Fundamentals of FPGA Architecture

Chapter · January 2017

CITATION
1

READS
9,398

2 authors, including:

Sahadev Roy
NIT, Arunachal Pradesh
84 PUBLICATIONS **283** CITATIONS

SEE PROFILE

Advanced Engineering

Fundamentals of FPGA Architecture

Gopal Krishna and Sahadev Roy

Fundamentals of FPGA Architecture

Gopal Krishna and Sahadev Roy

2.1 Introduction

Field programmable gate array (FPGA) first come in picture in early 1970s [1]. It is developed version of PLD. In PLD'S programmed after manufacturing in field, it has limited programmability. Field programmable gate is capable of implementing any digital circuit. This provides developer of creating wide array of logical structure minimum low cost.

Programmability is high in FPGA with minimum design time. Low power consumption, high speed input output, large parallelisms are the important features of it. In this chapter we will discuss about basic building blocks of FPGA's and core technologies of FPGA. To understand basic hardware we follow bottom to top approach. First discuss building blocks (block diagram) then we will discuss how they are combined and interconnected.

2.2 Requirement of FPGAs

By the early 1980's large scale integrated circuits (LSI) formed the back bone [2] of most of the logic circuits in major systems [3]. Microprocessors, bus/IO controllers, system timers etc were implemented using integrated circuit fabrication technology. Random glue logic [4] or interconnects were still required to help connect the large integrated circuits in order to:

1. Generate global control signals (for resets etc.)
2. Data signals from one subsystem to another sub system.

Systems typically consisted of few large scale integrated components and large number of SSI (small scale integrated circuit) and MSI (medium scale integrated circuit) components. Initial attempt to solve this problem led to development of Custom ICs which were to replace the large amount of interconnect. This reduced system complexity and manufacturing cost, and improved performance. However, custom ICs have their own disadvantages. They are relatively very expensive to develop, and delay introduced for product to market (time to market) because of increased design time [5]. There are two kinds of costs involved in development of custom ICs.

1. Cost of development and design

2. Cost of manufacture

Therefore the custom ICs method was only efficient for high volume products and which were not to market sensitive. An alternative to custom ICs FPGA were introduce, FPGA capable of implementing entire system on chip and it provide flexibility to reprogram for user. Development of FPGA removes the density relative to discrete SSI/MSI components (aproximately 10× of custom ICS).

One of the major advantage of FPGA over custom ICs is circuit implementation time is very less because physical layout, masking etc is absent in this process. Circuit implementation is done with help of the advanced CAD tools.

2.3 FPGA Evaluation

A digital system is mainly consisting of three components memory, microprocessor and logic element. Memory is used to store information like data of look-up table and content of database. Microprocessors are used for execution of software instruction written in word. Logical unit perform almost all operation like data display, signal processing, device to device interface. Data communication and time synchronisation is also provided by logical unit.

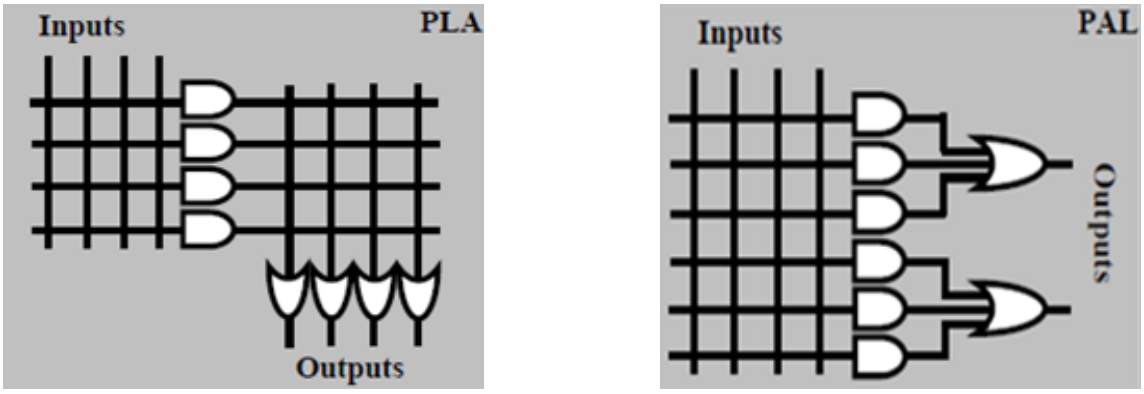
Programmable Read Only memory (PROM) was the first user programmable chip that could implement logical circuit. In programmable read only memory data lines are as output whereas address line as input. For its address input PROM requires a full decoder whereas logic functions require more than few product term. The PROM is used for implementing logical circuit.

FPGA is developed version of PLA. PLA is used for implementation of any Boolean function using sum of product form. Input buffer, programmable AND- matrix followed by programmable OR-matrix are basic part of PLA implementation. Basic function of input buffer is to provide original and its inverted value as PLA input [6]. Horizontal line is for the input which is implemented on AND matrix and product term line implemented virtually. So the size of AND matrix is twice than of product term of inputs.

First PLA was introduced by Philips in early 1970s, but some major drawbacks there in PLA poor speed performance and high manufacturing cost is very high. The two major disadvantages are due to the two level of configurable logic because programmable logic plane is not easy to manufacture

and it has some significant propagation delays. To overcome the drawbacks of PLA programmable array logic were introduce. PAL has single level programmability it contain wired AND followed by OR gate in same plane [7].

