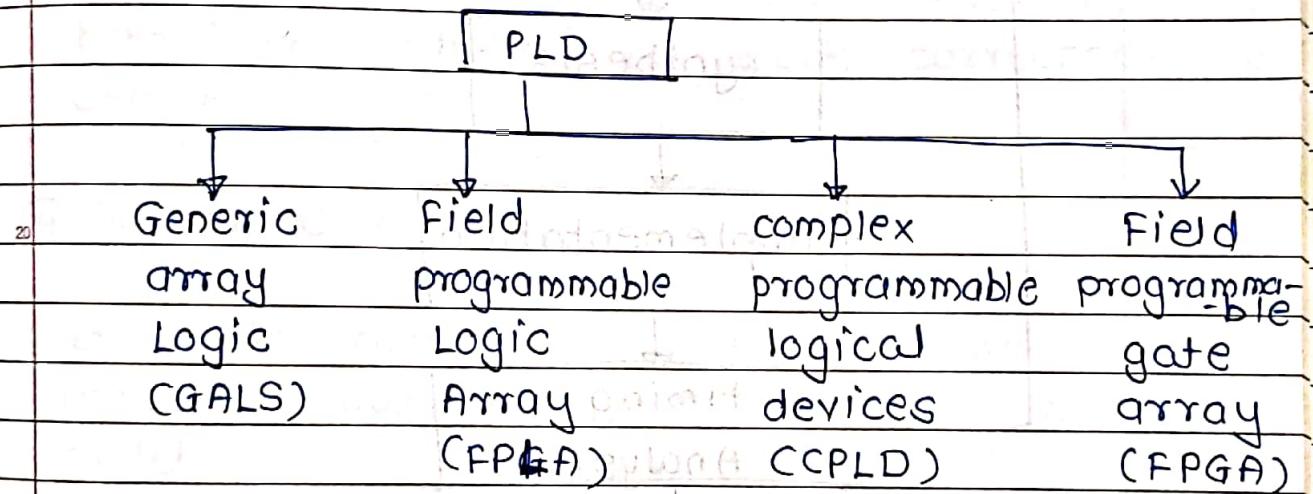


PLD Architectures of Applications

Syllabus :- Design flow, CPLD architecture, features, specifications, appl', FPGA architecture, features, specification, appl', simulation and synthesis tools, FPGA synthesis and implementation

