# Design of a Single Layer Programmable Structured ASIC Library

Thomas C.P. Chau, David W.L. Wu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
{cpchau, ylw}@cse.cuhk.edu.hk

Yan-Qing Ai, Brian P.W. Chan, Sam M.H. Ho, Oscar K.L. Lau, Steve C.L. Yuen, Kong-Pang Pun, Oliver C.S. Choy
Department of Electronic Engineering
The Chinese University of Hong Kong
{yqai, pwchan, mhho, kllau, clyuen, kppun, cschoy}@ee.cuhk.edu.hk

Philip H.W. Leong
School of Electrical and Information Engineering
University of Sydney
Australia
phwl@ee.usyd.edu.au

***Abstract*— A Structured Application-specific Integrated Circuit (SASIC) is a programmable fabric in which a small set of masks are customized for a particular application, serving to reduce the associated non-recurring engineering cost (NRE). In this paper we describe the implementation of a SASIC logic cell which is programmable via a single metal layer. A SASIC fabric prototype is fabricated and all implemented functions are verified on silicon. Experimental measurement verifies correct operation of our SASIC with a clock frequency of over 250MHz.**

***Keywords-application specific integrated circuits; digital integrated circuits; field programmable gate arrays***

## I. INTORDUCTION

Application-specific integrated circuits (ASICs) and field programmable gate arrays (FPGAs) are the two main technologies for implementing digital designs. ASICs have advantages in terms of area, power, and speed but an expensive set of masks must be created for every design. In contrast, the mask costs are amortized over all of the users in FPGAs at the expense of lower performance and higher power consumption. A SASIC is a programmable fabric offering a middle ground between the two in which a small set of masks is customized for a particular application, serving to reduce the associated NRE and turn-around time [11].

In the UMC 0.13μm single-poly 8-metal CMOS process, a full mask set for a standard cell ASIC consists of 40 layers [18] but a SASIC needs only one-tenth of those layers for configuration. Compared to FPGAs, SASICs offer improved density and speed as logic configuration and programmable routing can be implemented directly using metal and vias instead of static ram and multiplexers. As a result, SASICs are a compelling choice for medium volume production [3].

There have been previously reported SASIC designs but many are commercial and details of their implementations have not been published [6][9][11][12][13][16]. Of the reported ones in academia, such as [4][5][7][8][10], most have not been fabricated and proven on silicon.

In this paper, we describe a SASIC library in which a single type of logic block can be configured for both combinatorial logic functions and sequential functions. In contrast to the work of [14] and [15] having fixed metal routing structure, our design adopts standard-cell ASIC approaches so there is no predefined routing fabric and clock tree. This allows critical path optimization and multiple clock domain support. The design flow is also highly compatible with industrial-level commercial tools.

The key contributions of this paper are:

- A SASIC logic cell design programmable using a metal layer.
- A SASIC fabric that makes use of the logic cell library.
- A fabricated SASIC prototype for verification.

In the remainder of this paper, we present and evaluate our design. In Section II we describe the structure and the configuration framework of our SASIC library. In Section III, we present the evaluation fabric and circuits. Section IV shows the performance evaluation. Finally, Section V gives conclusions.

## II. SASIC LIBRARY

Unlike standard-cell ASICs whose cells are placed in arbitrary positions, the logic cells in a SASIC are constrained to predefined sites. Thus a universal logic cell structure that can be customized to implement all kind of functions is

978-1-4244-6613-9/10/$26.00 ©2010 IEEE

required. Meanwhile, the effort of customization, namely the number of programming mask, should be minimized.

### A. Logic Cell Structure

The logic cell is implemented as a lookup table style, as shown in Figure 1. This approach is common in FPGAs and SASICs and has been studied in our previous paper [1]. The logic cell has three major components: input inverters, a function generator and an output stage. Its regular structure makes it easily programmable by a single metal layer. The components are described in more detail below.