PAL generally contains flip-flops connected to the OR gate output so that sequential circuit can be realized. These are called as simple programmable logic devices (SPLDs) Fig. 1 shows simplified structure of PLA and PAL.



(a) (b)

Fig 1. *Illustration of simplified structure (a) PLA and (b) PAL.*

In latter on the development of SPLDs with development of technology it is possible to make device with higher density. In CPLDs chip density increased. In simple language we can say that CPLDs consist of multiple PLDs with some extra programmable interconnects on a single chip. The larger size of CPLD allows it to implement more complicated logic design on it.

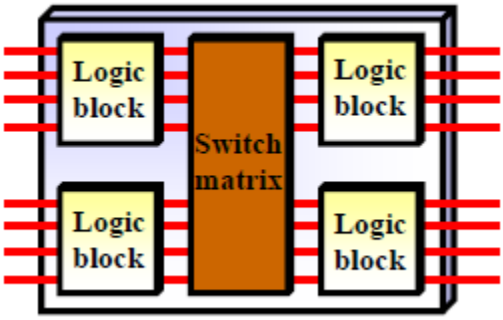


Fig 2. *Block diagram of Hypothetical CPLD*

Figure 2, is block diagram of hypothetical CPLD in this CPLD it is shown that it has four logic blocks it is simple PLDs. The number of PLD logic blocks may be more than four or less than four. These logic blocks may be more than four or less than four. These logic blocks also contain macro cell and interconnection similar as PLD.

In CPLD the switch matrix may be connected or not connected whereas in PLD it is fully connected. Some of the theoretical connection of logic blocks output and input is not supported within CPLD. Due to this it is difficult to achieve 100% utilization of macro cells .there is sufficient number of logic gates and flip-flops available in CPLD despite of that some hardware design is not allow to implement [8]. CPLD can hold larger design than PLD because of that it is widely used.

2.4 Hardware Design

Majority of electronic circuit is made up of the three basic components.

a. Combinational logic element

b. Memory element

c. Wiring to interconnect

2.4.1 Combinational Logic

A combinational logic is made up of small logic gates and interconnection them [5]. For example half adder is made up of a XOR gate and a AND gate this give sum output at s port which add input a AND b. For carry output port c is assigned. If a AND b both are 1 then carry will generate at and gate. Two half adder with one or gate can be combine to build a full adder. It has three input ports

(carry-in). Cascading of full adders can be done to make adder with wider word width. Another example of basic combinational circuit is multiplexer. 2 to 1 multiplexer has two inputs (in0, in1) and one select line (SEL). SEL line determines which input line appears at output. Wider multiplexer can be constructed to connect them. No clock is involved in combinational circuit. Every logic gate has its own propagation delay [9]. As the number of gates increases propagation delay also increases. The propagation delay of a complex combinational circuit comprises the sum of propagation delays of its gates along the longest path within the circuit, known as the critical path [10]. The propagation delay is determined by the critical path of logic circuit.

2.4.2 Sequential Logic

In opposite to combinational circuit the sequential logic has state (memory).in sequential logic clock is involve. It is combination of combinational logic with memory (Fig 3).

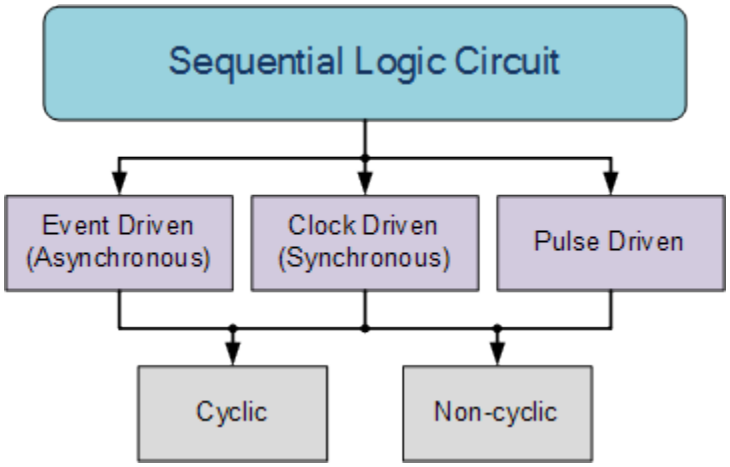


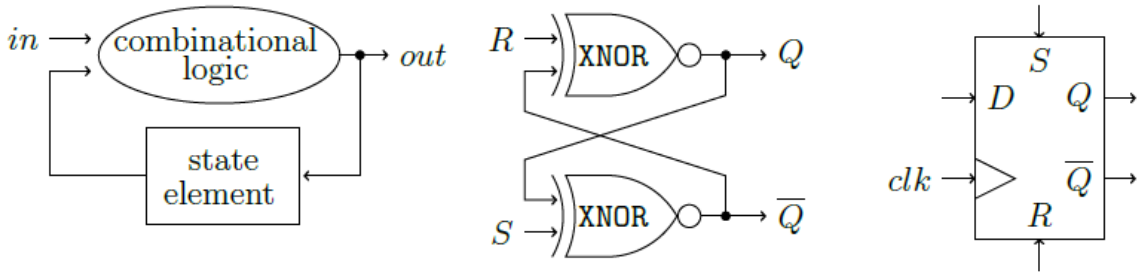
Fig 3. *Type of the sequential logic circuit.*

State element (Flip-Flop, latch, etc) are basic building block of sequential logic.