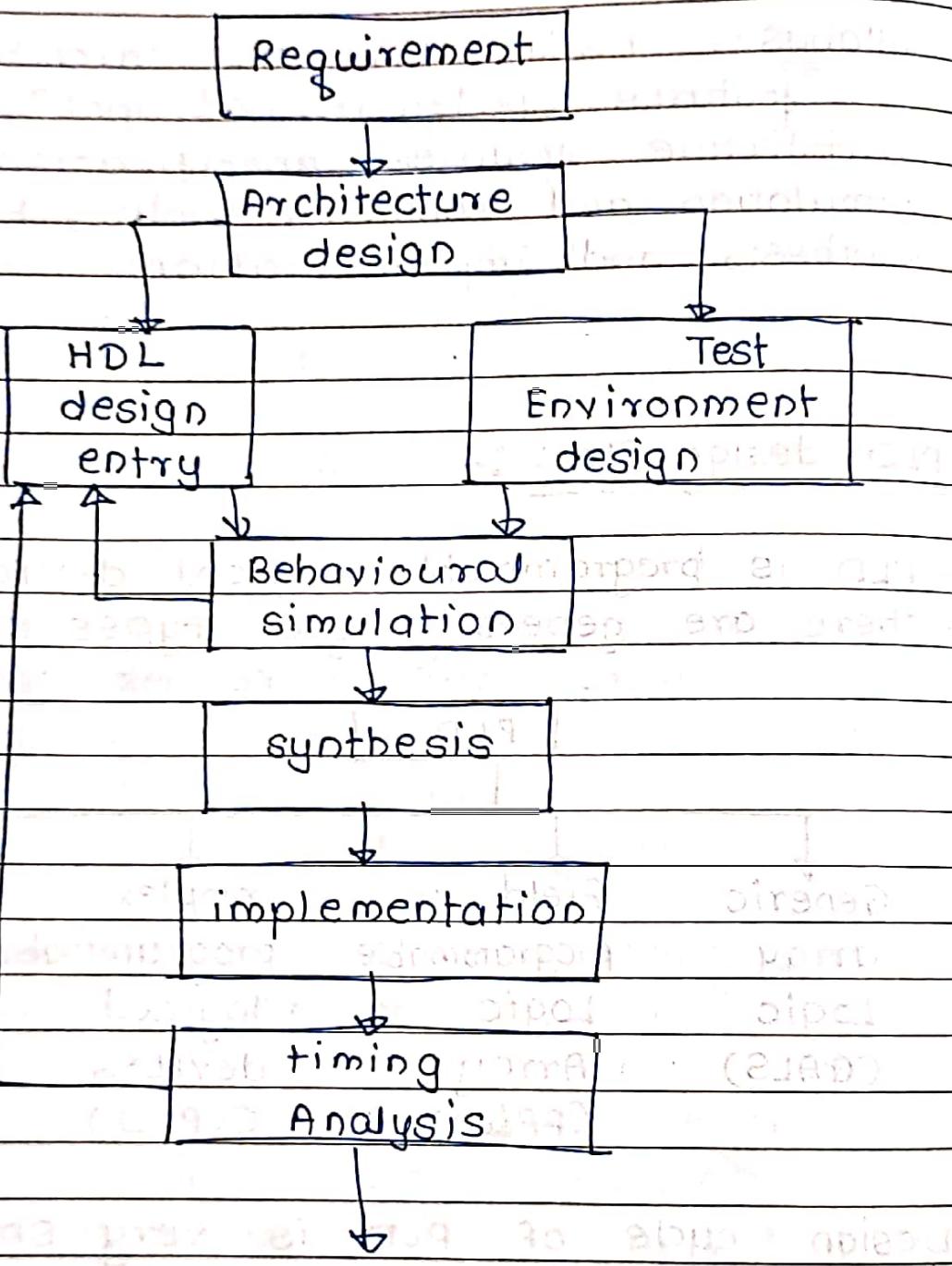
* PLD design flow :-

- PLD is programmable logical devices.
- There are generally four types of PLD's.



- Design cycle of PLD is very short, .
it has faster time to market.
- product updates are smooth and .
it has lowest investment risk.
- PLD are cheaper for low volume produ'
- Many types of PLD's are available according to the application.

- The flow diagram of PLD design are shown in following fig.



① Architecture design :-

This stage consists of analysis of -

- project requirement
- problem decomposition
- functional simulations

- Document of output of this stage is document which gives info about -

- device architecture
- structural block & their funⁿ
- interfaces

② HDL design entry :-

In this step design is described in hardware description languages such as -

- a) verilog
- b) VHDL

③ Test Environment design :-

It consists of writing a test environment and behaviour modelling to check correctness of design.

④ Behavioral simulation :-

It checks correctness of HDL design by comparing output of HDL model & behavioural model.

⑤ Synthesis :-

- It converts HDL description to netlist. which is digital schematic.
- synthesis is performed by HDL synthesizer.
- for correct HDL code, synthesis should not be any problem.

⑥ Implementation :-

Implemented maps the synthesised netlist on device's internal structure. Implementation consists of placement and routing.

⑦ Timing Analysis :-

It checks whether the implemented design satisfies timing constraints.

* CPLD Architecture :-

- complex programmable logic devices (CPLD) architecture consist of following main blocks.

- 1> PAL block
- 2> Interconnect Matrix
- 3> Input Output blocks.

1> PAL blocks (function block-FB)

- Programmable Array Logic (PAL) block is functional building block which consists of array of AND and OR gates.

- AND gates accept input from IO blocks, these term or Ored using six nos of OR gates.

- Each PAL block consists of

- AND-OR gates

- MUX

- clocked flip flop

- PAL logic is implemented using sum of product form.