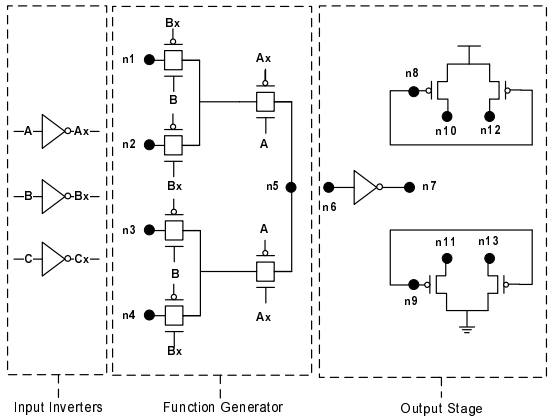


Figure 1. Logic cell structure

1) *Input inverters:* Generate complementary signals (Ax, Bx, Cx) from the inputs (A, B, C).

2) *Function generator:* This is a 4-to-1 multiplexer with n1-n4 being programmable to power, ground, C and Cx via metal strips. The function generator is implemented using transmission gates to enable operation at low supply voltage [17].

3) *Output stage:* Contains a CMOS inverter, a pair of NMOS transistors and a pair of PMOS transistors. These are configurable to three types of CMOS output buffers with driving strength 1x, 1.67x and 3.33x, as well as several special configurations described below. Referring to Fig. 1, the logic cell has 1x driving strength if n7 is used as the output. If n7 is connected to n8 and n9, n10 is connected to n11, a driving strength of 1.67x is produced at n10. If n7 is connected to n8 and n9, n10 to n13 are all connected together, the logic cell has a driving strength of 3.3x at n10.

### B. Combinatorial Functions

Logic cells are configurable to numerous combinatorial logic elements:

1) *2 and 3 inputs functions:* 76 distinct logic functions, each with three driving strengths, are realized by connecting n1-n4 of the function generator to power, ground, C or Cx.

2) *Buffers and inverters:* Buffers and inverters with three different driving strengths are formed by treating n6 as input and configuring the output stage as described in Section II.A.

3) *Tri-state buffers and inverters:* There are two types of tri-state buffers with different driving strengths. The strong one consists of two logic cells as shown in Figure 2a. When the En pin is logic 1, the tri-state buffer acts as a buffer, otherwise the NMOS and PMOS pair are both off and the output is high impedence. Another type of tri-state inverter is configurable by one logic cell but the driving strength is weaker. It uses transmission gates to block input signal such that the output can be high impedence.

4) *Filler cell:* As shown in Figure 2b, the filler cell acts as a supply rail decoupling capacitor to occupy white space on the chip.

5) *Tie cells:* Tie1 provides a DC level of logic 1 while Tie0 provides a DC level of logic 0. An example of Tie1 configuration is shown in Figure 2c.

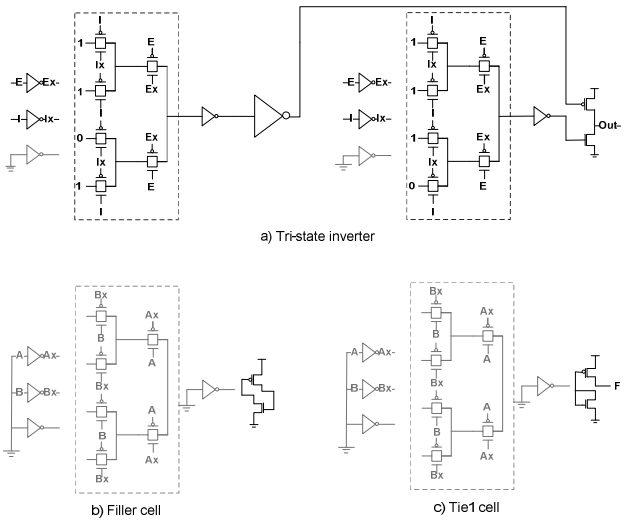


Figure 2. Logic cells configured as combinatorial functions

### C. Sequential Functions

The logic cell can implement two sequential functions, a D-latch and a gated-clock cell. The D-latch makes use of a feedback path to hold its state. The gated clock cell is realized by combining a D-latch and controlling logic.