- 1. Event Driven – asynchronous circuits that change state immediately when enabled.
- 2. Clock Driven – synchronous circuits that are synchronized to a specific clock signal.
- 3. Pulse Driven – which is a combination of the two that responds to triggering pulses?

2.4.3 Asynchronous sequential logic

The most basic asynchronous sequential logic is SR latch which mean set reset [11]. It is not possible to neither construct by using two cross coupled NOR gate. When S and R both at logic low feedback loop ensure that the output will be in previous state. If $S = 1$ and $R = 0$ set output occur $Q = 1$ and $\bar{Q} = 0$. Similarly for reset input S=0 and R=1 reset output occur $Q = 0$ and $\bar{Q} = 0$. S and R are not allowed to give both logic high input because $Q = 0$ and $\bar{Q} = 0$not possible. SR latch is level sensitive (5 volt is consider as one state while zero volt as other). So these circuits can maintain state but it is purely driven by its input no synchronization is there known as asynchronous sequential circuit (Fig 4). Time delay is only due to the propagation delay of gates.



(a) **(b)** **(c)**

Fig 4. *Illustration of (a) Asynchronous sequential circuit with feedback, (b) SR (NOR) latch, (c) D-flip flop.*

2.4.4 Synchronous Sequential Logic

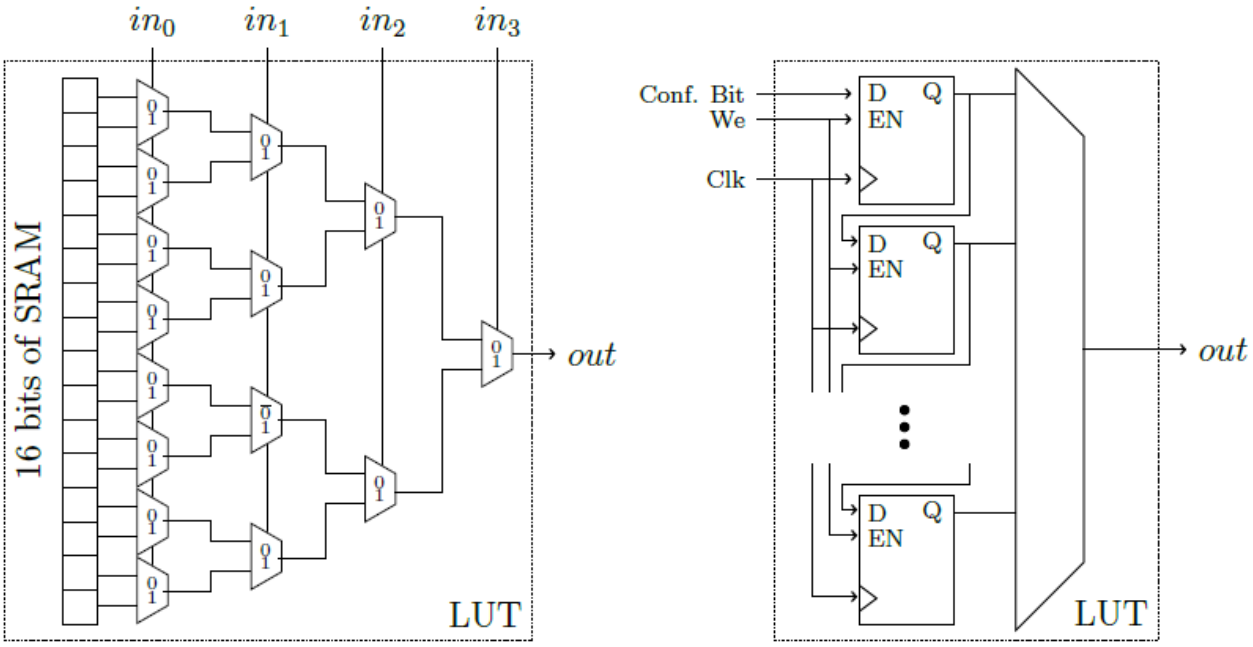
In synchronous sequential circuit all memory element is synchronized with fixed clock signal. It is sequential circuit can be positive edge triggered (0 to 1) or negative edge triggered (1 to 0). The basic sequential logic is D flip-flop. It is used to store the value of input at occurring of clock edges. D flip-flop is implemented using to latches with additional logic gates. The basic need of sequential logic is due to its simplicity and reliability. The whole operation of logic circuit should be getting completed in a single clock time. The clock time is defined by critical path of combination path in flip-flop.

2.5 The Key to Reconfigurable Hardware

A combinational logic is made up of basic logic gate connected through wire. In FPGA these gates are simulated using look-up table (LUT).

2.5.1 Internal Architecture of LUT

A, n-input LUT requires 2^n bits of SRAM to store lookup table and 2^n: 1 multiplexer to read individual bit. In general 4-input look-up table is used but nowadays 6 input look-up table is used (Fig 5).



(a) **(b)**

Fig 5. *Four input LUT (a) circuitry for read (b) circuitry for write.*

The value in0 to in3 are used to determine which SRAM bit is given at output. Boolean function is stored in SRAM cell. SRAM cell is simply a shift register with one-bit width and 2^n bit depth. The Bit is shifted bit-by-bit into LUT when FPGA is programmed.

