FPGA

The full form of **FPGA** is "**Field Programmable Gate Array**". It contains ten thousand to more than a million logic gates with programmable interconnection. Programmable interconnections are available for users or designers to perform given functions easily. A typical model FPGA chip is shown in the given figure. There are I/O blocks, which are designed and numbered according to function. For each module of logic level composition, there are **CLB's (Configurable Logic Blocks)**.

CLB performs the logic operation given to the module. The inter connection between CLB and I/O blocks are made with the help of horizontal routing channels, vertical routing channels and PSM (Programmable Multiplexers).

The number of CLB it contains only decides the complexity of FPGA. The functionality of CLB's and PSM are designed by VHDL or any other hardware descriptive language. After programming, CLB and PSM are placed on chip and connected with each other with routing channels.

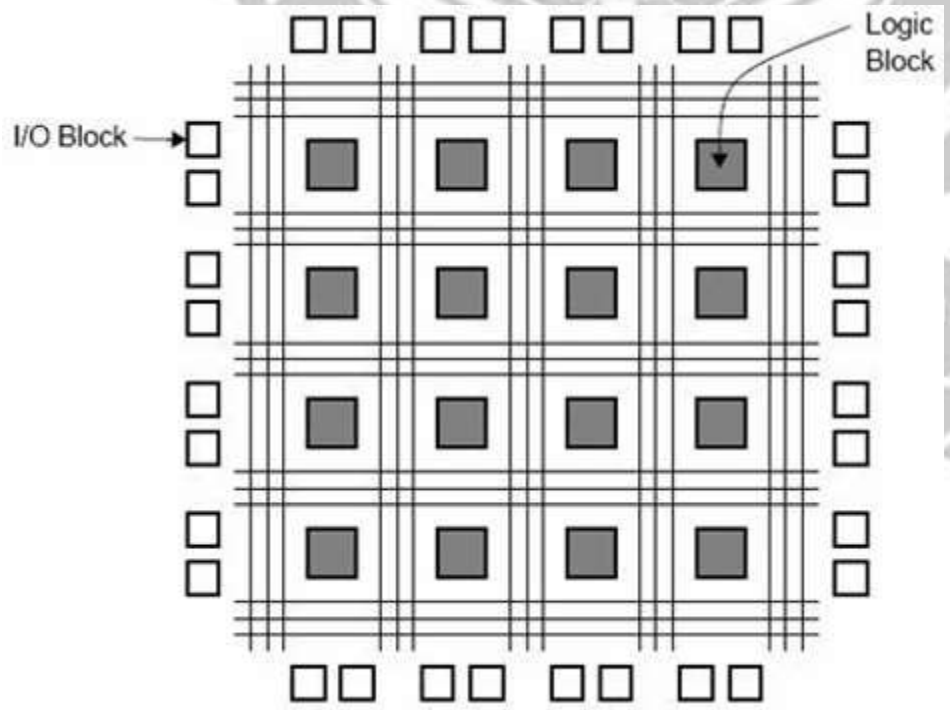

Fig 5.1.1: FPGA Block

[Source: Neil H.E. Weste, David Money Harris ―CMOS VLSI Design: A Circuits and

Systems Perspective]

Advantages

- It requires very small time; starting from design process to functional chip.
- No physical manufacturing steps are involved in it.
- The only disadvantage is, it is costly than other styles.

Gate Array Design

The **gate array (GA)** ranks second after the FPGA, in terms of fast prototyping capability. While user programming is important to the design implementation of the FPGA chip, metal mask design and processing is used for GA. Gate array implementation requires a two-step manufacturing process.

The first phase results in an array of uncommitted transistors on each GA chip. These uncommitted chips can be stored for later customization, which is completed by defining the metal interconnects between the transistors of the array. The patterning of metallic interconnects is done at the end of the chip fabrication process, so that the turn-around time can still be short, a few days to a few weeks. The figure given below shows the basic processing steps for gate array implementation.

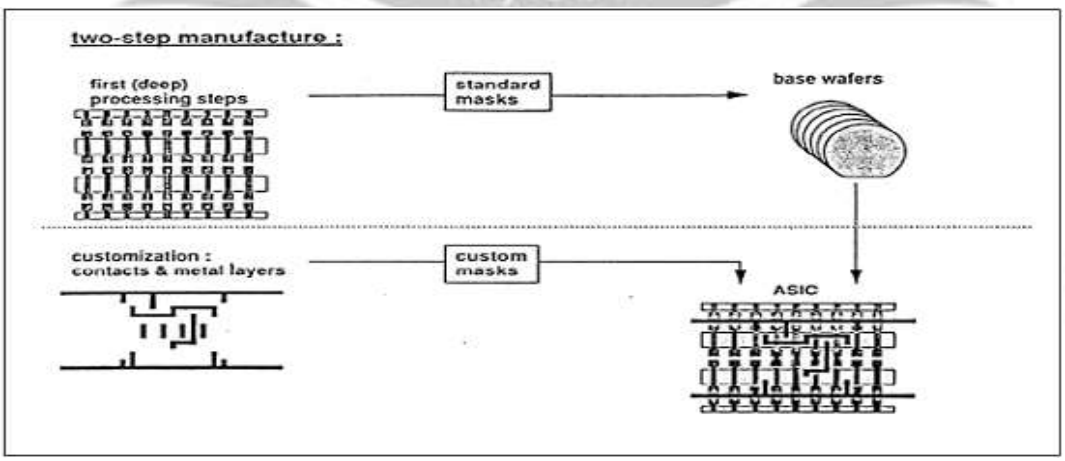

Fig 5.1.2: Array Implementation

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

Typical gate array platforms use dedicated areas called channels, for inter-cell routing between rows or columns of MOS transistors. They simplify the interconnections. Interconnection patterns that perform basic logic gates are stored in a library, which can then be used to customize rows of uncommitted transistors according to the netlist.