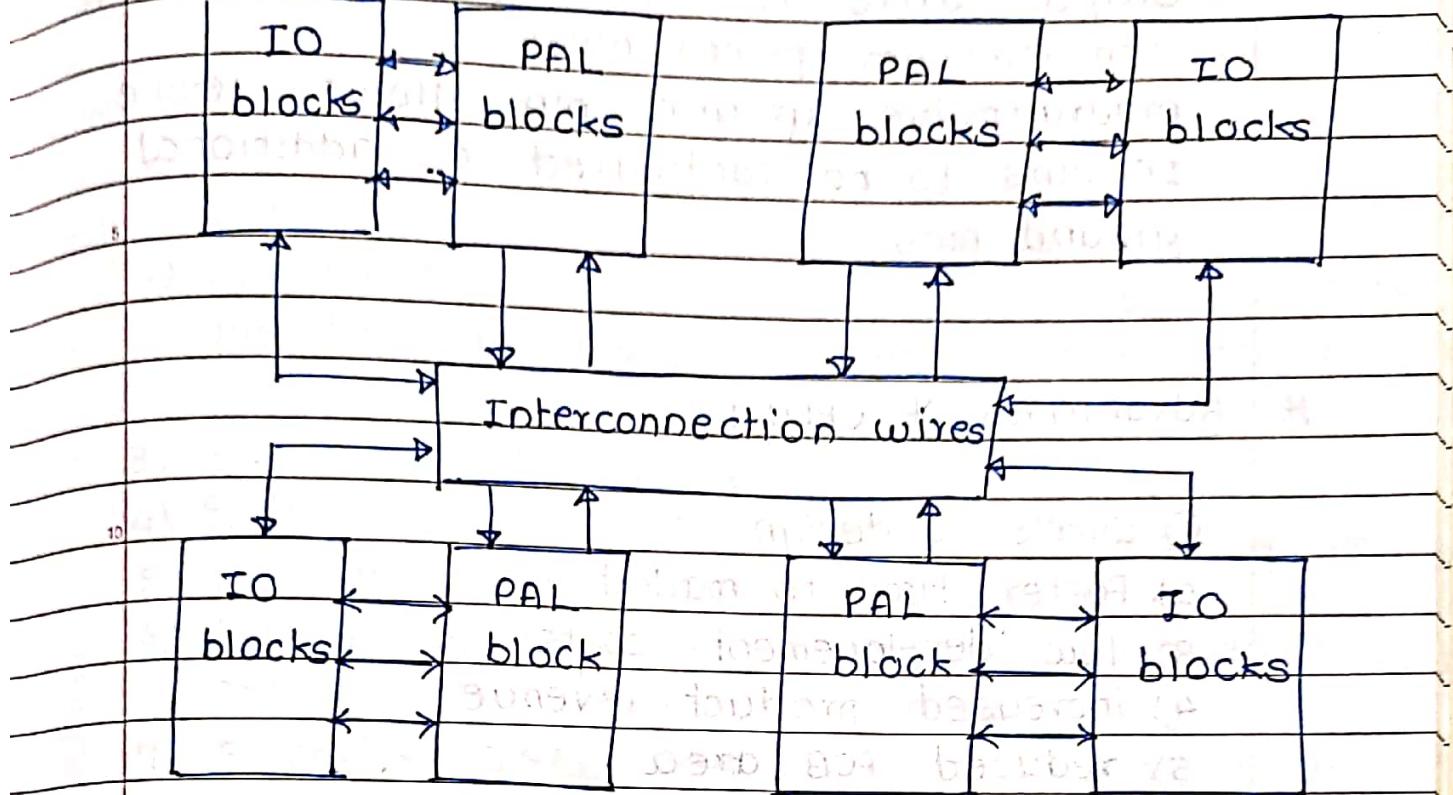


Fig. Arch^r of CPLD

2) Interconnect Matrix of based and L3.

- The interconnect matrix connects signals to function blocks, all IOB outputs and all o.
- FB output drives the fast connect matrix.
- Any of these may be selected to drive each FB with uniform delay.

3) IO block :-

- The IO block interfaces between internal logic and the device IO pins.
- Each IO block consists of input buffer, output driver, output enable selection multiplexer and user programmable ground control.
- Each input buffer is compatible with 5V TTL, 3.3V CMOS, and 2.5V CMOS signal.

- output drive is used for fast switching with minimum power noise.
- programmable ground pin allows device IO pins to be configured as additional ground pins.

* Advantages of CPLD :-

- 1) Simple to design
- 2) Faster time to market
- 3) low development costs
- 4) increased product revenue
- 5) reduced PCB area
- 6) lower cost
- 7) large no of input output pins
- 8) can be used for complex logic designs

* Applications of CPLD :-

- Application of CPLD's are given as -
- 1) graphics controller design
 - 2) UART, cache controller, LAN controller design
 - 3) random glue logic to, prototyping small gate array
 - 4) battery operated equipment
 - 5) Hardware platform for VHDL/ Verilog logic sign courses
 - 6) complex state machines
 - 7) wide detectors
 - 8) Fast memory interfaces

* features of CPLD :-

Features of xc9500 CPLD are -

- ① Large density of macrocells (36 to 288) with 800 to 6400 usable gates.
- 2) Optimized for high performance 2.5V systems due to 5ns pin to pin logic delay on all pins.
- 3) Superior pin locking and routability.
- 4) Endurance of 10000 program/erase cycles.
- 5) Flexible 8x18 functional block.
- 6) Global and product term clocks, output enables, set and reset signal.
- 7) Extensive IEEE std 1149.1 boundary scan (JTAG) support.
- 8) Programmable power reduction mode in each macro cell.
- 9) User programmable ground pin capability.
- 10) Settle rate control on individual outputs.
- 11) Extended pattern security features for design protection.
- 12) 3.3V or 5V IO capabilities.
- 13) High drive 24mA outputs.
- 14) Advanced CMOS 5V fast flash technology.
- 15) Supports parallel programming of multiple XC9500 devices.

* Specifications of CPLD :-

a) Packaging and power requirement specification :-

- IC package
- supply voltage
- operating current
- Power dissipation

b) Memory :-

- RAM
- ROM
- Content addressable memory (CAM)

