KTU NOTES

ECT 304: VLSI Circuit Design

Module 1

Introduction

- VLSI stands for Very Large Scale Integration.
- A circuit consists of discrete components such as resistors, capacitors, diodes, transistors etc., connected are called discrete circuit.
- In an integrated circuit, the entire circuitry, i.e., all the passive and active components are housed on the same substrate.
- Depending on the complexity of the integrated circuit, we have different levels of integration.

Introduction

Level of Integration	No. of transistors per chip	Typical product
SSI (Small Scale Integration)	10 – 100	Logic gates, Flip Flops.
MSI (Medium Scale Integration)	100 – 1000	Counters, Multiplexers, adders.
LSI (Large Scale Integration)	1000 – 10,000	8-bit micro-processor, RAM, ROM
VLSI (Very Large Scale Integration)	10^4 – 10^7	16 & 32 bit micro-processor, DRAM
ULSI (Ultra Large Scale Integration)	10^7 – 10^9	Special processors

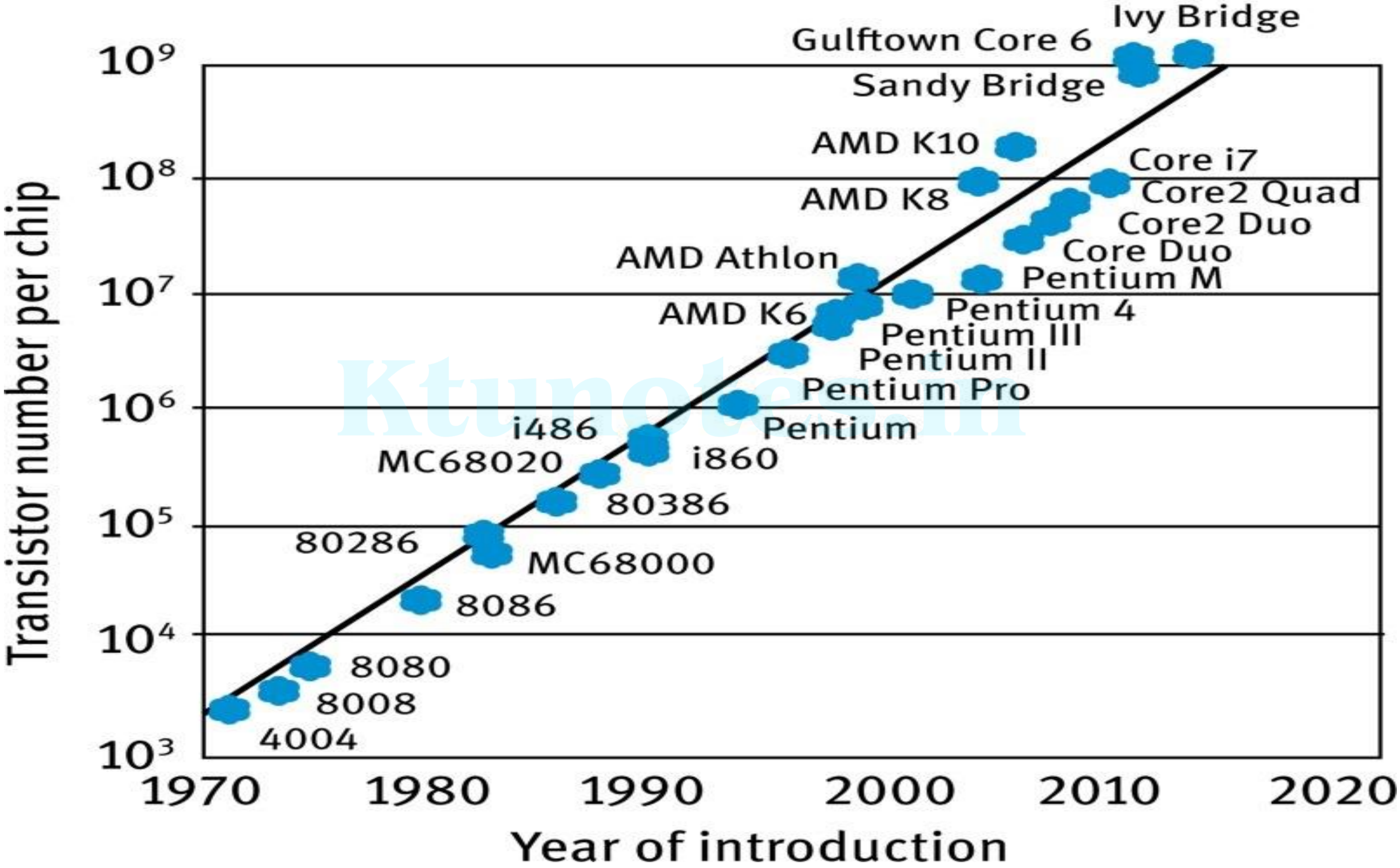
Introduction

- Various semiconductor materials are Silicon, Germanium and Gallium Arsenide etc.
- Among these, Si is chosen as substrate for 95% of the Semi-conductor manufacturing.