In most of the modern GAs, multiple metal layers are used for channel routing. With the use of multiple interconnected layers, the routing can be achieved over the active cell areas; so that the routing channels can be removed as in Sea-of-Gates (SOG) chips. Here, the entire chip surface is covered with uncommitted nMOS and pMOS transistors. The neighboring transistors can be customized using a metal mask to form basic logic gates.

For inter cell routing, some of the uncommitted transistors must be sacrificed. This design style results in more flexibility for interconnections and usually in a higher density. GA chip utilization factor is measured by the used chip area divided by the total chip area. It is higher than that of the FPGA and so is the chip speed.

Standard Cell Based Design

A standard cell based design requires development of a full custom mask set. The standard cell is also known as the polycell. In this approach, all of the commonly used logic cells are developed, characterized and stored in a standard cell library.

A library may contain a few hundred cells including inverters, NAND gates, NOR gates, complex AOI, OAI gates, D-latches and Flip-flops. Each gate type can be implemented in several versions to provide adequate driving capability for different fan-outs. The inverter gate can have standard size, double

size, and quadruple size so that the chip designer can select the proper size to obtain high circuit speed and layout density.

Each cell is characterized according to several different characterization categories, such as,

- Delay time versus load capacitance
- Circuit simulation model
- Timing simulation model
- Fault simulation model
- Cell data for place-and-route
- Mask data

For automated placement of the cells and routing, each cell layout is designed with a fixed height, so that a number of cells can be bounded side-by-side to form rows. The power and ground rails run parallel to the upper and lower boundaries of the cell. So that, neighboring cells share a common power bus and a common ground bus. The figure shown below is a floorplan for standard-cell based design.

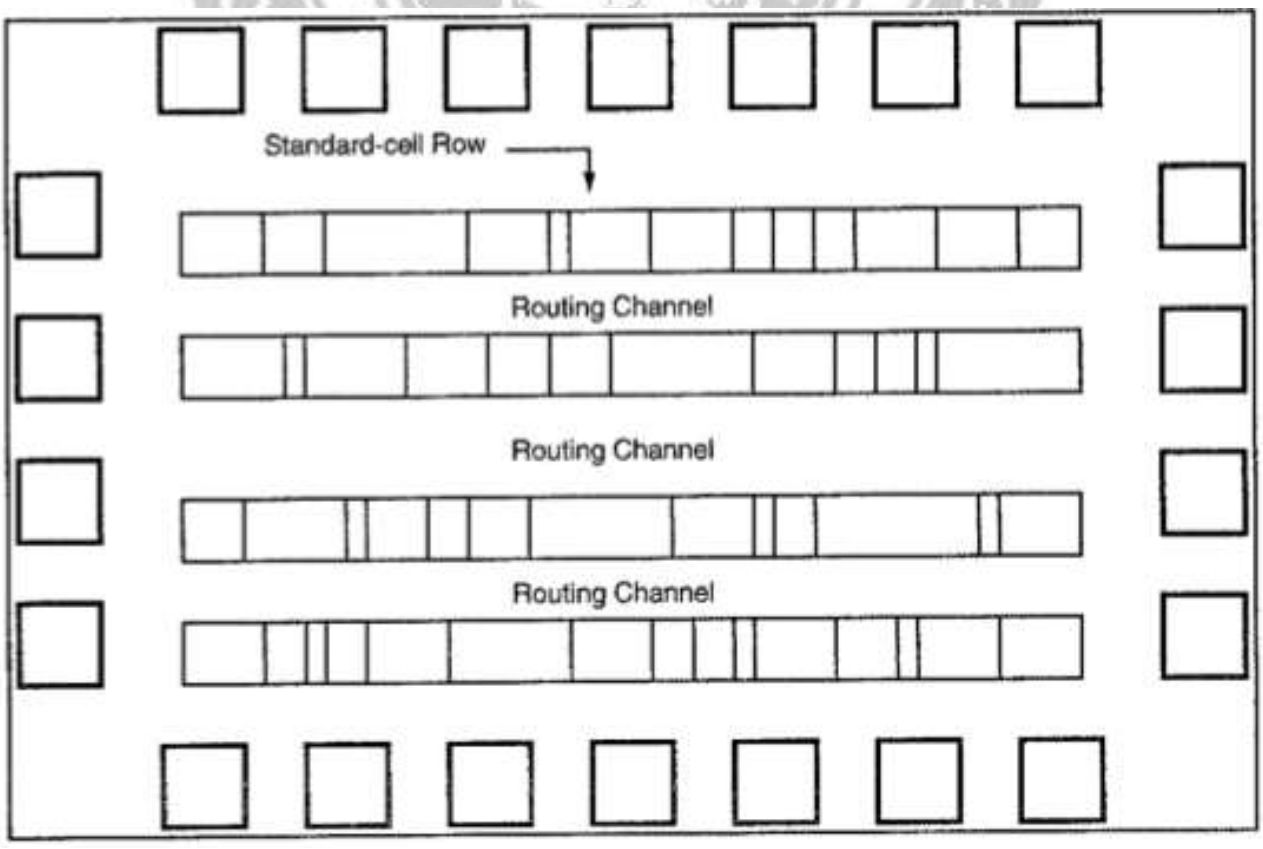

Fig 5.1.3: Floorplan For Standard-Cell Based Design

[Source: Neil H.E. Weste, David Money Harris ―CMOS VLSI Design: A Circuits and Systems Perspective]

