# AN 11 GHZ FPGA WITH TEST APPLICATIONS

*Chao You[1], Jong-Ru Guo[1], Michael Chu, Kuan Zhou, Russell P. Kraft, John F. McDonald**

ECSE Department
Rensselaer Polytechnic Institute
110 8th Troy, NY 12180
email: you.chao@gmail.com, chum2@rpi.edu

*Bryan Goda*

Department of Electrical Engineering and Computer Science
US Military Academy
West Point, NY 10996
email: Bryan.Goda@usma.edu

## ABSTRACT

FPGAs have been a popular topic among electrical engineers for over a decade. Modern FPGAs are denser, faster and use less power compared to PLD and CPLD. This paper presents a high speed FPGA that operates in the gigahertz range. An improved reconfigurable circuit is described. State-of-the-art IBM SiGe BiCMOS technology is used to realize this high speed FPGA. A measurement on a fabricated chip proves that this FPGA is capable of operating at 11 GHz, a SONET OC-192 standard, for optical applications.

## 1. INTRODUCTION

A Field Programmable Gate Array (FPGA) is a reconfigurable device. With the advantage of being programmable, FPGAs have gained attention for shortening Design-to-Market times. Besides the configurable feature, FPGAs have speed advantages over general computing devices [1]–[5]. For example, processor designers use FPGAs to emulate their designs and simulate new generation microprocessors [2].

An FPGA's speed can be further increased by using state-of-the-art IBM High Performance (HP) BiCMOS technology. The first microwave range FPGA was reported in 2000 [6], [7]. That FPGA was a Current Mode Logic (CML) version of the Xilinx XC6200 series. That design could not be scaled significantly due to its high power consumption. This paper presents an improved reconfigurable FPGA logic circuit capable of high speed with greatly reduced power consumption. In addition, the new basic cell structure also provides the old XC6200 FPGA with enhanced routing capabilities.

*This work is sponsored by Defense Advanced Research Project Agency (DARPA) on contract number N66001-02-1-8919
1 Both authors have the same contribution to this work.

## 2. BASIC CELL IMPROVEMENT

### 2.1. XC6200 Introduction

This FPGA design uses the Xilinx XC6200 as an initial starting point. The XC6200 was developed for co-processing in embedded DSP system applications by Xilinx. The XC6200 is open-source in both the hardware and software domains [11], which gives our design more resources from which to obtain design information. The XC6200 FPGA is a multiplexer based FPGA. A multiplexer can provide any combinational logic result if inputs are properly selected. Among general-purpose functional blocks, such as NAND gates, Lookup Tables and AND-OR gates, a multiplexer based logic block has shorter gate delays than other types [8], [9]. Another reason for choosing the XC6200 is that a multiplexer based FPGA can be implemented in CML circuits without significant loss of performance.

A block diagram of the XC6200 basic cell is shown in Fig. 1. Each basic cell has eight inputs and four outputs to its neighbors. In every 4×4 block there are shared buses (N4, S4, E4 and W4 in this figure), which are called FastLANEs™. The shared bus connects four cells in the same row or column together. FastLANE is used to deliver a signal into a cell that is inside a gate array. FastLANE can reduce signal buffering and thus use fewer basic cells.

The four FastLANEs' and four neighbor cells' outputs are sent to three 8:1 multiplexers as shown in Fig. 1. The outputs from the 8:1 multiplexers are input signals of the function unit, where the final logic result is computed. The output signal from the function unit is sent to 4:1 multiplexers in four directions. The 4:1 multiplexers can be used to send out function results or redirect a signal from one direction to another, the so-called redirection function. A simplified block diagram is shown in Fig. 2

