become low-to-high nodes. The optimization tool must be ready to incorporate these changes, and take the resultant side-effects into account.

3.2 Conclusions

In this paper, we have introduced several methods for enabling power optimization in VPGAs. Our approaches provide power-performance flexibility without the need to violate the unique rules of the VPGA environment. This gives us an “energy-aware” structured ASIC architecture.

Gate sizing, the simpler of the two approaches, requires very little area overhead, as a maximum-sized output driver may simply be “cut in half” and prepared for via patterning. Gate sizing in VPGAs is clearly a viable approach.

The mixed-swing approach may incur a significant area penalty due to the need to route the reduced voltage supply. Nonetheless, it performs very well in terms of power-performance flexibility. The low-swing output modes are, of course, the most efficient.

Perhaps the best single approach to implementing power savings in VPGAs would be to combine the gate sizing and ER mixed-swing approaches into a single, multi-faceted approach. An algorithm which enables this type of simultaneous optimization is presented in [3].

4. REFERENCES

- [1] J. Cong and S. Xu. Delay-oriented technology mapping for heterogeneous fpgas with bounded resources. In *Proc. IEEE/ACM International Conference on Computer-aided Design (ICCAD 98)*, pages 40–44, 1998.
- [2] O. Coudert, R. Haddad, and S. Manne. New algorithms for gate sizing: A comparative study. In *Proc. 33rd IEEE/ACM Design Automation Conference (DAC 96)*, pages 734–739, 1996.
- [3] A. Nayak, M. Haldar, P. Banerjee, C. Chen, and M. Sarrafzadeh. Power optimization of delay constrained circuits. In *Proc. Application Specific Integrated Circuit/System-on-a-Chip Conference (ASCI/SOC 2000)*, September 2000.
- [4] L. Pileggi, A. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, and K. Tong. Exploring regular fabrics to optimize the performance-cost trade-off. In *Proc. 40th ACM/IEEE Design Automation Conference (DAC 2003)*, pages 782–787. ACM Press, 2003.
- [5] R. Puri, L. Stok, J. Cohn, D. Kung, D. Pan, D. Sylvester, A. Srivastava, and S. Kulkarni. Pushing asic performance in a power envelope. In *Proc. 40th IEEE/ACM Design Automation Conference (DAC 2003)*, pages 788–793, June 2003.
- [6] R. Taylor and H. Schmit. Enabling energy efficiency in via-patterned gate array devices. In *Proc. 41st IEEE/ACM Design Automation Conference (DAC 2004)*, June 2004. To be published in the proceedings.
- [7] K. Tong, V. Kheterpal, V. Rovner, and L. Pileggi. Regular logic fabrics for a via patterned gate array (vpga). In *Proc. IEEE Custom Integrated Circuits Conference (CICC 2003)*, September 2003.
- [8] K. Usami and M. Horowitz. Clustered voltage scaling for low-power design. In *Proc. International Symposium on Low Power Design (ISLPD 95)*, pages 3–8, April 1995.