Abstract:

This report concerns FPGAs (**F**ield **P**rogrammable **G**ate **A**rray**s**). The basic FPGA blocks, I/O, CLBs (**C**ombinational **L**ogic **B**lock**s**), and routing architecture, are discussed to impart a basic understanding of FPGA operation. The static RAM implementation method for the programming elements of FPGAs is briefly discussed. An in depth investigation of one common static RAM chip, the Xilinx XC5200, providing physical examples of I/O, CLB, and routing configurations used in industry, is included in the body of this report. Strengths and weaknesses of static RAM FPGAs are discussed in the conclusion.

Field Programmable Gate Arrays:

Introduction to FPGAs:

Xilinx introduced Field programmable gate arrays, or FPGAs, in 1985. Figure 1 is a conceptual model of an FPGA.[1] FPGA are constructed of three basic elements: logic blocks, I/O cells, and interconnection resources. A useful analogy for an FPGA is the layout of a city. The logic blocks correspond to city blocks that are occupied by different businesses receiving products from various suppliers within the city, just as the logic blocks receive data from other logic blocks within the FPGA, and processing those products for consumption by other firms or end users, just as logic block outputs are sent to other blocks and ultimately to the device utilizing the FPGA. FPGAs and our mythical city both utilize interconnections between blocks, wire segments for FPGAs and streets and telephone connections for the city, that can be flexibly designed to meet changing needs with routers in both cases and stoplights in one. The final elements in the model are the mechanisms for interaction with the outside world; I/O cells to the FPGA as airports, freeways, and long distance telephone lines are to the city. The rest of this report will explore in greater detail implementations of this basic three-element model.

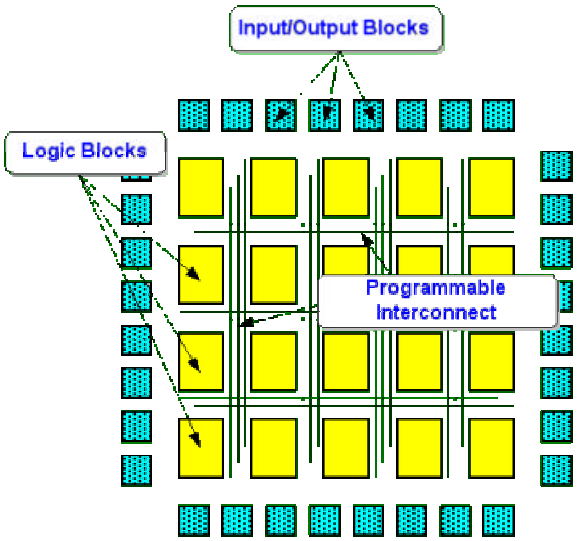


Figure 1: Basic FPGA Configuration

Configurable Logic Blocks:

The heart of the FPGA lies in the CLBs. CLBs appear in rows and columns within all FPGAs and implement the logic functions desired by the programmer. Most CLBs accomplish this with a lookup table[2]. Lookup tables (LUTs) are digital memory arrays that contain truth tables for any logic function that can be implemented by the given number of logic inputs for a CLB. The output of the CLB is then the logical result of the function recorded in the lookup table. In order to program the CLBs, truth tables be loaded into the LUTs of each CLB. Refer to page 3 for an example of the CLB architecture for a Xilinx XC5200 chip.

I/O Blocks:

I/O blocks provide for interaction with the outside world. An I/O pin can be used for input or output.4 I/O blocks can contain logic functionality, although high logic utilization decreases pin placement flexibility, as I/O blocks utilized in logic cannot be reassigned mid-design.[5]

Interconnection (Routing) Architecture:

The routing architecture usually covers 60-90% of FPGA chip area[2] and fittingly will require the longest description of the three basic FPGA elements. The routing architecture of FPGAs is constructed of wires segmented into various lengths intersecting each other at routing switches[3]. The most popular programmable switch element (PSE) technology, static RAM, for implementing these routing switches is briefly discussed in the next section. Two types of routing architecture are common[2]: row based routing, where only horizontal channels are used to connect CLBs, and symmetrical routing, where vertical and horizontal channels are utilized, as in figure 1. Direct connection wires link neighboring CLBs across routing channels. Connections to distant blocks are implemented through programmable switch matrices[4] (PSMs), which contain a set of PSEs that switch perpendicular wires. The wires routed through PSM are either single lines, which must pass through one PSE for each CLB bypassed, or double lines, which pass two CLBs for every switch. Long lines skip switching all together. The implementation of complex routing techniques is described later for the Xilinx XC5200.