Full Custom Design

In a full-custom design, the entire mask design is made new, without the use of any library. The development cost of this design style is rising. Thus, the concept of design reuse is becoming famous to reduce design cycle time and development cost.

The hardest full custom design can be the design of a memory cell, be it static or dynamic. For logic chip design, a good negotiation can be obtained using a combination of different design styles on the same chip, i.e. standard cells, data-path cells, and **programmable logic arrays (PLAs)**.

Practically, the designer does the full custom layout, i.e. the geometry, orientation, and placement of every transistor. The design productivity is usually very low; typically a few tens of transistors per day, per designer. In digital CMOS VLSI, full-custom design is hardly used due to the high labor cost. These design styles include the design of high-volume products such as memory chips, high-performance microprocessors and FPGA.

FPGA stands for Field Programmable Gate Array and, it is a one type of semiconductor logic chip which can be programmed to become almost any kind of system or digital circuit, similar to PLDs. PLDS are limited to hundreds of gates, but FPGAs supports thousands of gates. The configuration of the FPGA architecture is generally specified using a language, i.e., HDL (Hardware Description language) which is similar to the one used for an ASIC (Application Specific Integrated Circuit).

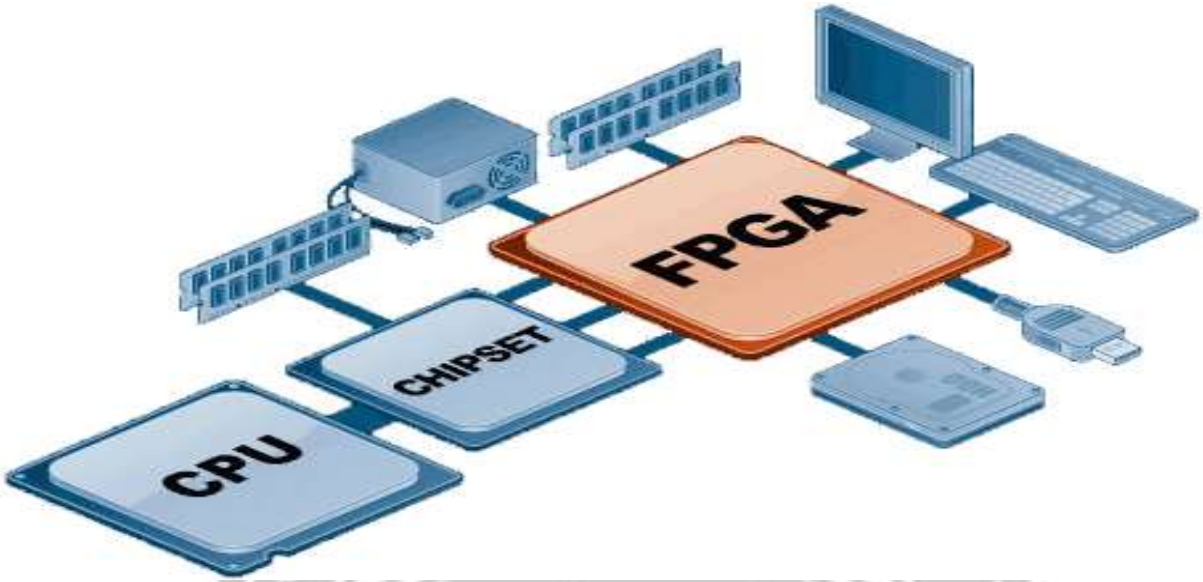

Fig 5.1.4: FPGA

[Source: Neil H.E. Weste, David Money Harris ―CMOS VLSI Design: A Circuits and Systems Perspective]

FPGAs can provide a number of advantages over a fixed function ASIC technology such as standard cells. Normally, ASICs takes months to manufacture and the cost of them will be thousands of dollars to obtain the device. But FPGAs are fabricated in less than a second, the cost will be from a few dollars to a thousand dollars. The flexible nature of the FPGA comes at a significant costing area, power consumption and delay. When compared to a standard cell ASIC, an FPGA requires 20 to 35 times more area, and the speed's performance will be 3 to 4 times slower than the ASIC. This article describes about the FPGA basics and FPGA architecture module that includes I/O pad, logic blocks and switch matrix. FPGAs are some of the new trending areas of VLSI.

FPGA Architecture

The general FPGA architecture consists of three types of modules. They are I/O blocks or Pads, Switch Matrix/ Interconnection Wires and Configurable logic blocks (CLB). The basic FPGA architecture has two dimensional arrays of

logic blocks with a means for a user to arrange the interconnection between the logic blocks. The functions of an FPGA architecture module are discussed below:

- CLB (Configurable Logic Block) includes digital logic, inputs, outputs. It implements the user logic.
- Interconnects provide direction between the logic blocks to implement the user logic.
- Depending on the logic, switch matrix provides switching between interconnects.
- I/O Pads used for the outside world to communicate with different applications.