### D. Layout

The layout of a logic cell is illustrated in Figure 3. Its dimension is 10.4μm x 4.8μm. The third layer of metal (M3) connects the fixed vias to program the function generator and the output stage. Since M3 is also used in routing, the layout is optimized to minimize blocking. 50% of M3 layers are reserved as routing channels which are the columns not being occupied by fixed vias.

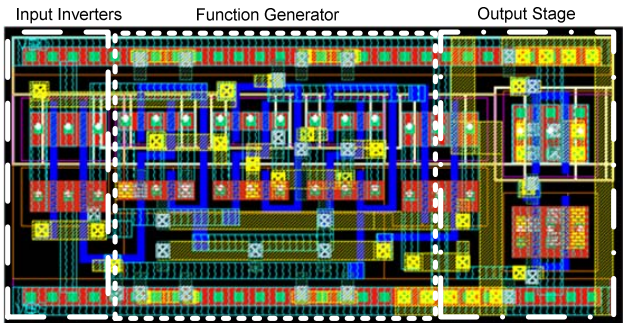


Figure 3. Logic cell layout

### E. *Library Generation*

As described in Section II, the SASIC library contains a number of combinatorial and sequential functions with different driving strengths. There are 244 cells in total. It would be inefficient and inflexible if they are all designed by hand. Thus, we developed an automatic library generation flow.

Firstly, a logic cell layout template is implemented which acts as an empty shell for layout generation. Then, a number of customization templates are prepared, namely the metal strips which connect the fixed vias. These two steps produce all the layout components used in library layout.

Meanwhile, all the logic cell functions are generated. As the logic cell has three inputs, there are $2^3$=8 input combinations, with $2^8$=256 output functions which each of them is described by a truth table. UC Berkeley Espresso is used to minimize the function of each truth table. After that, Synopsys' Formality compares the functions of all the truth table and eliminates duplicated functions, producing a unique list of functions.

Combining the layout components and the function list, all the logic cell layouts and Verilog netlists are generated. Encounter Library Characterizer (ELC) characterizes the generated logic cells, and records their timing and area information in the Liberty format. Finally, the generated library is verified by design rule check (DRC), layout versus schematic (LVS) and logic equivalence check (LEC).

## III. Evaluation Prototype

To evaluate our SASIC logic cell library, we have fabricated a SASIC prototype using the designed logic cells.

### A. *SASIC Fabric*

The chip-level architecture of our SASIC fabric contains flip-flops, embedded blocks and input/output pins.

1) *Flip-flop:* Dedicated universal D-flip-flops with scan support are used in our SASIC. Although a master-slave D-flip-flop can be implemented using two logic cells configured as a D-latch, a dedicated flip-flop is faster and has more features. The flip-flop is D-type with set, reset, enable, scan and dual-outputs. It is supplemented an antenna cell which fixs antenna rule violations caused by using long routing wires.

2) *Embedded Blocks:* Logic cells in our SASIC are fine-grained logic. They are flexible at the expense of area, speed and power overhead. Hence, coarse-grained elements known as embedded blocks are implemented. Our SASIC is able to incorporate numerous types of embedded blocks, such as RAMs, arithmetic units and processors. In our latest prototype, we only included block RAMs.

3) *Routing and Clock Domain:* Unlike FPGAs and some SASICs [14][15], our SASIC has no prefabricated routing pattern. The logic and clock signals are routed flexibly on at least two metal layers (M3 and M4). This routing scheme has advantages such as compatibility with industrial standard-cell ASIC design flow, support of multiple clock domains and low-power gated clock design.

The fabric is fabricated in UMC 0.13µm CMOS process. The back-end design flow is Cadence-based, in which the logic cell library is implemented by Virtuoso, simulated by Encounter Library Characterizer and Spectre mixed-signal Simulator, placed and routed by SoC Encounter.