c) Performance specification :-

- Internal frequency
- propagation delay
- speed grade

Real time specification of coolrunner-II

CPLD are given below:-

core voltage = 1.8V

Macrocells = 82-512

I/Os = 21-270

IO tolerance = 1.5V, 1.8V, 2.5V, 3.3V

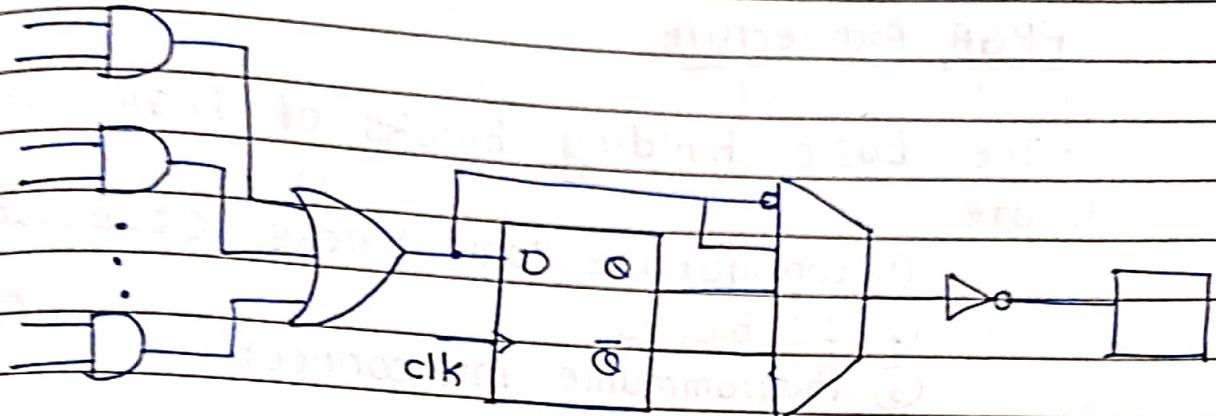
TPD IF max = 3.8 / 823

ultra low standby power = 28.8 uW

* Macrocell :-

Macrocell is basic functional block of CPLD. The macrocell array is prefabricated array of high level logic function such as

flip flop, ALU function, registers etc.



AND-OR Gate array

D-Flip flop

Registers

to IO block

Fig. Arch^r of macrocell

- These logic functions are simply placed at regular preditined position and manufactured on a wafer, usually called master slice.
- Creation of circuit with a specified function is accomplished by adding metal interconnects to the chips on the master slice late in the manufacturing process, allowing the function of chip to be customized as desired.
- Macrocell consists of AND-OR gate array, D flip flop, registers etc.
- Macrocell implements the function in sum of product form.
- Macrocell array master slices are usually prefabricated and stockpiled in large quantities regardless of customer order.
- The fabrication according to individual customer specification may be finished in a shorter time compared to standard cell or full custom design.

* FPGA :- Field Programmable Gate Array

FPGA Architecture

- The basic building blocks of FPGA archr are -

- ① configurable logic blocks (CLB) (Logic cell)
- ② IO blocks
- ③ Programmable interconnect

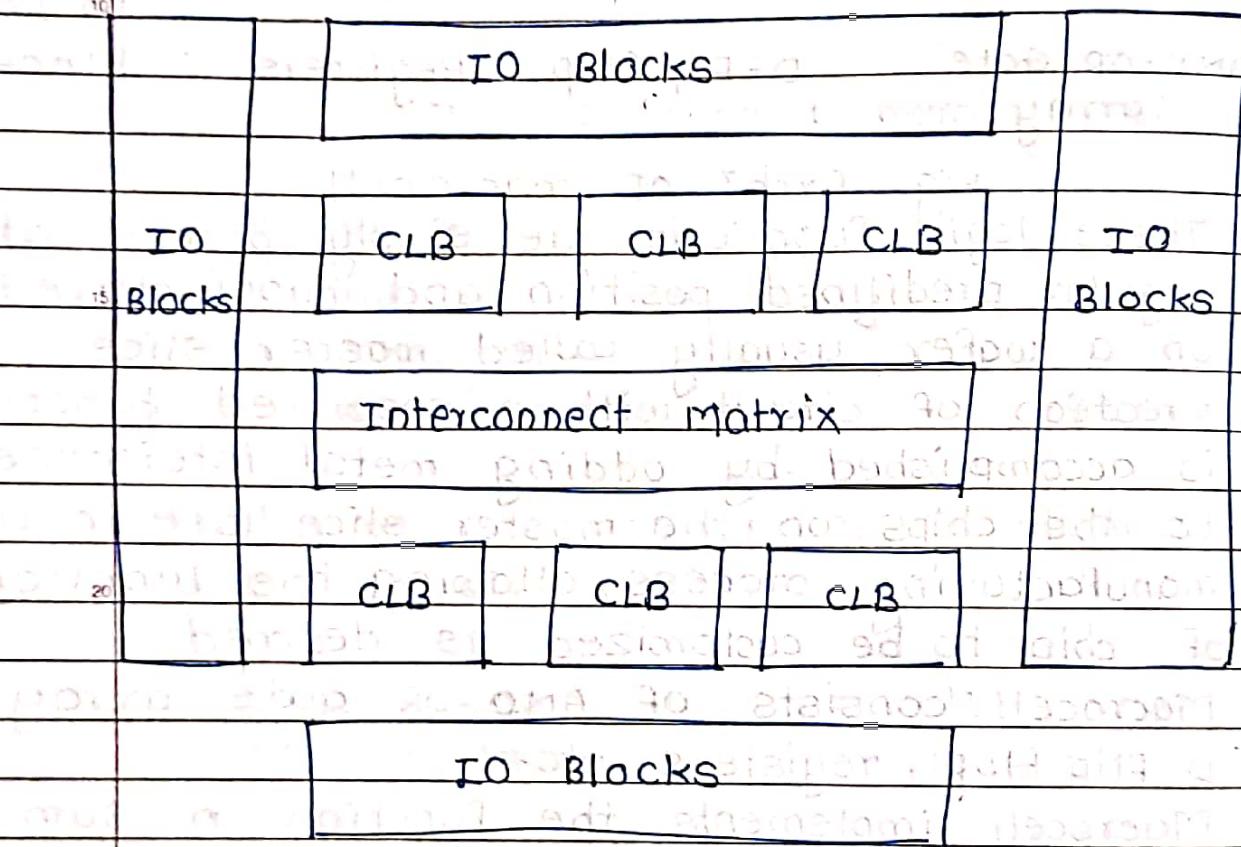


fig. Architecture of FPGA

a) Logic block

- Each logic block has smaller number of input and outputs.
- The most commonly used logic block is lookup table which consists of storage cells that are used to implement a small