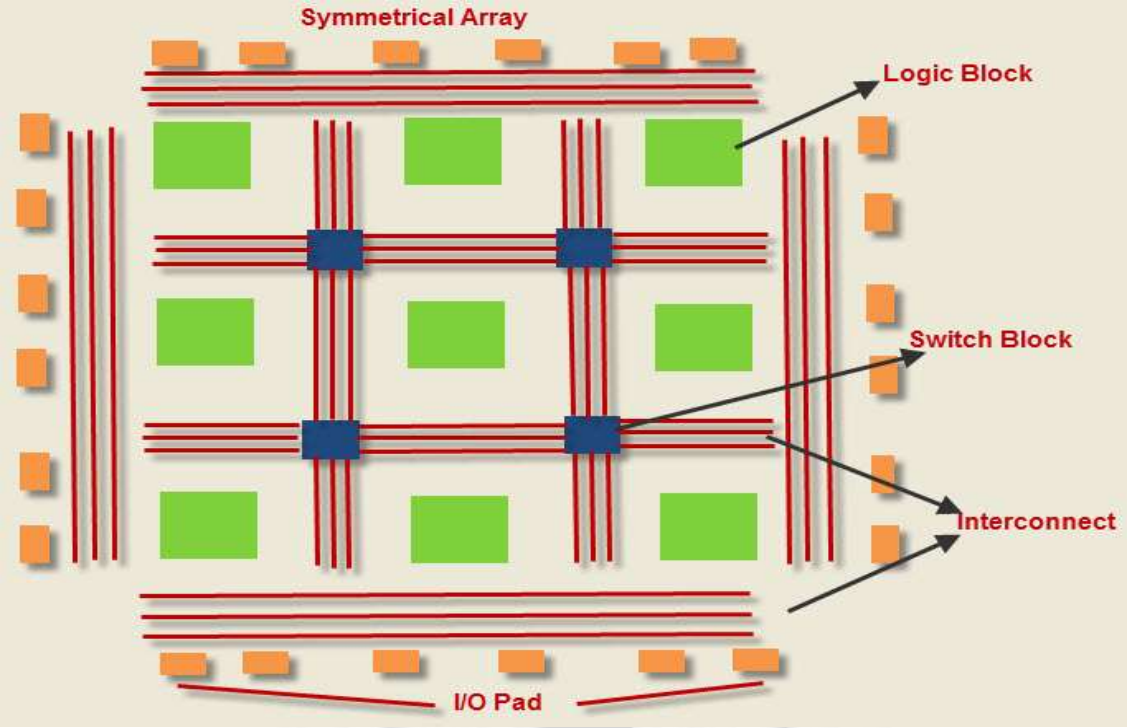

Fig 5.1.5: FPGA Block

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

FPGA Architecture

Logic Block contains MUX (Multiplexer), D flip flop and LUT. LUT implements the combinational logical functions; the MUX is used for selection

logic, and D flip flop stores the output of the LUT. The basic building block of the FPGA is the Look Up Table based function generator. The number of inputs to the LUT vary from 3,4,6, and even 8 after experiments. Now, we have adaptive LUTs that provides two outputs per single LUT with the implementation of two function generators.