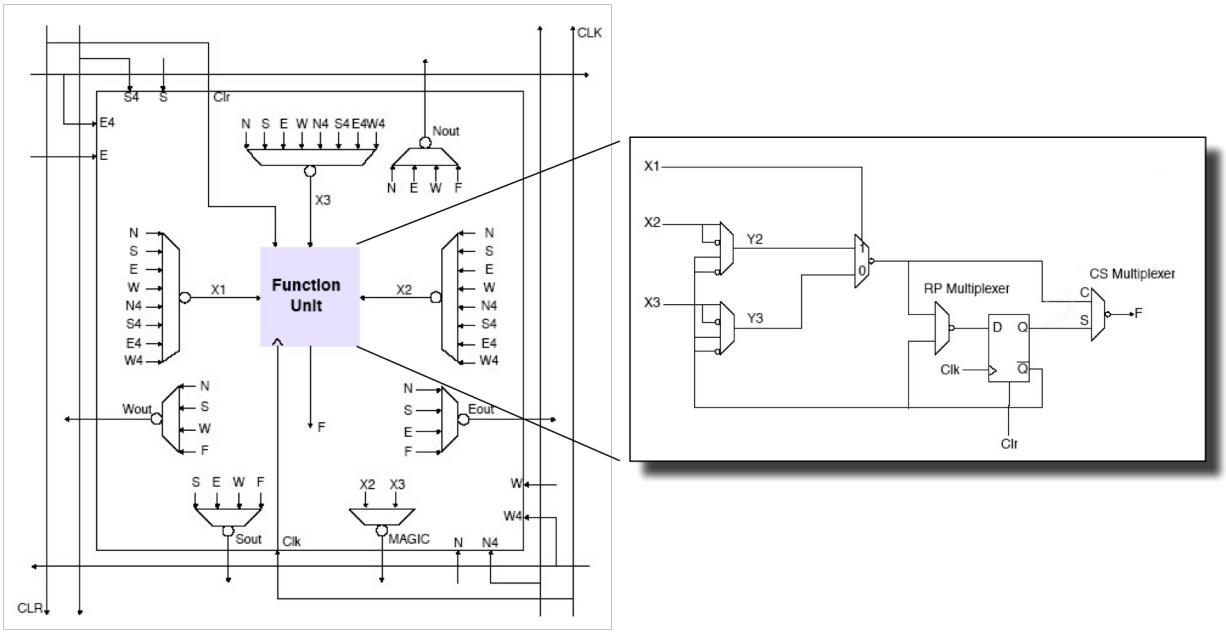


**Fig. 1** XC6200 basic cell and function unit [10]. Three 8:1 multiplexers provide 3 inputs for the function unit. The function unit generates one logic result (combinational or sequential). The logic result and some direction signals are selected and one final result in each direction is delivered to the neighbor cells.

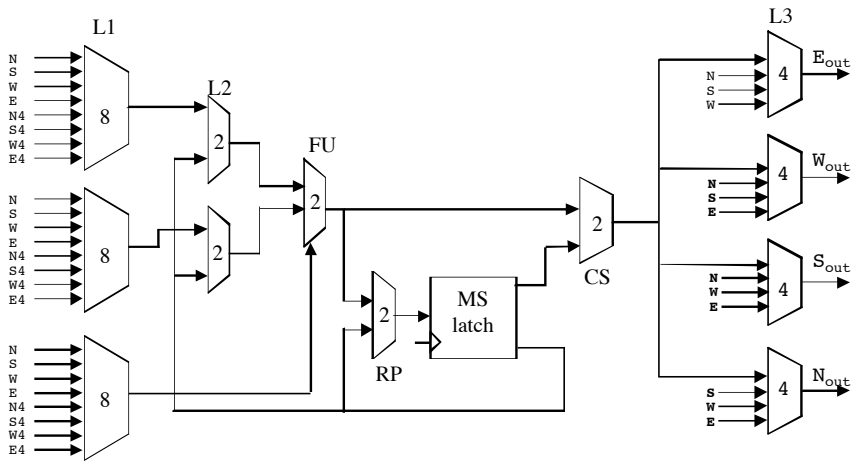


**Fig. 2** Simplified Xilinx XC6200 basic cell from [10]. FU here refers to the 2:1 multiplexer, which generates the combinational logic result.