LUT can also be used as memory element on FPGA. When FPGA is programmed LUT is used as distributed RAM. The multiple LUTs are combined to make wider or deeper memories.

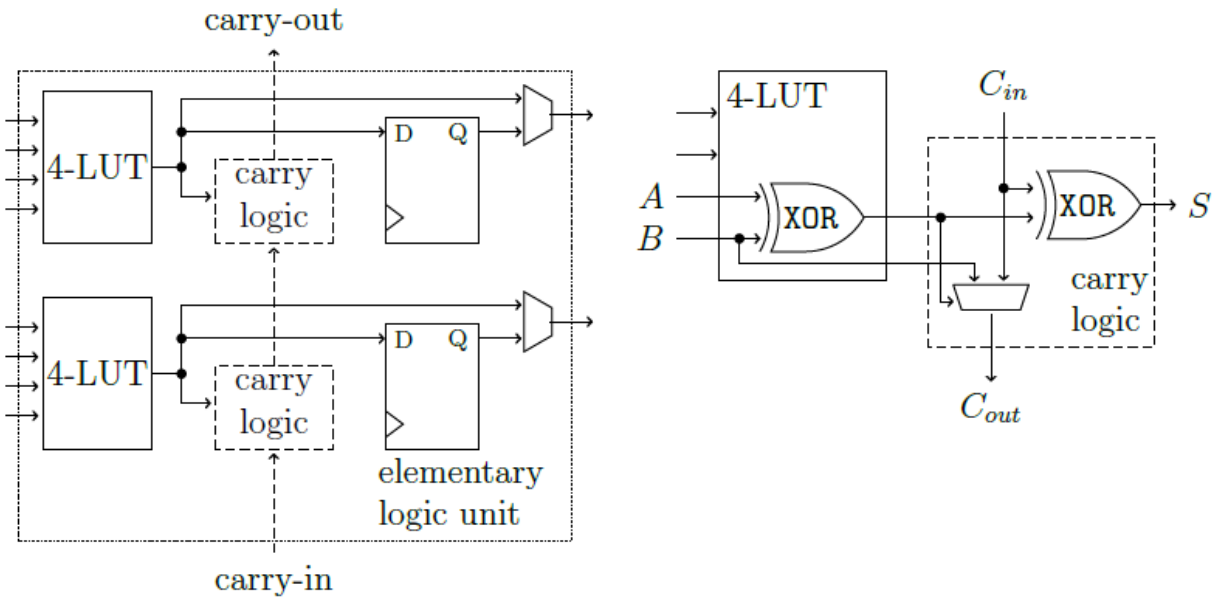
2.6 FPGA Architecture

After discussing LUT now we will discuss how these LUT are combined and distributed over entire FPGA architecture. Some fixed number of LUT are grouped and embedded on programmable logic device which is called as elementary logic unit (Xilinx refer it as slices).

Four main basic structural element of elementary logic unit are

1. Number of LUT
2. Proportional number of one bit register.
3. Several multiplexer.
4. Arithmetic/carry logic

An example of elementary logic unit is given in Fig 6. Which contain 4-input LUT and two Flip-flop register (Fig 6). Modern FPGA has more LUTs per elementary logic unit.



(a) **(b)**

Fig 6. *Functionality within an (a) elementary logic unit and (b) a full adder constructed by combining a LUT with elements of the carry logic (right).*

Each LUT is paired with a memory element which is flip-flop to store result. Flip-Flops are the second type memory in FPGA.

2.7 Routing Architecture

Direct wires to neighbouring elementary logic units allow combining multiple units to build more sophisticated circuits such as adders and multipliers. Modern FPGAs provide enough configurable resources to host an entire system on chip (SoC) [12]. To build that much complex network proper routing is required this is known as interconnect. The interconnections of the Logic Island are arranged as a two-dimensional array on FPGA. In this pattern any logic island can communicate to any logic island even at other end at matrix (Fig 7).

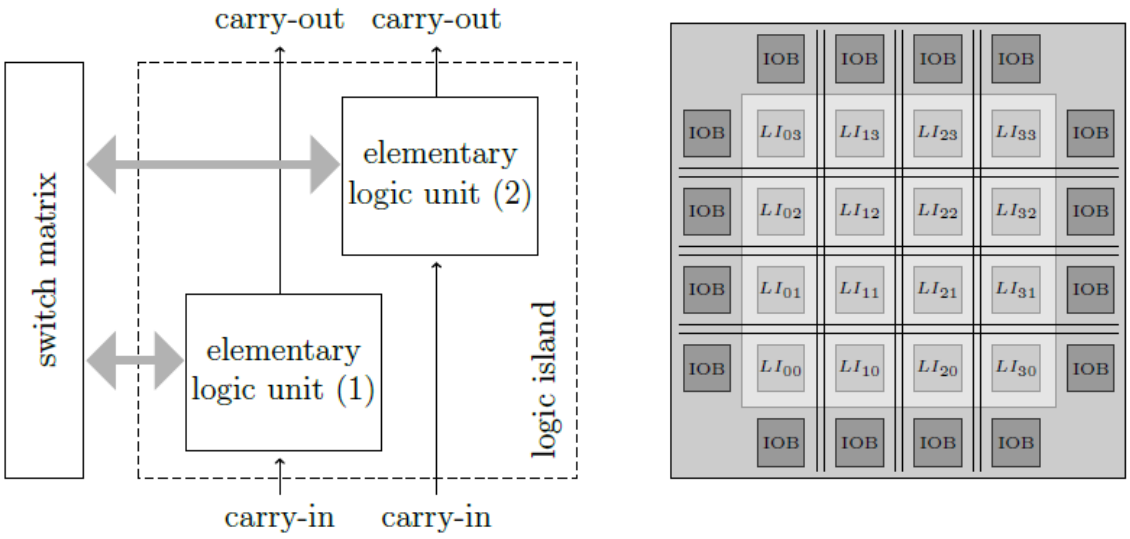


Fig 7. *Internal structure of Logic Island (left) and two dimensional arrangement of Logic Island with input output blocks.*