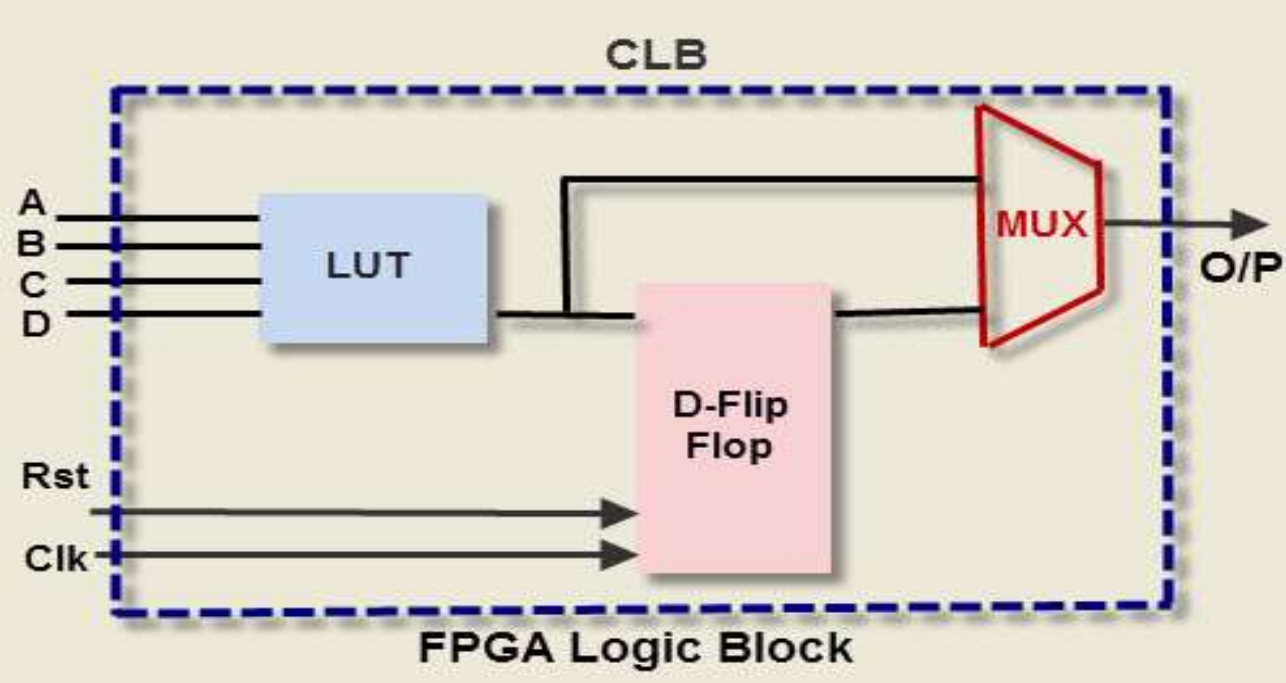

Fig 5.1.6: FPGA Logic Block

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

FPGA Logic Block

Xilinx Virtex-5 is the most popular FPGA, that contains a Look up Table (LUT) which is connected with MUX, and a flip flop as discussed above. Present FPGA consists of about hundreds or thousands of configurable logic blocks. For configuring the FPGA, Modelsim and Xilinx ISE softwares are used to generate a bitstream file and for development.

Types of FPGAs Based on Applications

Field Programmable Gate Arrays are classified into three types based on applications such as Low-end FPGAs, Mid-range FPGAs and high-end FPGAs.

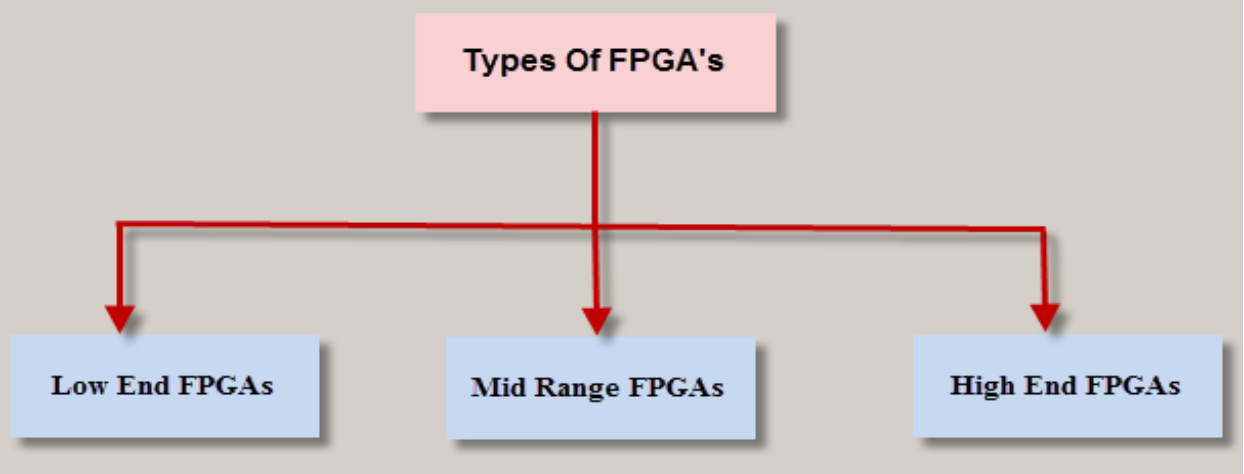

Fig 5.1.7: Types of FPGA'S

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

Types of FPGAs

Low End FPGAs

These types of FPGAs are designed for low power consumption, low logic density and low complexity per chip. Examples of low end FPGAs are Cyclone family from Altera, Spartan family from Xilinx, fusion family from Microsemi and the Mach XO/ICE40 from Lattice semiconductor.

Mid Range FPGAs

These types of FPGAs are the optimum solution between the low-end and high- end FPGAs and these are developed as a balance between the performance and the cost. Examples of Mid range FPGAs are Arria from Altera, Artix-7/Kintex-7 series from Xlinix, IGL002 from Microsemi and ECP3 and ECP5 series from Lattice semiconductor.

High End FPGAs

These types of FPGAs are developed for logic density and high performance. Examples of High end FPGAs are a Stratix family from Altera, Virtex family from Xilinx, Speedster 22i family from Achronix, and ProASIC3 family from Microsemi.

Applications of FPGA:

FPGAs have gained rapid growth over the past decade because they are useful for a wide range of applications. Specific application of an FPGA includes digital signal processing, bioinformatics, device controllers, software-defined radio, random logic, ASIC prototyping, medical imaging, computer hardware emulation, integrating multiple SPLDs, voice recognition, cryptography, filtering and communication encoding and many more.

Usually, FPGAs are kept for particular vertical applications where the production volume is small. For these low-volume applications, the top companies pay in hardware costs per unit. Today, the new performance dynamics and cost have extended the range of viable applications.