Our SASIC prototype consists of 1578 logic cells, 106 flip-flops, 5 RAM blocks and 56 I/O pins.

### B. *Evaluation Circuits*

To evaluate the performance, a test unit (tunit) and an extracted part of an active dynamic backlight controller (peak unit) are mapped to the prototype.

The tunit consists of a 19-stage ring oscillator, an 8-bit counter and an 8x8 multiplier, and a scan chain is inserted for functional test. The ring oscillator is synthesized from 19 NAND3 logic cells cascaded together while the counter and multiplier are synthesized from RTL written by Verilog. The counter counts from 0 to 255 in every cycle. The multiplier is pipelined and it takes two cycles to input data. The peak unit computes the LED brightness at 180 small regions of each full HD 1080p resolution video frame at 1.5Gbps. The target speed of the counter and multiplier is 100MHz during logic synthesis, placement and routing.

6 test chips are bonded. The logic cell supply voltage (Vdd) is 1.2V and the voltages of input/output pads are 3.3V. The test chips are plugged into a test board and tested using a HILEVEL ETS868 Griffin Test System (ETS868) and logic analyzer.

## IV. Measurement Results

### A. *Functional Tests*

To test the functionality of tunit, a scan chain was inserted into the counter and multiplier. During physical synthesis, scan flip-flops were stitched into a chain. Then we used Synopsys's TetraMAX ATPG to generate test patterns automatically. Test patterns were shifted into the circuit

through the scan chain and then computation results were shifted out for analysis. All the test chips passed the scan test.

To test the peak unit, 2.3 million lines of test vectors were generated from simulation bench. Using ETS868, the test vectors were fed into the test chips and output results were verified. At Vdd 1.2V, the test chips passed the test.

### *B. Speed*

For the ring oscillator, the frequencies of all test chips were measured using an oscilloscope. The measurement results of all test chips at each Vdd were averaged as shown in Figure 4. At nominal Vdd, the frequency of the ring oscillator (f) is 88.31MHz. Thus the delay of one stage of NAND3 is 1/(2*19*f) = 0.30ns.

For the counter, we generated the clock using a pattern generator and verified the output with a logic analyzer. At each Vdd, the clock-frequency was increased until erroneous outputs were observed. The measurement results of all test chips at each Vdd were averaged as shown in Figure 4. The maximum clock-frequency achieved at nominal Vdd is 131MHz which exceeded the target speed.

For the multiplier, the data was input to the multiplier through a register and results were latched out through another register. The minimum period measured is the combinatorial delay. The reciprocals of delay at each Vdd are shown in Figure 4. The maximum value at nominal Vdd is 285.71MHz.

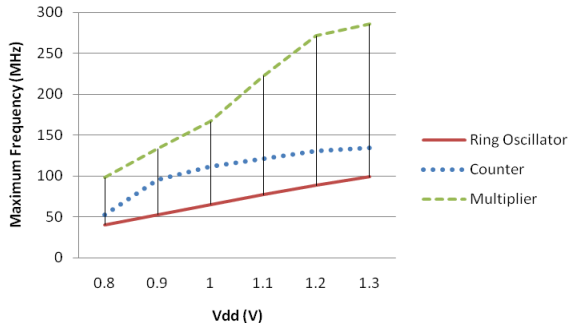


Figure 4. Maximum frequencies of tunit

### *C. Power*

In every test period, the average power of the test chips was measured and illustrated in Figure 5. The power consumption increases linearly with Vdd.