- logic function.
- Each cell is capable of storing single logic value ie. 0 or 1.
- LUT's are available in two input or three input LUT.
- circuit dia for two input LUT is shown as,

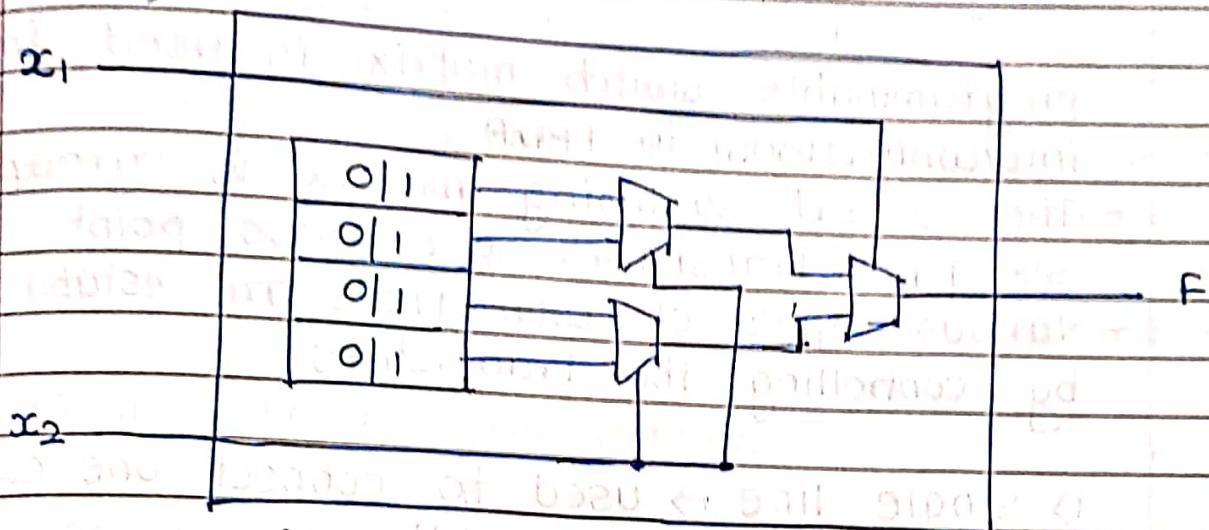


fig. circuit for 2 i/p LUT

- When this circuit is implemented in FPGA, the logic blocks are programmed to realise the necessary function and the routing channels are programmed to make the required interconnection b/w logic blocks.

b) I/O Blocks

- FPGA are actually be used to interface betⁿ diff I/O standards.
- Modern FPGA output signals with fast edge requires termination to prevent reflections and maintain signal integrity.
- High pin count packages can not accommodate external termination register.

- Thus, a digitally controlled impedance (DCI) is used which eliminates the need of external register and improves signal integrity.

c) Programmable interconnect

- programmable switch matrix is used for interconnections in FPGA.
- The actual switching matrix is group of six pass transistors per cross point.
- Various types of connections are established by controlling the transistors.

i) single line → used to connect one CLB to another

ii) double lines → these wires travels past two CLBs before hitting the switch; these are used for longer conn' due to shorter delay.

iii) long lines → Long lines travels all the way across or down a row or column and are driven by tri-state driver.

iv) global clock line → These lines are optimized for case as clock input to the CLB, providing short delay and minimum skew.

* Advantages of FPGA :-

- i) very fast custom logic
- ii) massive parallel operation
- iii) much faster than DSP engines
- iv) faster than microcontroller and micro-processor
- v) more flexible than ASIC
- vi) more affordable and less risky than ASIC
- vii) allow unlimited product differentiation
- viii) reprogrammable at any time
- ix) no NRE, minimum order size or inventory risk.
- x) less power consumption

* Applications of FPGAs :-

Typical applications of FPGA are -

- i) random logic design
- 2) integrating multiple SPLD's
- 3) device controllers
- 4) communication encoding & filtering
- 5) small to medium size sys with SRAM
- 6) simple state machines
- 7) board integrations
- 8) complex controllers/ interfaces
- 9) software defined radio
- 10) Router, switches and gateway
- 11) Data mining systems
- 12) Industrial imaging
- 13) computer hardware emulation

* Features of FPGA

- FPGA consists of few thousand to the several million gates.
- FPGA architecture is more granular compared to CPLD.
- FPGA consumes less power than CPLD.
- FPGA can't implement large functions in one pass through the logic array.
- FPGA contain array of logic cells and are less partitioned than CPLD.
- A number of input output pins are lesser.
- Lowest switch resistance.
- very low capacitances
- Non-volatile (Antifuse FPGA only)
- software is easy to place and route
- very fast and custom logic
- more flexible than dedicated chipset
- faster than microcontrollers and microprocessors.
- More flexible than dedicated chipset.
- Reprogrammable at anytime.