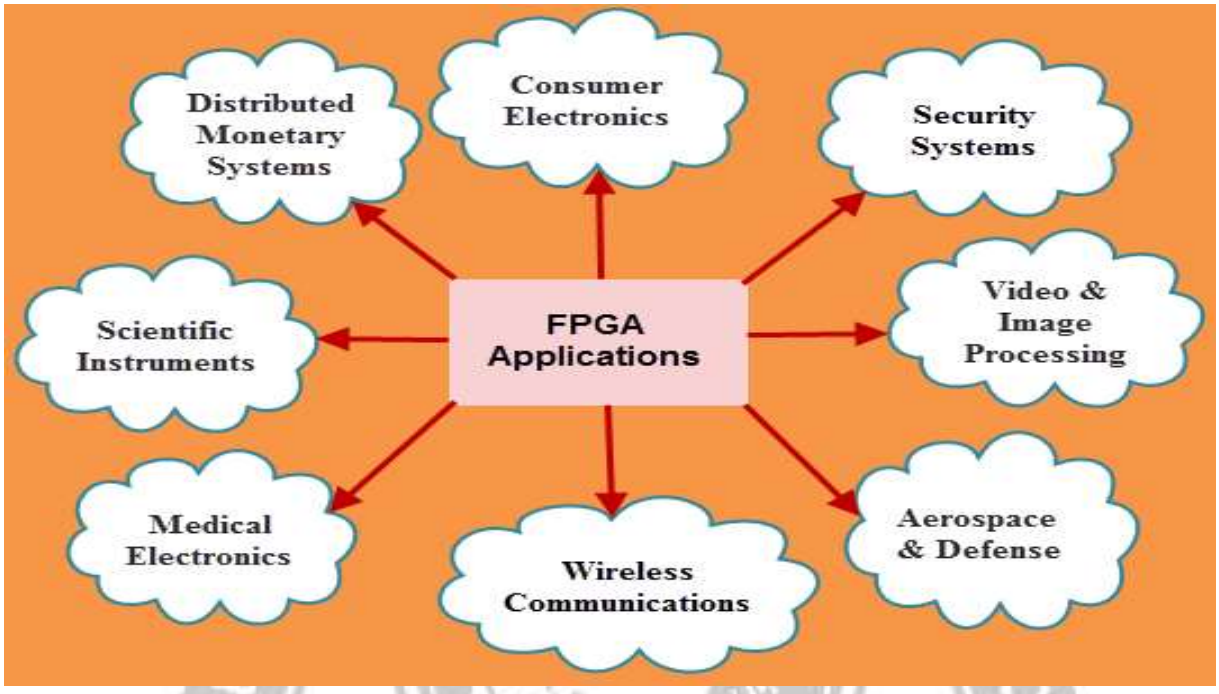

Fig 5.1.8: Applications of FPGA

[Source: Neil H.E. Weste, David Money Harris ―CMOS VLSI Design: A Circuits and Systems Perspective]

Applications of FPGA

Some More Common FPGA Applications are: Aerospace and Défense, Medical Electronics, ASIC Prototyping, Audio, Automotive, Broadcast, Consumer Electronics, Distributed Monetary Systems, Data Centre, High Performance Computing, Industrial, Medical, Scientific Instruments, Security systems, Video & Image Processing, Wired Communications, Wireless Communications.

Electronic industry has simulations and prototyping as their important segments since a long period. Electronic companies design the hardware dedicated to their products with their standards and protocols which makes it challenging for the end users to reconfigure the hardware as per their needs. This requirement for hardware led to the growth of a new segment of customer-configurable field programmable integrated circuits called FPGAs. In this article, we discuss FPGA Architecture and Applications.

FPGA Architecture

FPGAs are prefabricated silicon chips that can be programmed electrically to implement digital designs. The first static memory based FPGA called SRAM is used for configuring both logic and interconnection using a stream of configuration bits. Today's modern EPGA contains approximately 3,30,000 logic blocks and around 1,100 inputs and outputs.