Static RAM:

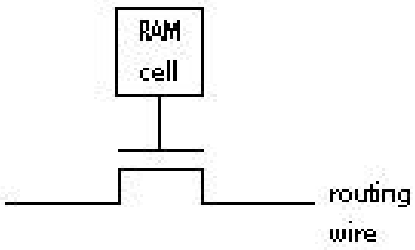


Figure 2: Static RAM Programmable Switch

The most common programmable element used for FPGA implementation is static RAM. The most basic static RAM switch is displayed in figure 2[2]. Static RAM FPGAs use permanent memory, usually PROM, to store the logic configuration of the chip. Upon power up, each RAM cell gets a value based upon the PROM configuration. When the cell is high, the transistor is conducting and current flows, when low, the transistor is cutoff and no current flows.

Xilinx XC5200, Static RAM FPGA:

Configurable Logic Blocks:

Figures 3 and 4[5] illustrate the operation of a CLB for the Xilinx XC5200 FPGA chip. The diagram on the left is of one of the four identical logic cells that constitute each CLB. The segment labeled F contains a lookup table for four inputs (F4-F1). The trapezoidal objects are 2:1 mutiplexers. The chip enable (CE), clock (CK) and clear (CLR) signals travel to this cell and all others in the architecture via global long lines. Each cell can be cleared individually or all can be cleared at once. Each logic cell can implement either a D flip-flop or a latch. When the clock transitions high, the D flip-flop (FD) passes the output of the programmed logic operation to the output (Q). From DI to DO, a feed-through path that does not change the logic of the input can be implemented. This is used in routing applications discussed later.

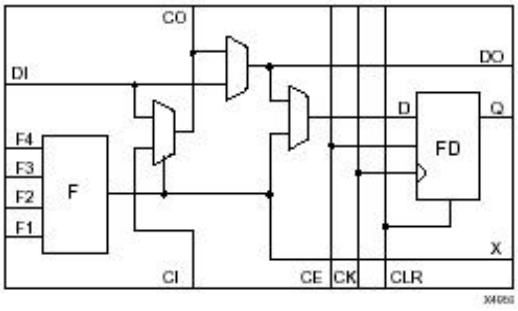


Figure 3: XC5200 Logic Cell (Four Cells Per CLB)

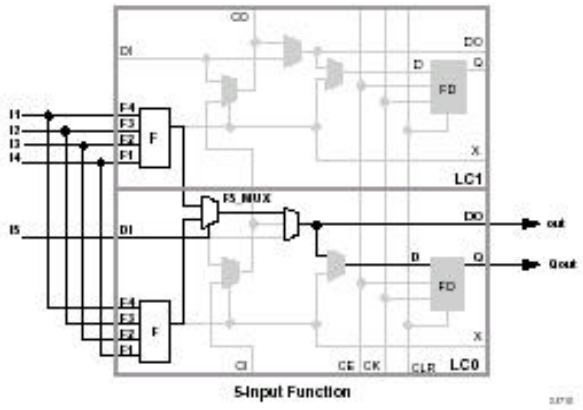


Figure 4: Parallel LUTs Implement a 5-Input Function

The diagram on the right is that of a CLB when programmed to implement a 5-input logic function[5]. In this case, two lookup tables (F) are used for input, each fed with the same four logic lines. The fifth input is used to toggle the 2:1 mux between the lookup tables, adding a fifth bit to the logic function. There are four lookup tables in each CLB, so four independent four-input logic functions or two independent five input logic functions can be implemented in each block.