A small number of elementary logic units are grouped together into a coarser grained unit that we refer to as Logic Island. In this example logic island consist of two elementary logic units. Each logic element in Logic Island is separated set of wire which is adjacently connected to different logic island [13]. For general communication interconnection is done by switch matrix.

2.7.1 Interconnect

Interconnect is used for communicate between different logic island (LIs) these interconnect are configurable. It consists of horizontal and vertical channels (bundle of wire) and vertical channels forming grid contain a logic island in every grid cell (Fig 8).

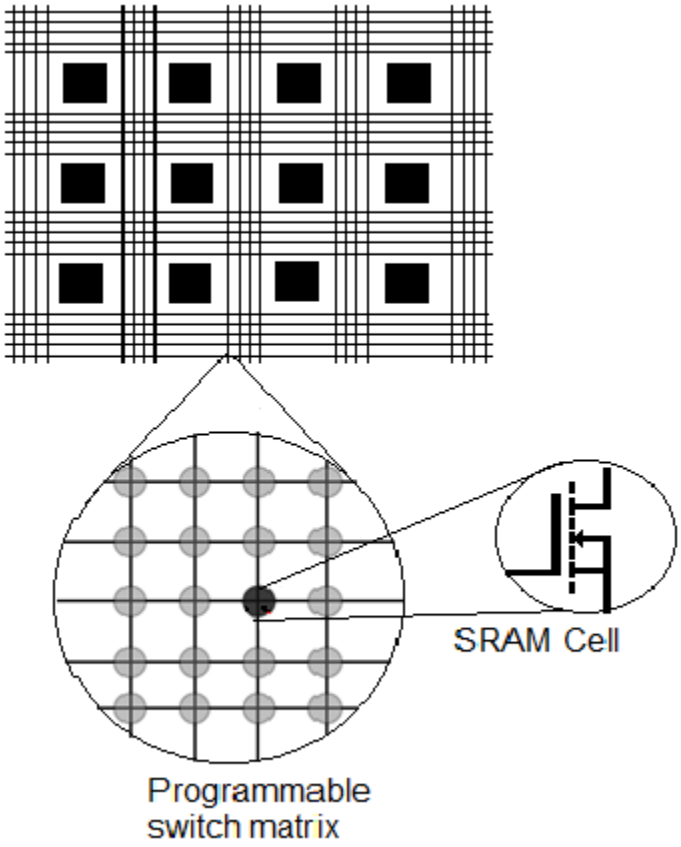


Fig 8. *Routing architecture with switch matrix, programmable links at intersection points, and programmable switches.*

At interconnect of routing channel there is a programmable links which determine how wire is connected, how input output is routed in particular logic island. All wires are connected to

additional three wires at interconnection point but which connection is active is determine by programmable switch.

2.8 High Speed I/O

For communicating with large number of logic islands on FPGA large number of I/O blocks are there at peripheral of FPGA. The IoBs are programmed to serve for different need and allow FPGA to communicate with multitude of other devices. Two min I/O standard are single ended or for high performance differential (used in SATA, 10G Ethernet etc.). IoBs can be programmed to implement the physical layer of many common communication schemes. High speed input output can be achieved with the help of extremely fast transreceivers. Few fast transreceivers are available in markets are VIRTEX-7 HT FPGA from Xilinx providing 28 Gb/s serial bandwidth each. Aggregate bandwidth is more than terabit can be achieved because it has sixteen 28Gb/s and seventy two 13Gb/s transreceiver.

2.9 FPGA Programming

To illustrate a typical FPGA design flow, we will examine the tools of the Xilinx tool chain, as well as their intermediate circuit representations. The most important steps and tools of the design flow to produce an FPGA-circuit are depicted in Fig 9.