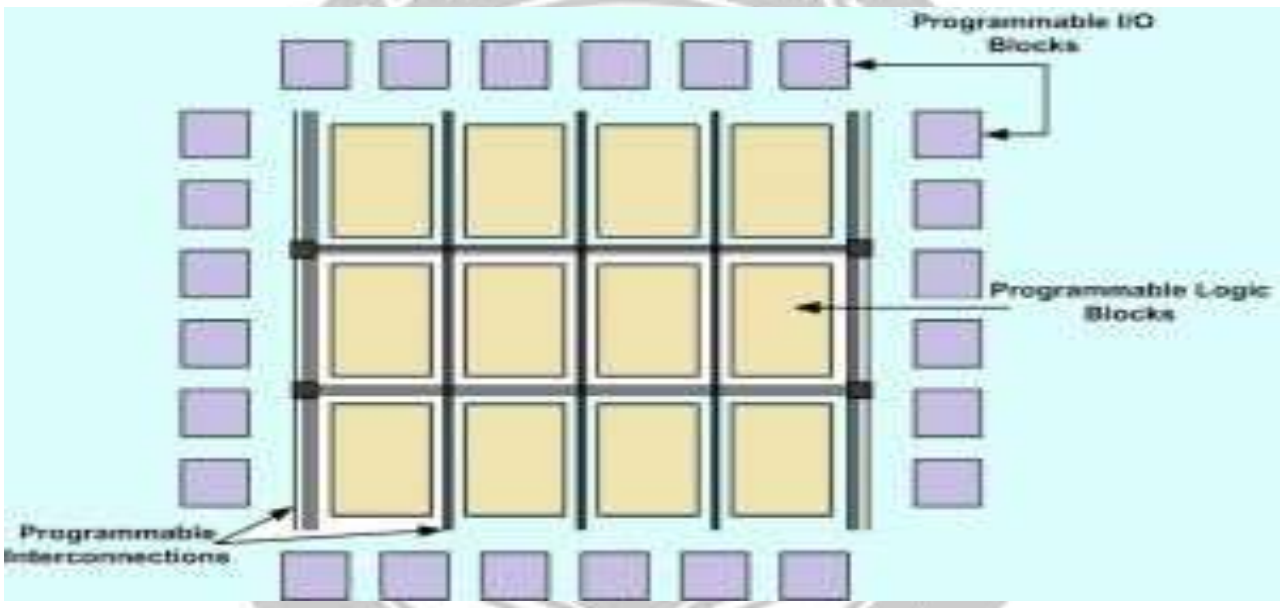

Fig 5.1.9: FPGA Architecture

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

FPGA Architecture

The FPGA Architecture consists of three major components

- Programmable Logic Blocks, which implement logic functions
- Programmable Routing (interconnects), which implements functions
- I/O blocks, which are used to make off-chip connections

Programmable Logic Blocks

The programmable logic block provides basic computation and storage elements used in digital systems. A basic logic element consists of programmable combinational logic, a flip-flop, and some fast carry logic to reduce area and delay cost.

Modern FPGAs contain a heterogeneous mixture of different blocks like dedicated memory blocks, multiplexers. Configuration memory is used throughout the logic blocks to control the specific function of each element.

Programmable Routing

The programmable routing establishes a connection between logic blocks and Input/Output blocks to complete a user-defined design unit.

It consists of multiplexers pass transistors and tri-state buffers. Pass transistors and multiplexers are used in a logic cluster to connect the logic elements.

Programmable I/O

The programmable I/O pads are used to interface the logic blocks and routing architecture to the external components. The I/O pad and the surrounding logic circuit form as an I/O cell.

These cells consume a large portion of the FPGA's area. And the design of I/O programmable blocks is complex, as there are great differences in the supply voltage and reference voltage.The selection of standards is important in I/O architecture design. Supporting a large number of standards can increase the silicon chip area required for I/O cells.With advancement, the basic FPGA Architecture has developed through the addition of more specialized programmable function blocks.The special functional blocks like ALUs, block RAM, multiplexers, DSP-48, and microprocessors have been added to the FPGA, due to the frequency of the need for such resources for applications.

The below snap shows an example of an FPGA Board.

Fig 5.1.10: FPGA Board

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

FPGA Architecture Design Flow

FPGA Architecture design comprises of design entry, design synthesis, design implementation, device programming and design verification.Design verification includes functional verification and timing verification that takes place at the time of design flow. The following flow shows the design process of the FPGA.

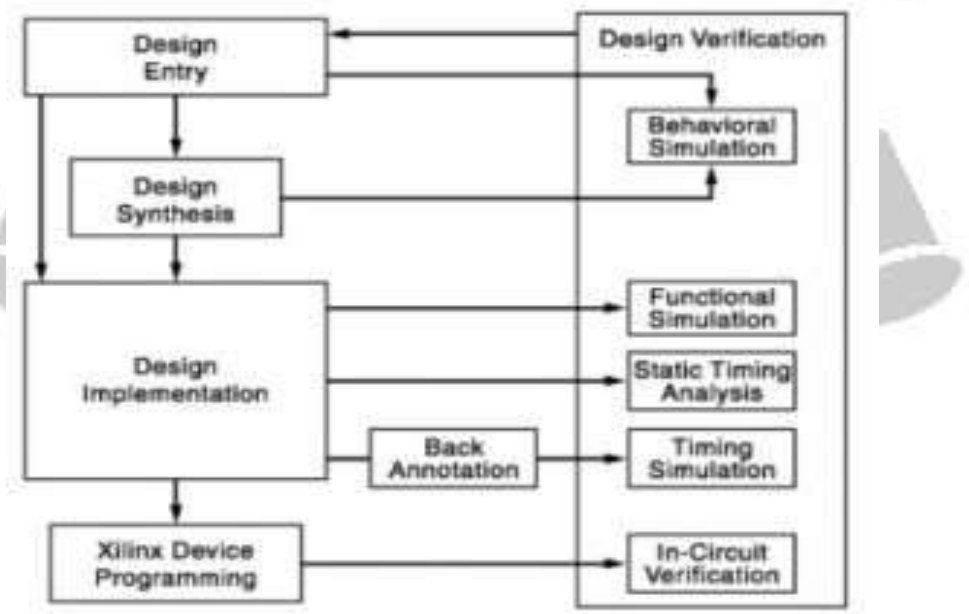

Fig 5.1.11: FPGA Architecture Design Flow

[Source: Neil H.E. Weste, David Money Harris ―CMOS VLSI Design: A Circuits and Systems Perspective]

Design Entry

The design entry is done in different techniques like schematic based, hardware description language (HDL) and a combination of both etc. If the designer wants to deal with hardware, then the schematic entry is a good choice. If the designer thinks the design in an algorithmic way, then the HDL is the better choice. The schematic based entry gives the designer a greater visibility and control over the hardware.

Design Synthesis

This process translates VHDL code into a device netlist format, i.e., a complete circuit with logical elements. The design synthesis process will check the code syntax and analyze the hierarchy of the design architecture.

This ensures the design optimized for the design architecture. The netlist is saved as Native Generic Circuit (NGC) file.