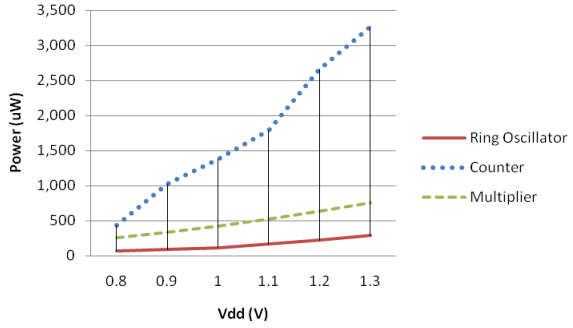


Figure 5. Power consumptions of the tunit

## V. CONCLUSION

In this paper, we introduced a single metal layer programmable logic cell library which is used in the implementation of a SASIC fabric. 6 SASIC prototype test chips were produced. We showed that our test chips can achieve operation above 250MHz under typical conditions and can operate at supply voltages as low as 0.8V. Future research will focus on improving the library generation flow to be process independent and reducing the effort of logic cell characterization.

## ACKNOWLEDGMENT

The authors gratefully acknowledge support from the Innovation Technology Fund (Grant GHP/028/07SZ), the Hong Kong Special Administrative Region, China.

## REFERENCES

[1] T. C. Chau, P. H. Leong, S. M. Ho, B. P. Chan, S. C. Yuen, K.-P. Pun, O. C. Choy, and X. Wang, "A comparison of via-programmable gate array logic cell circuits," in Proc. 17th ACM SIGDA Int. Symp. Field-Programmable Gate Arrays, 2009, pp. 53-62.

[2] T. Pi and P. J. Crotty, "FPGA lookup table with transmission gate structure for reliable low-voltage operation," U.S. Patent 6667635, Dec. 23, 2003.

[3] B.Zahiri, "Structured ASICs: opportunities and challenges," in Proc. 21st Int. Conf. Computer Design, 2003, pp. 404-409.

[4] L.Pileggi, H.Schmit et al., "Exploring regular fabrics to optimize the performance-cost tradeoff," in Proc. 40th Annual Design Automation Conference, 2003, pp. 782-787.

[5] C.Patel, A.Cozzie, H.Shmit and L.Pileggi, "An architectural exploration of via-patterned gate-arrays," in Proc. 2003 Int. Symp. Physical design, 2003, pp. 184-189.

[6] Kun-Cheng Wu and Yu-Wen Tsai, "Structured ASIC: evolution or revolution?," in Proc. 2004 Int. Symp. Physical design, 2004, pp. 103-106.

[7] Y. Ran and M. Marek-Sadowska, "Designing via-configurable logic blocks for regular fabric," IEEE Trans. VLSI, vol.14, pp. 1-14, January 2006.

[8] M.-C. Li, H.-H. Tung, C.-C. Lai, R.-B. Lin, "Standard cell like via-configurable logic block for structured ASICs," in Proc. IEEE Computer Society Annual Symp. VLSI, 2008, pp. 381-386.

[9] T. Okamoto, "Design methodology and tools for NEC Electronics' structured ASIC ISSP," in Proc. Int. Symp. Physical Design, pp. 90-96, 2004.

[10] N. Jayakumar and S. P. Khatri, "A metal and via mask programmable VLSI design methodology using PLA," in Proc. IEEE/ACM Int. conf. Computer-aided design, 2004, pp. 590-594.

[11] Faraday Structured ASIC, http://www.faraday-tech.com/html/products/structuredASIC.html

[12] Altera, HardCopy Structured ASICs, http://www.altera.com/

[13] LSI Logic, RapidChip, http://rapidchip.lsilogic.com/

[14] A. Levinthal and R. Herveille, "FlexASIC structured array: a solution to the DSM challenge," DesignCon 2005.

[15] Z.Orbach, Z. Wurman, R.Zeman, and L. Cooke, "Customizable and programmable cell array," US Patent 6,331790,18 2001

[16] ViAsic, Standard metal solutions for structured ASICs and configurable SOCs, http://viasic.com

[17] T. Pi and P. J. Crotty, "FPGA lookup table with transmission gate structure for reliable low-voltage operation," U.S. Patent 6667635, Dec. 23, 2003.

[18] UMC 0.13µm Technology Overview, http://www.europractice-ic.com/technologies_UMC.php?tech_id=0.13um