* FPGA specifications:-

The specifications of XC8S200 FPGA are,

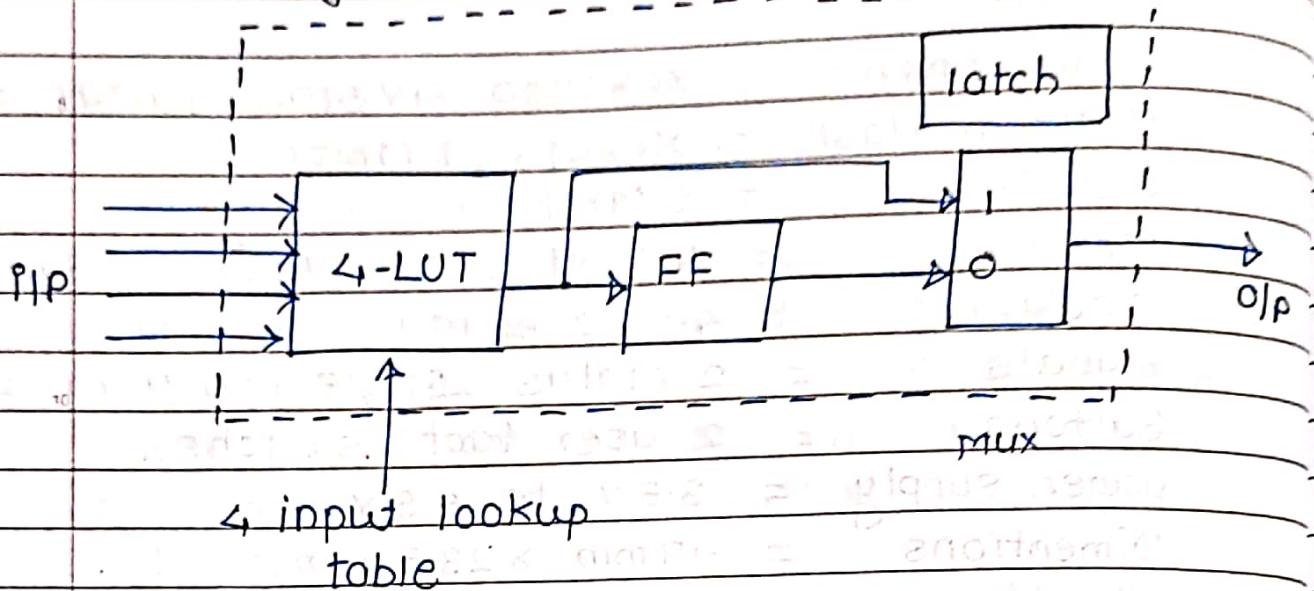
xilinx FPGA	= XC8S200 -LY0100 Spartan 3
platform flash	= XCFO1S, 1 Mbit
SPI Flash	= 8 Mbit
memory	= 4 Mbit SRAM, 512×8 , 55 ns
clock signals	= 49.152 MS MHS
Buttons	= 2 status LED, 8 low user LED
power supply	= 3.5 V to 5.5 V
Dimensions	= 47 mm x 23.5 mm
Weight	= 12 gm

* configurable logic block :- (CLB)

- Configurable logic block is a basic functional block of an FPGA often known as CLB.
- The most common FPGA architecture consists of an array of logic blocks called as CLB or logic array blocks.
- Generally all the routing channels have the same width.
- Multiple IO pads may fit onto height of one row or the width of one column in the array.
- An application circuit must be mapped into an FPGA with adequate resources.
- While the no of CLBs and IOs required is easily determined from the design, the no of routing tracks needed may vary.

considerably even among designs with same amount of logic.

- The general arch^t of CLB is shown as,



- In general, a CLB consists of few logical cells. A typical cell consists of 4 input LUT, a full adder, D flip flop as shown in above fig.

- The LUTs are split into two 3 input LUT's. In normal mode those are combined into 4 input LUT using a mux.

- In arithmetic mode, their outputs are fed to FA.

- The output can be either synchronous or asynchronous, depending on the programming of the mux.

- In practice, entire part of a full adder are put as functions into the 4 LUTs in order to save space.