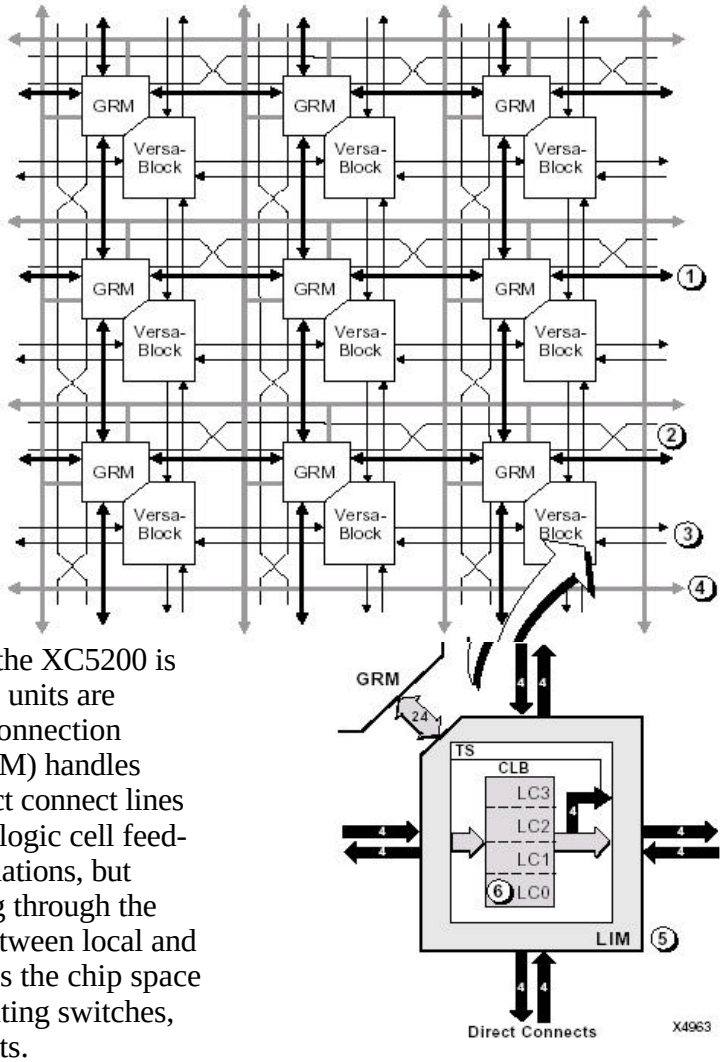
I/O Blocks:

The I/O blocks of XC5200 are completely decoupled from the internal logic of the CLBs.[5] The I/O blocks are attached to the internal logic through a ring of inter-connect cells which form a ring around the chip. The extra routing layer provides connection to nearby CLBs as well as far away CLBs through long lines. The XC5200 can be connected with TTL or CMOS logic.

Interconnection (Routing) Architecture:

Figure 5 displays the routing architecture of the Xilinx XC5200 FPGA[5]. This chip has six levels of routing hierarchy: single length lines (1), double length lines (2), direct connects (3), long lines/global lines (4), local interconnection matrices (5), and logic cell feed-through paths (6). The global routing matrix (GRM) contains the switch matrix architecture discussed earlier in this report. The GRM routs logic signals over the single, double and long lines, then communicates to the CLB via a 24-line interface to

the LIM. These matrices connect far-away sections of the chip as well as link all CLBs to a global command structure.



The remaining routing architecture for the XC5200 is contained within the Versa-Block units. These units are comprised of the CLBs, as well as local interconnection matrices. The local interconnection matrix (LIM) handles connections to neighboring CLBs through direct connect lines that bypass the GRMs. The LIM also handles logic cell feed-through paths, which do not perform any calculations, but merely re-power a signal that has faded passing through the chip. This splitting of the routing resources between local and global areas simplifies router design, decreases the chip space necessary for routing, and decreases use of routing switches, which add resistance and capacitance to circuits.

Figure 5: Routing Architecture for XC5000

Conclusions:

The flexibility of FPGAs gives them a distinct advantage over other programmable logic devices on the market. The advantage is most apparent in application where time to market concerns are paramount[3]. Because FPGAs are reprogrammable and can implement any sort of logic circuit, designs can be modified after initial implementation. With one time programmable technologies such as AISCs, logic is set at the factory and no changes can be made after manufacture. The drawbacks to FPGA use involve speed and space. Switching gate resistances and capacitances make for slow logic and poor logic density.

---

[1] www.optimagic.com

[2] Brown 1992, *Field Programmable Gate Arrays*, Kluwer Academic Publishers

[3] Bostock 1996, *FPGAs and Programmable LSI: a Designers Handbook*, Oxford: Butterworth-Hieneman

[4] Wakerly 2000, *Digital Design: Principals and Practice*

[5] www.xilinx.com