### 2.2. Basic Cell Improvement

The XC6200 basic cell gives users flexibility through multiple selection steps. As shown in Fig. 2, input signals are selected by 5-stage multiplexing selection until final function results are delivered to neighbor cells. On the other hand, these multiple selection steps greatly slow down the basic cell's operation. A logic result is selected by the 2:1 CS multiplexer and the 4:1 output multiplexer before the logic result is delivered to neighbor cells.

An improved basic cell structure is shown in Fig. 3. The idea to improve the basic cell performance is to leave the selection procedures to each cell's input multiplexers. Each cell will provide function results without selecting them.

As shown in the new structure, each cell sends the combinational logic result, sequential logic result and redirection result directly to its neighbor cells. The only function left for the 4:1 output multiplexer in Fig. 1 is to

redirect a signal from one direction to another. Instead of sending just one final output, each basic cell now provides three outputs.

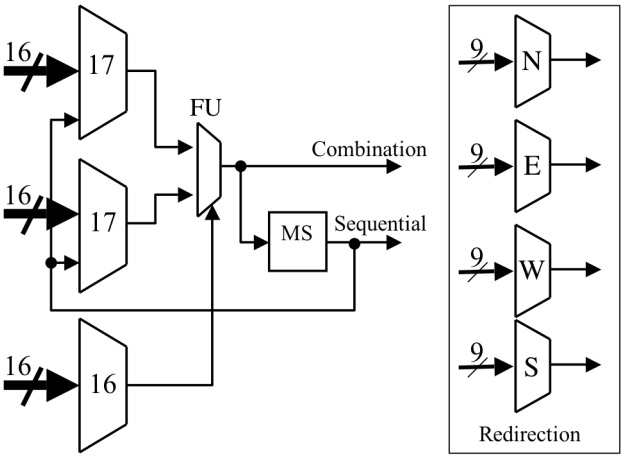


**Fig. 3** Improved basic cell structure. The FU here is a 2:1 multiplexer. The FU generates the combinational result and sends the result directly to four neighbor cells. The MS latch generates a sequential result and sends the result directly to four neighbor cells. Since logic results are sent directly to neighbor cells, three gate delays are saved**.**

On the input side of each basic cell, instead of receiving two inputs in each direction, a basic cell now receives four inputs (three inputs from neighbor cells and one FastLANE™). A 16:1 (the 17:1 multiplexer has one more input from the MS latch feedback) multiplexer is used at the front side. The 16:1 multiplexer uses a two-level 4:1 multiplexer. The chip area is approximately 20% larger than the original design with the routing and speed benefit.

### 2.3. Improvement Summary

From the previous description, one can see that the new basic cell (BCII) structure has three less gate delays for all logic function results. Those three gates are the 2:1 CS multiplexer, the 4:1 output multiplexer and one signal driver after the CS multiplexer. Instead of 5 gate delays, BCII only has 3 gate delays. A faster system clock can be used in the FPGA, thus providing higher performance.

In addition to providing better performance than the original XC6200 basic cell, the BCII structure also provides more routing capabilities than the original XC6200. The XC6200 basic cell only provided one signal to its neighbors each time. The BCII provides three signals. Suppose one cell needs both combination and sequential results from its east neighbor. With the old structure, the eastern basic cell can only provide the combination result through its output multiplexer. The sequential result has to be routed through other basic cells to arrive at its final destination. In the BCII structure, the eastern basic cell can provide both signals together. Basic cells are saved in this scenario.

In summary, the BCII structure provides less gate delays and more routing capabilities over the old basic cell structure. High-speed FPGAs can be built with this new basic cell.

## 3. HIGH-SPEED DEMUX CONFIGURED BY FPGA

### 3.1. 2×4 FPGA configuration

To prove the speed and functionality of this high speed FPGA, a four channel DEMUX was implemented in an FPGA by the state-of-the-art IBM SiGe 7HP technology ($f_T$=120 GHz). The block diagram is shown in Fig. 4.