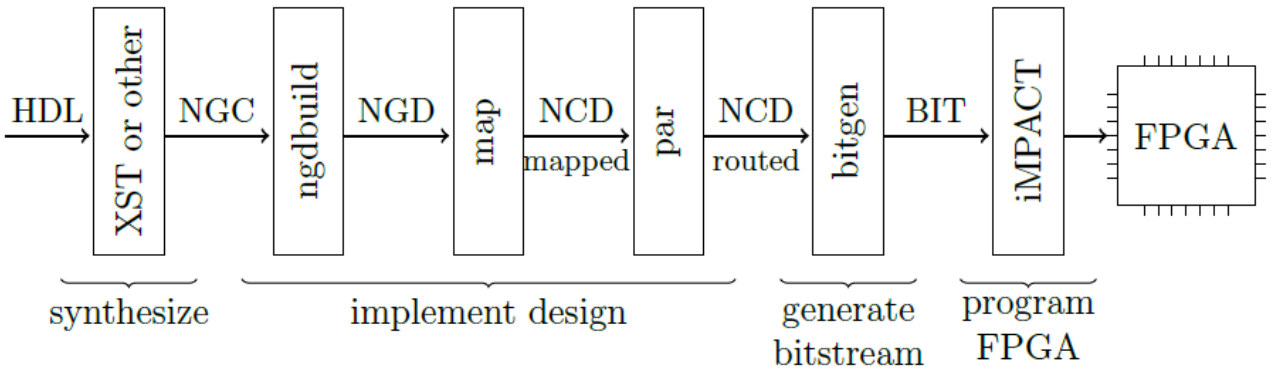


Fig 9. *FPGA design flow: Xilinx tool chain and intermediate circuit specification formats.*

Programming on FPGA is same as connecting wires in circuit this is done by using a hardware description language (HDL) such as VHDL or Verilog. Synthesizer converts HDL into gate level netlist, native generic circuit (NGC) format mapped to technology library provided by Xilinx. At this level third party synthesizer can also be used.

2.9.1 Translate

Tool **ngbuild** merge and translate all input netlists and constrain into a single netlist and save it as native generic database (NGD) file. User constrain file (UCF) is specify by the FPGA designer at time of manufacturing constrained are used for assigning special physical element of FPGA (e.g.: I/O pin, clock etc) in the design as well as timing require for design. NGC netlist is based on UNISIM library whereas NGD netlist based on SIMPRIM library. NGC allow behavioural simulation and NGD allow timing simulation [14].

2.9.2 Map

The map tool maps the SIMPRIM primitives in an NGD netlist to specific device resources such as logic islands, I/O blocks, etc. The map tool then generates a native circuit description (NCD) file that describes the circuit, now mapped to physical FPGA components.

2.9.3 Place and Route

The par tool performs placement and routing. The physical element specified in NCD file is specify at particular location on FPGA and interconnected. Place and route is most time consuming process in design flow it is based on simulated annealing algorithms. The par tool takes the mapped NCD file and generates routed NCD files which also contain the routing information.

2.9.4 Bitstream Generation

The implemented design has to dump on FPGA readable format. This is done by the **bitgen** tool, it encode design in binary known as Bitstream. Then Bitstream is loaded on FPGA using JTAG cable. Inside FPGA a finite state machine control by Bitstream which extract configuration data from Bitstream.

2.10 Auxiliary On-Chip Components

The logic resources of FPGAs discussed so far are in principle sufficient to implement a wide range of circuits. However, to address high-performance and usability needs of some applications, FPGA

vendors additionally intersperse FPGAs with special silicon components, such as dedicated RAM blocks (BRAM), multipliers and adders (DSP units), and in some cases even full-edged CPU cores. Hence, [HGV+08] observed that the model for FPGAs has evolved from a \bag of gates to a \bag of computer parts.

2.10.1 Block RAM (BRAM)

The BRAM is a configurable module, which is attached with variety of BRAM interface controllers. Figure 10 shows the BRAM blocks.

Main features

1. Fully automated generation and configuration of HDL through EDK Platgen/Simgen tools.
2. Number of BRAM primitives utilized is a function of the configuration parameters for: memory address range, number of byte-write enables, the data width, and the targeted architecture.
3. Both Port A and Port B of the memory block can be connected to independent BRAM Interface Controllers: LMB (Local Memory Bus), OPB (On-chip Peripheral Bus), PLB (Processor Local Bus), and OCM (On-Chip Memory).
4. Supports byte, half-word, word, and double word transfers provided the correct number of byte-write enables have been configured.

2.10.2 Digital Signal Processing (DSP) Units

FPGA is very widely used for digital signal processing (filter design, Fourier transform, convolution etc.), these all are mathematical based expression. Large number of adder and multiplier is incorporated in FPGA (hundred too few thousand). Adder and multiplier can be realising using LUTs and input-output in FPGA [15]. These all are power efficient high performance and minimum space circuit. Similar to other components of FPGA, the digital signal processing unit also needed and combine with the adjacent DSP section. For example we consider Xilinx DSP$*E slice which has three input ports of different width (25bits,18 bits and 48 bits) and it provide 25*18 bit multiplier with a pipelined second stage that can perform as subtractions or adder (48 bit) also with optimal accumulation feedback. The DSP unit of FPGA is used for various application like multiply, multiply and accumulate, multiply and add/subtract, three input addition , barrel shifting,

wide bus multiplexing etc are various operation which can perform on FPGA . It can also be used in different modes. These all function can be implemented in one or two clock cycle one over a one wide input. For implementing hash function fast multiplier is very useful.