* Difference betn CPLD and FPGA :-

CPLD

FPGA

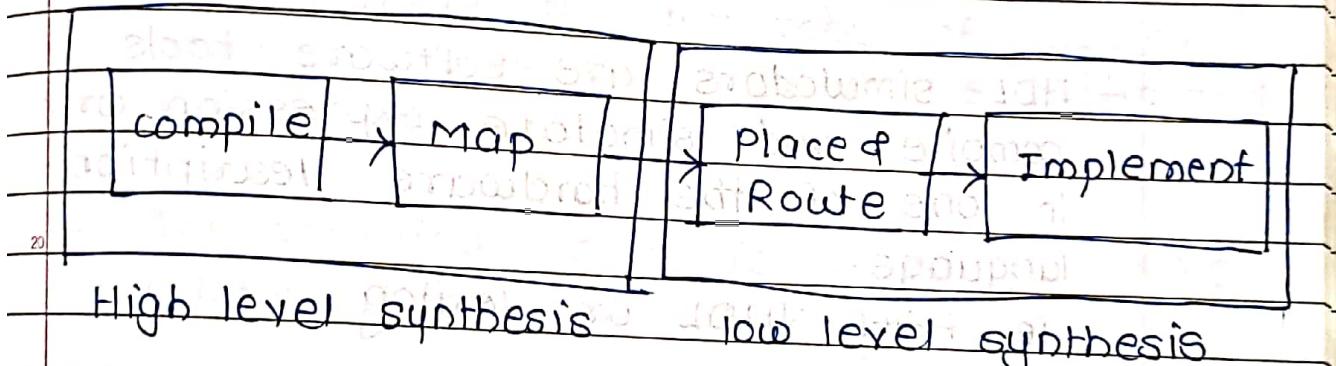
- 1) CPLD is complex programmable logical devices.
 - 2) CPLD are coarse grain devices. i.e. few blocks of logic with flip flop and combinational logic.
 - 3) Macrocell is basic building block.
 - 4) CPLD do not need boot ROM.
 - 5) CPLD can be used for small designs only.
 - 6) CPLD starts working as soon as they are powered up. (instant on)
- 1) FPGA is fine grain devices. i.e. they contains 100 to one million tiny logic block called as LUT.
 - 2) configurable logic block is basic building block.
 - 3) FPGA need boot ROM.
 - 4) for very large digital designs.
 - 5) FPGA has to load configuration data from external ROM and setup the fabric before it can start functioning, there is a time delay between power ON and FPGA starts working.

- 7) Non-volatile. CPLD remains programmed and retain their ckt after powering down.
- 7) FPGA goes blank as soon as powered off.
- 8) since CPLD are more simpler than FPGA and the no of interconnects are less, timing analysis becomes easier.
- 8) Density of FPGA is large. ∴ timing analysis is less deterministic.
- 8) lower idle power consumption.
- 8) Higher idle power consumption.
- 10) cheaper for implementing simple ckt.
- 10) FPGA are expensive.
- 11) More secure due to design storage within built in non volatile memory.
- 11) Less secured as FPGA uses external memory.
- 12) very small amount of logic resources.
- 12) massive amount of logic and storage
- 13) process technologies are EPROM, EEPROM and flash.
- 13) Process technologies are SRAM, antifuse & EEPROM
- 14) ideal for complex block with large I/O.
- 14) ideal for simple block with few input.

* FPGA simulation and synthesis tools :-

simulation describe the behavior of the ckt in terms of input signals, the output signal, knowledge of delay, behaviour described in terms of occurrences of events and waveform on signal.

- synthesis is reverse process which interface hardware from description. The synthesis tool will infer a hardware architecture from VHDL model.
- When writing VHDL code, think of the hardware that synthesis tools would infer from it.



technology independent technology dependent.

- Synthesis can be high level or low level.
- High level synthesis can be done at compile and mapping time.
- low level synthesis can be done at placement, routing and implement time.

- compile time synthesis is technology independent but mapping, placement, routing and implement time synthesis is technology dependent.
- Synthesis is process in which abstract form of desired ckt behaviour ie. register transfer level logic is converted in to design implementation in terms of logic gates.
- some synthesis tools generates bitstreams for programmable logic devices such as PALs or FPGA; while other targets in the creation of ASIC.
- HDL simulators are software tools that compile and simulate expression written in one of the hardware description language.
ie. VHDL or Verilog
- some common FPGA simulation tools are-
 - ① ISE simulator (Xilinx - VHDL93)
 - ② Incisive Enterprise simulator
(cadence design systems - VHDL)
 - ③ ModelSim (Mentor graphics - VHDL)

* FPGA synthesis

- synthesis is the process which translates VHDL or Verilog code into device netlist format.

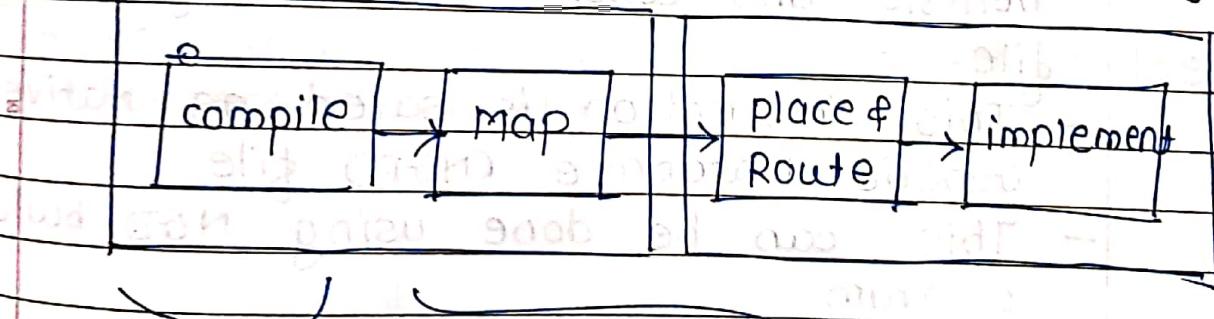
Verilog or VHDL code

synthesis

NGC file

- synthesis process will check code syntax and analyse the hierarchy of design which ensures that design is optimized for design archt.
- The resulting netlist is saved to NGC (Native Generic circuit) file.