Design Implementation

The implementation process consists of

- Translate
- Map
- Place and Route

Translate

This process combines all the input netlists to the logic design file which is saved as NGD (Native Generic Database) file. Here the ports are assigned to the physical elements like pins, switches in the design. This is stored in a file called User Constraints File (UCF).

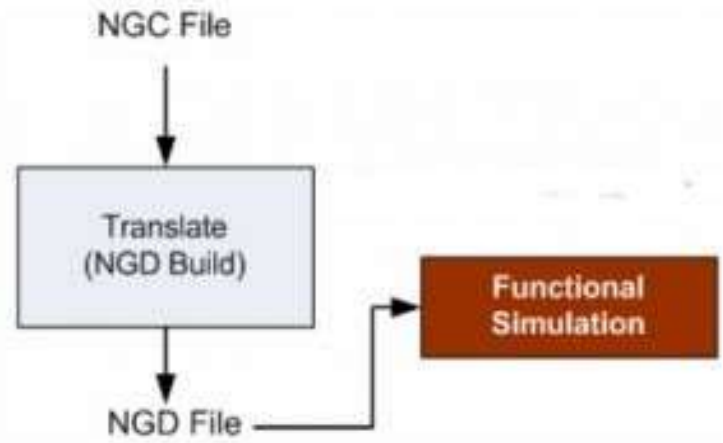

Fig 5.1.12: Translate

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

Map

Mapping divides the circuit into sub-blocks such that they can be fit into the FPGA logic blocks. Thus this process fits the logic defined by NGD into the combinational Logic Blocks, Input-Output Blocks and then generates an NCD file, which represents the design mapped to the components of FPGA.

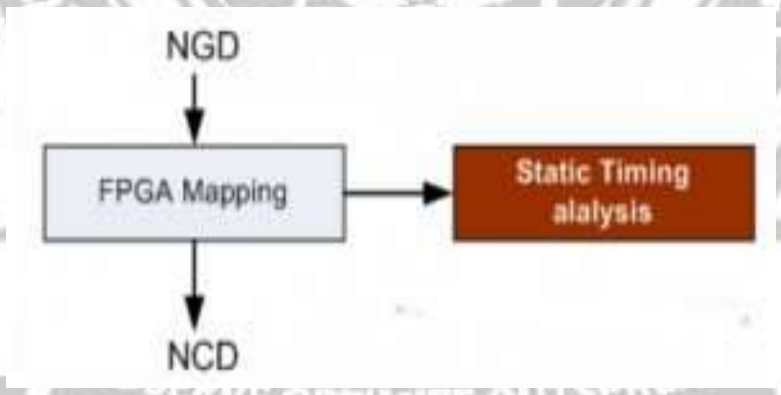

Fig 5.1.13: Map

[Source: Neil H.E. Weste, David Money Harris —CMOS VLSI Design: A Circuits and Systems Perspective]

Routing

The routing process places the sub-blocks from the mapping process into the logic block according to the constraints and then connects the logic blocks.

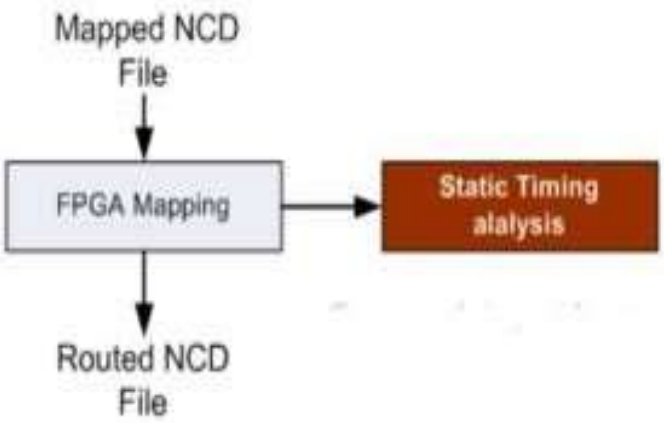

Fig 5.1.14: Routing

[Source: M.J. Smith, ―Application Specific Integrated Circuits]

Device Programming

The routed design must be loaded into the FPGA. This design must be converted into a format supported by the FPGA. The routed NCD file is given to the BITGEN program, which generates the BIT file. This BIT file is configured to the FPGA.

Design Verification

Verification can be done at various stages of the process.

1.Behavioral Simulation (RTL Simulation)

Behavioral simulation is the first of all the steps that occur in the hierarchy of the design. This is performed before cheap lace dresses the synthesis process to verify the RTL code.

In this process, the signals and variables are observed and further, the procedures and functions are traced and breakpoints are set.

2. Functional Simulation

Functional simulation is performed post-translation simulation. It gives the information about the logical operation of the circuit.

3. Static Timing Simulation

This is done post mapping. Post map timing report gives the signal path delays. After place and route, timing report takes the timing delay information. This provides a complete timing summary of the design.

Applications of FPGA

- FPGAs have gained a quick acceptance over the past decades. Here are the some of the applications of FPGAs in various technologies.
- Users can apply them to the wide range of applications like random logics, SPLDs, device controllers, communication encoding and filtering.
- The emulation of entire large hardware systems via the use of many interconnected FPGAs.
- They offer a powerful solution for meeting machine vision, industrial networking, motor control and video surveillance.
- FPGAs are used in custom computing machines.
- FPGAs provide a unique combination of highly parallel custom computation and low-cost computation.