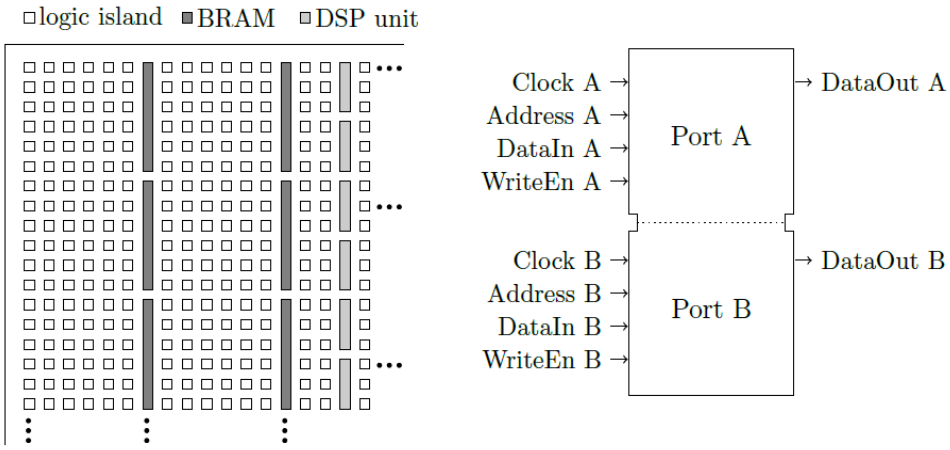


Fig 10. *FPGA layout with interspersed (a) BRAM blocks and DSP units, and (b) the simplified interface of a dual-ported BRAM block.*

2.11 Commercial FPGA Devices

Some commercial FPGA devices are available in market; Altra Stratix is one of them. It consist major five components all these are connected through programmable routing network.

These five components are:

1. Logic array blocks (LABs):- to implement logic array function.
2. Memory block:- for storing data
3. DSP unit:- for speed up mathematical operation like multiply and accumulate
4. Phased locked loop modules:- to alter the phase and frequency of input clock
5. I/O pads: - to connect to outside world.

Each logic array block contains ten logic elements (LE) and these can operate in normal mode or dynamic arithmetic mode. In normal mode each logic element is configure of four input lookup table (LUT) and a register. The output of Logic Element is output of lookup table or the

output of register whose data is also part of Look-up table. This type of mode is very useful in implementing the logic arbitrary logic functions. For implementing arithmetic operation logic element can be configure in dynamic arithmetic mode one for sum output and two is carry output signal. The carry output signal is connected to adjacent logic element through dedicated routing. The addition result is generated by one of the two input LUTs and it also depends on the carry input value, where each two LUT compute the sum for a possible carry-in of either 0 or 1 [16]. The both carry-out is generated similarly which depends on the carry input value either it is 0 or 1.

The Altra Stratix 2 is second generation of Altra Stratix. The difference to the earlier version is that it has modified logic array block used for implementation arbitrary logic function. The Stratix 2 logic array block has 8 adaptive logic modes (ALMs). Figure 11 shows high level schematic of adaptive logic mode.

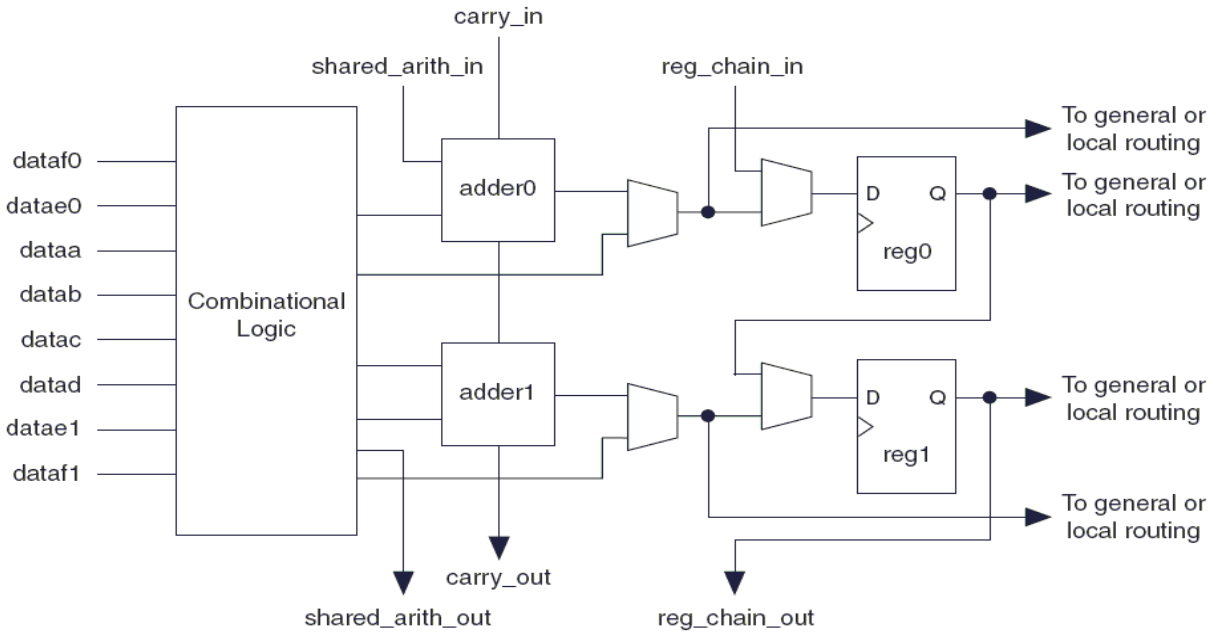


Fig 11. *High level diagram of the Stratix II ALM [Altera05b.]*

There is four mode of operation in ALM arithmetic, share arithmetic, extended LUT and normal mode. There is four output in normal mode of operation, two is generated by register of ALM and other two are generated by combination logic circuit. these are configure in several different ways : a pair of four input LUTs, a pair of five input LUTs, a pair of six input LUTs, a five and a three input LUTs , a five and a four input LUTs and many more. In latter three cases, some input signal is

share between different LUTs. It can also implement 6 input and one output function. Also using extended LUT mode, 7-input function can also be implemented in single ALM.