The DEMUX is configured by a 2×4 gate array. A 4-bit LFSR is used to generate pseudo random serial data. The Sample-Hold (S-H) circuits, shown in Fig. 4, are implemented by a 1×4 gate array. The S-H circuit is configured as shown in Fig. 5. When the SEL signal is asserted, serial data will be latched in the MS-Latch. Otherwise, the latched data will be held in the latch. Table 1 shows the truth-table values of the four SEL1 – SEL4 signals.

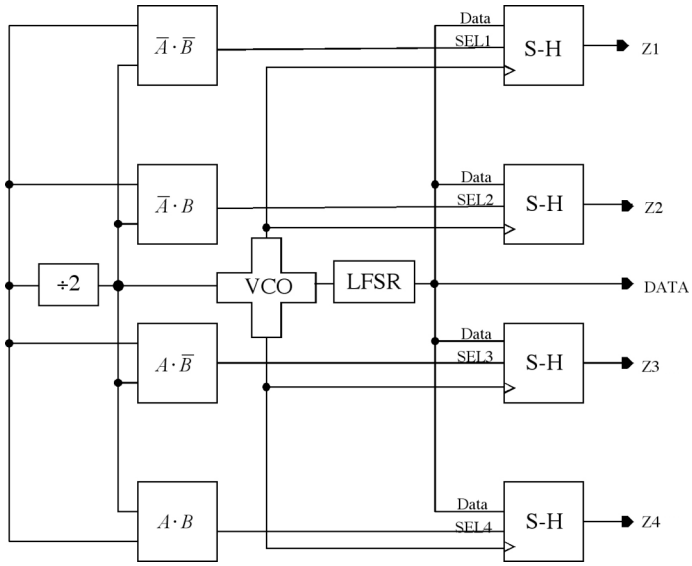


**Fig. 4** A four channel DEMUX implemented in a 2×4 gate array

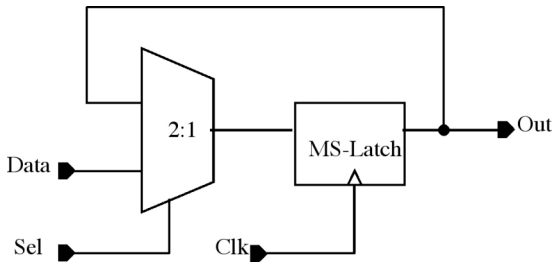


**Fig. 5** Sample-Hold (S-H) circuit implemented by the basic cell**.**

**Table 1.** The SELn signals' combinational logic

| A | B | SEL1 | SEL2 | SEL3 | SEL4 |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

### 3.2. Chip Measurement

The fabricated chip is shown in Fig. 6. The chip area is 1.09 mm × 1.68 mm. Areas "A" and "B" are the 2×4 gate arrays. Area "C" is assistance circuitry, including the VCO, LFSR, divide-by-2 circuit and signal drivers.

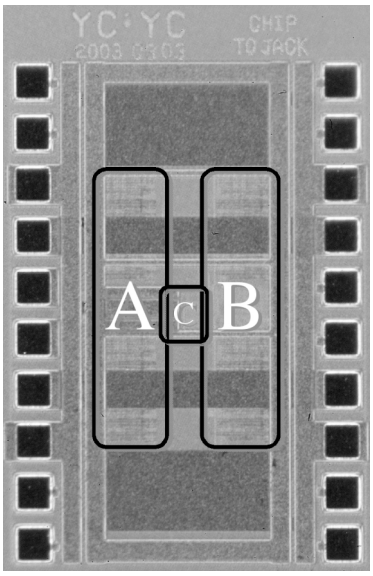


**Fig. 6** Photograph of the 1.09 mm × 1.68 mm chip - 4:1 decoder (A), four Sample-Hold circuits (B), and VCO and LFSR (C).