High Level Synthesis Low Level Synthesis



technology
independent

technology dependent

- High level synthesis can be done at compile and mapping time.
- Low level synthesis can be done at placement, routing and implementation time.
- Synthesis at compile time is technology independent.
- Synthesis at mapping, placement, routing & implementation time is technology dependent.

* FPGA Implementation :-

FPGA implementation consists of three steps

- ① Translate
- ② Map
- ③ Place and route

① Translate

- Translate process combines all the input netlists and constraint to a logic design file.
- This information is saved as Native Generic Database (NGD) file.
- This can be done using NGD build program.
- Here, defining constraint means assign the targeted device to the physical elements such as pin, switches, buttons of targeted device.

- This information is stored in the file named UCF (User constraints file).

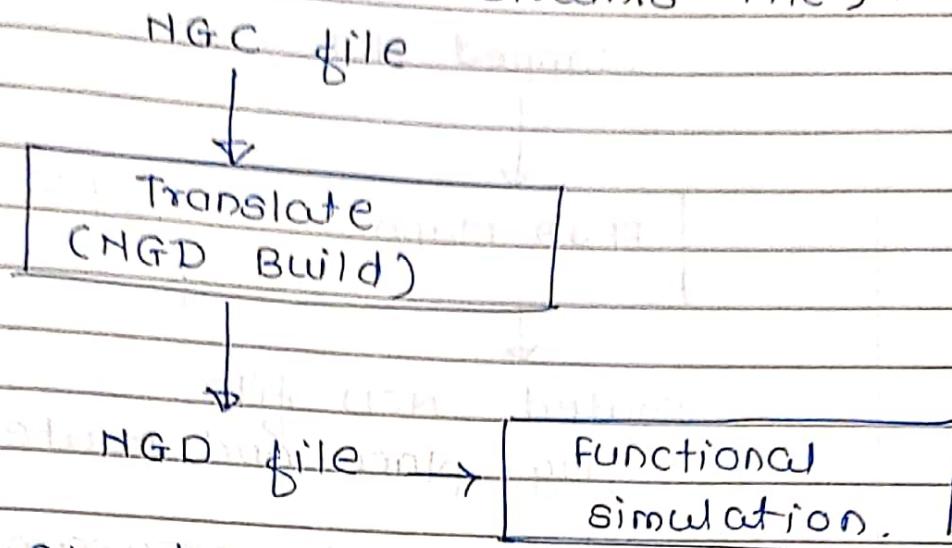


Fig. translate process.

② Map

- Map process divides the whole circuit with logical elements into subblocks such that they can benefit into FPGA logic blocks.
- Map process fits the logic defined by NGD file into targeted FPGA elements such as configurable logic blocks, input output blocks and generates NCD file (Native Circuit description file).
- NCD represents the design mapped to the components of FPGA.

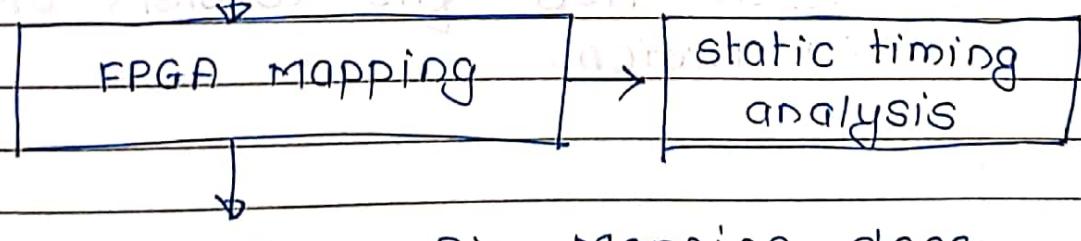
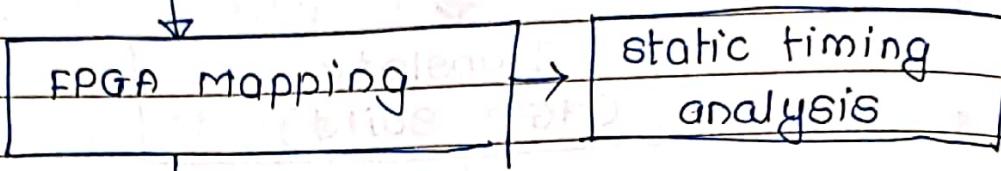


Fig. Mapping stage.

③ Place and Route :-

Mapped NCD File



Routed NCD file

Fig. place and route stage.

- place-and-route program (PAR) is used in this stage.
- The place-and-route process places the subblocks from map process into logic blocks according to constraints and connects the logic blocks.
- If a sub-block is placed in a logic block which is very near to an I/O pin, then it may save the time but affect some other constraint.
- So, a trade-off between all the constraints is taken into account by the place-and-route.
- The PAR tool takes the mapped NCD file as input and produces completely routed NCD file as output.
- Output NCD file consists of routing information.