In the arithmetic mode ALM is suitable for circuit such as accumulator and adder, in addition to four outputs in normal mode one more carry output also produces. The carry out of one ALM is carry-in signal for adjacent ALM and connected through routing for fast propagation.

In similar to the arithmetic mode share arithmetic is also same except it has additional carry chain. In shared arithmetic mode the second carry chain is fed into next adder which is in same or in adjacent ALM. For second carry chain carry-in signal to share_airth_in and carry-out signal is the shared_airh_out signal. Due to the presence of two carry chains in ALM it is possible to implement a circuit that can add three two bit numbers. This design is suitable of designing adder trees.

Summery

In this chapter, we discussed requirement of FPGA and evolution of FPGA, design architecture, basics of combinational and sequential circuit implementation using FPGA. We also further discussed regarding the implementation of asynchronous and synchronous sequential logic circuit. The internal architecture of LUT base circuit also discussed here. Stratix ALM also briefly discussed in this chapter.

References

[1] Meyer-Baese, U., & Meyer-Baese, U. (2007). Digital signal processing with field programmable gate arrays (Vol. 65). Berlin: springer.

[2] Monmasson, E., & Cirstea, M. N. (2007). FPGA design methodology for industrial control systems—A review. IEEE transactions on industrial electronics, 54(4), 1824-1842.

[3] Weste, N., Harris, D., & Banerjee, A. (2005). Cmos vlsi design. A circuits and systems perspective, 11, 739.

[4] Sulaiman, N., Obaid, Z. A., Marhaban, M. H., & Hamidon, M. N. (2009). Design and implementation of FPGA-based systems-a review. Australian Journal of Basic and Applied Sciences, 3(4), 3575-3596.

[5] Roy, S., & Bhunia, C. T. (2014, January). Minimization algorithm for multiple input to two input variables. In Control, Instrumentation, Energy and Communication (CIEC), 2014 International Conference on (pp. 555-557). IEEE.

[6] Ghosh, S., & Ray, S. S. (2007). 4th Generation Programmable Logic Computing: A Road Map. IETE Technical Review, 24(6), 439-452.

[7] Xia, Q., Robinett, W., Cumbie, M. W., Banerjee, N., Cardinali, T. J., Yang, J. J.,. & Snider, G. S. (2009). Memristor− CMOS hybrid integrated circuits for reconfigurable logic. Nano letters, 9(10), 3640-3645.

[8] Brown, S., & Rose, J. (1996). Architecture of FPGAs and CPLDs: A tutorial. IEEE Design and Test of Computers, 13(2), 42-57.

[9] Kumar, R., Roy, S., & Bhunia, C. T. (2016). Study of Threshold Gate And CMOS Logic Style Based Full Adder Circuits. In Proc. IEEE, 3rd Int. Conference on Electronics and Communication Systems (ICECS), IEEE (pp. 173-179).

[10] Roy, S., & Bhunia, C. T. (2015). On synthesis of combinational logic circuits. International J of Computer Applications, 127(1), 21-6.

[11] McAuley, A. J. (1992). Four state asynchronous architectures. IEEE transactions on computers, 41(2), 129-142.

[12] Wang, X., & Ziavras, S. G. (2004). Parallel LU factorization of sparse matrices on FPGA-based configurable computing engines. Concurrency and Computation: Practice and Experience, 16(4), 319-343.

[13] Smith, S. C. (2007). Design of an FPGA logic element for implementing asynchronous NULL convention logic circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 15(6), 672-683.

[14] Tee, H. H. (2010). FPGA unsolicited commercial email inline filter design using Levenshtein distance algorithm and longest common subsequence algorithm (Doctoral dissertation, University of Malaya).

[15] Sasao, T., Nagayama, S., & Butler, J. T. (2007). Numerical function generators using LUT cascades. IEEE Transactions on Computers, 56(6), 826-838.

[16] Cherepacha, D., & Lewis, D. (1996). DP-FPGA: An FPGA architecture optimized for datapaths. VLSI Design, 4(4), 329-343.

Authors

Gopal Krishna: M.Tech in VLSI & Embedded System Design (Department of Electronics & Communication Engineering) from National Institute of Technology, Arunachal Pradesh, India. Completed B.Tech from Poornima College of Engineering, Jaipur, under Rajasthan Technical University.

krishna100491@gmail.com

Sahadev Roy: Author's is currently working as an Assistant Professor in the Department of Electronics & Communication Engineering at National Institute of Technology, Arunachal Pradesh, India. He has more than 13 years' experience in research and teaching field. He received his B.Sc. degree in Chemistry from Burdwan University, B.Tech degree in Electronics and Communication Engineering from Kalyani University and M.Tech in Mechatronics from Bengal Engineering and Science University, Shibpur India. He received PhD in Electronics and Communication Engineering from National Institute of Technology, Arunachal Pradesh. He has published more than 50 research articles in International referred journals. He served as the Editor-in-Chief of International Journal of Advanced Engineering and Management (ISSN 2456-8066) monthly blind peer-reviewed journal published by the Technical and Scientific Publisher. His research interest includes Robotics, Microwave and VLSI design.

sahadevroy@gmail.com