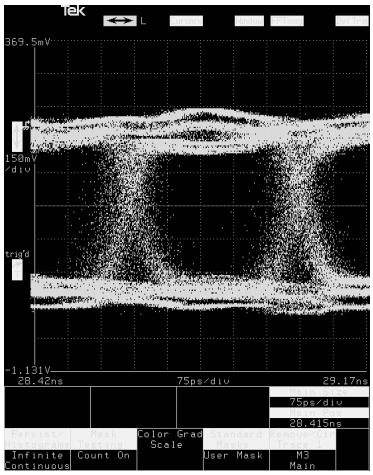


**Fig. 7** Chip measurements waveform. The channel "A" output eye diagram. Channel "A" bit rate is 2.7 Gb/s. The full four channel DEMUX bit rate is 10.8 Gb/s.

The chip measurement is taken at room temperature. Testing power supply is -3.4 V. The total chip current is 840 mA.

Approximately 45% of the power is consumed by the input and output pad drivers. A 2.7 Gb/s channel "A" eye diagram is shown in Fig. 7. The corresponding DEMUX bit rate is 10.8 Gb/s for this 1:4 DEMUX.

Compared to the BiCMOS version of XC6200 FPGA [7], this FPGA has approximately a 50% performance increase. Each cell has more routing capabilities than the original design.

## 4. CONCLUSION

This paper introduces a high-speed FPGA and configures it as a high-speed DEMUX as a test application. The FPGA has new features that have better performance and increased routing capability. An 11 Gb/s 1:4 DEMUX is demonstrated in the IBM 7HP technology. An IBM 8HP SiGe FPGA chip is being designed with still further improvements. The new chip is predicted to operate in applications at 20 Gb/s. Other implementations that utilize bigger gate arrays have been submitted for fabrication.

## 5. REFERENCES

[1] A. DeHon, "DPGA-Coupled Microprocessors: Commodity ICs for the Early 21st Century," *IEEE Workshop on FPGAs for Custom Computing Machines*, pp. 31-39, April 1994.

[2] M. Butts, "Futures Directions of Dynamically Reprogrammable System," *IEEE Custom Integrated Circuits Conference*, pp. 487-494, May 1995.

[3] T. Moeller, D. Martinez, "Field Programmable Gate Array Based Radar Front-End Digital Signal Processing,"

*Field-Programmable Custom Computing Machines*, 1999, FCCM '99, Proceedings Seventh Annual IEEE Symposium on, pp. 178-187, April 1999.

[4] S. Knapp, "Using Programmable Logic to Accelerate DSP Functions," Xilinx Application Note, Xilinx Inc. March 1998.

[5] J. Arnold, D. Buell, D. Hoang, D. Pryor, N. Shirazi, M. Thistle, "The Splash 2 processor and applications," *VLSI in Computers and Processors*, 1993, ICCD '93. Proceedings, 1993 IEEE International Conference on, pp. 482-285, Oct. 1993

[6] B. Goda, J. McDonald, S. Carlough, T. Krawczyk, R. Kraft, "SiGe HBT BiCMOS FPGAs for Fast Reconfigurable Computing," *IEE Proceedings of Computers and Digital Techniques*, vol. 147, no. 3, pp. 189-194, May 2000

[7] B. Goda, "SiGe HBT BiCMOS Field Programmable Gate Arrays for Fast Reconfigurable Computing," Ph.D. Thesis, Rensselaer Polytechnic Institute, 2001

[8] S. Singh, J. Rose, P. Chow, D. Lewis, "The Effect of Logic Block Architecture on FPGA Performance," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 3, pp. 281-287, March 1992.

[9] A. Gamal, J. Greene, J. Reyneri, E. Rogoyski, K. El-ayat, A. Mohsen, "An Architecture for Electrically Configurable Gate Arrays," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 2, pp. 394-398, April 1989.

[10] Xilinx, "XC6200 Field Programmable Gate Arrays," Xilinx Inc. 1999.

[11] P. H. W. Leong, C. W. Sham, W. C. Wong, H. Y. Wong, W. S. Yuen and M. P. Leong, "A bitstream Reconfigurable FPGA Implementation of the WSAT Algorithm," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 1, pp. 197-201, Feb 2001.