Moore's Law

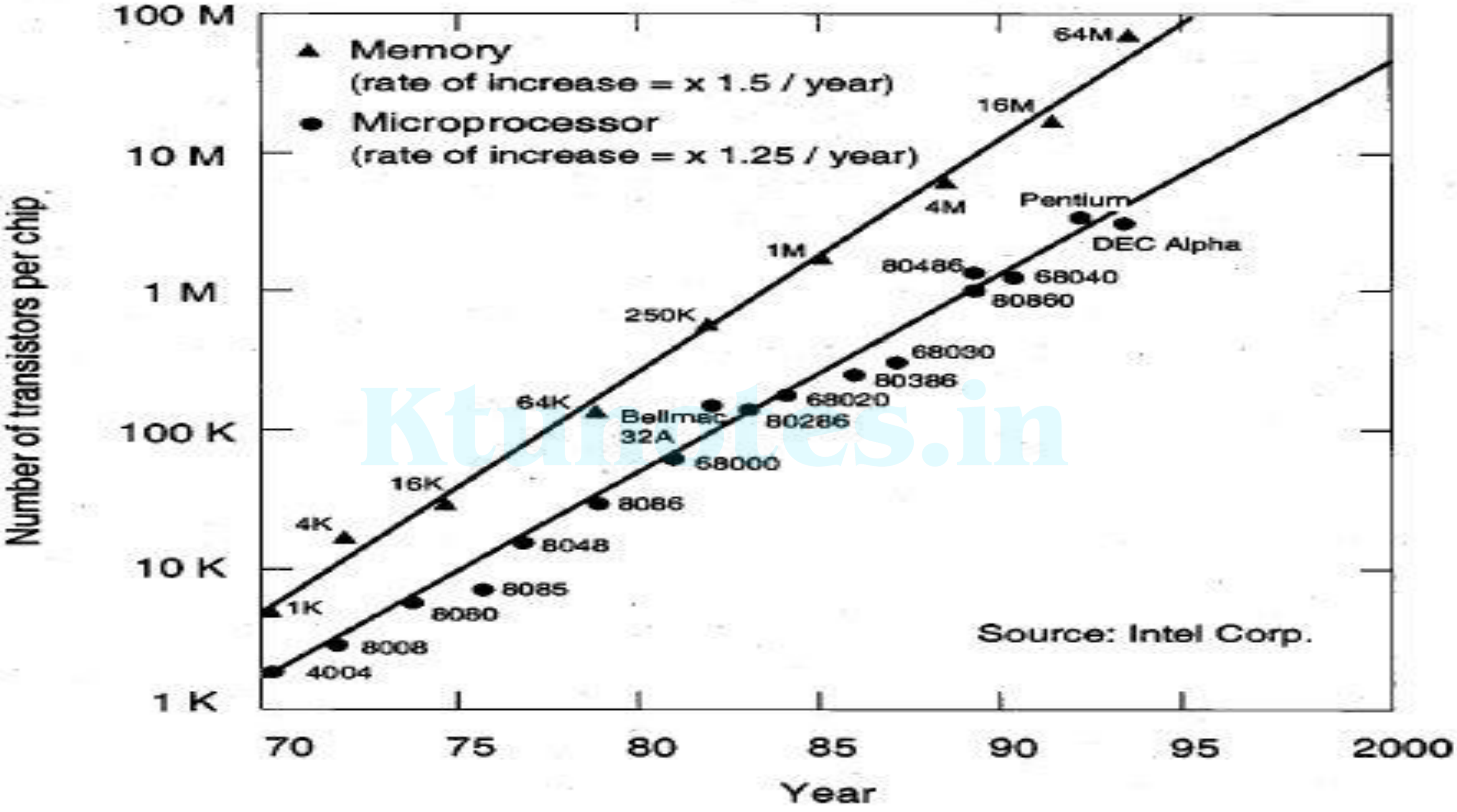
- In early 1960’s, Gordon Moore observed that plotting the number of transistors that can be most economically manufactured on a chip gives a straight line on a semi-logarithmic scale.
- At that time, he observed that transistor count doubles every 18 months. This observation has been called Moore’s Law.
- Moore’s Second Law:
  - *The cost of a semi-conductor chip fabrication plant doubles every four years.*

Moore's Law



Ivy Bridge
Gulftown Core 6
Sandy Bridge
AMD K10
AMD K8
Core i7
Core2 Quad
Core2 Duo
Core Duo
Pentium M
AMD Athlon
Pentium 4
AMD K6
Pentium III
Pentium II
Pentium Pro
Pentium
i486
i860
MC68020
80386
80286
MC68000
8086
8080
8008
4004
10^9
10^8
10^7
10^6
10^5
10^4
10^3
Transistor number per chip
1970
1980
1990
2000
2010
2020
Year of introduction

Moore's Law



Level of integration versus time for memory chips and logic chips.

ASIC

- ASIC [“a-sick”] is an acronym for ***Application Specific Integrated Circuit.***
- As the name indicates, ASIC is a non-standard Integrated circuit that is designed for a specific use or application.
- Generally an ASIC design will be undertaken for a product that will have a large production run , and the ASIC may contain a very large part of the electronics needed on a single integrated circuit.

ASIC

- Examples of ASIC IC’s are: a chip for a toy bear that talks; a chip for a satellite; a chip designed to handle the interface between memory and a microprocessor for a workstation CPU; and a chip containing a microprocessor as a cell together with other logic.

ASIC

- Examples of ASIC IC’s are: a chip for a toy bear that talks; a chip for a satellite; a chip designed to handle the interface between memory and a microprocessor for a workstation CPU; and a chip containing a microprocessor as a cell together with other logic.

- Two ICs that might or might not be considered as ASICs are, a controller chip for a PC and a chip for a modem. Both of these examples are specific to an application (shades of an ASIC) but are sold to many different system vendors (shades of a standard part). ASICs such as these are sometimes called application-specific standard products (ASSPs).

Advantages of ASIC

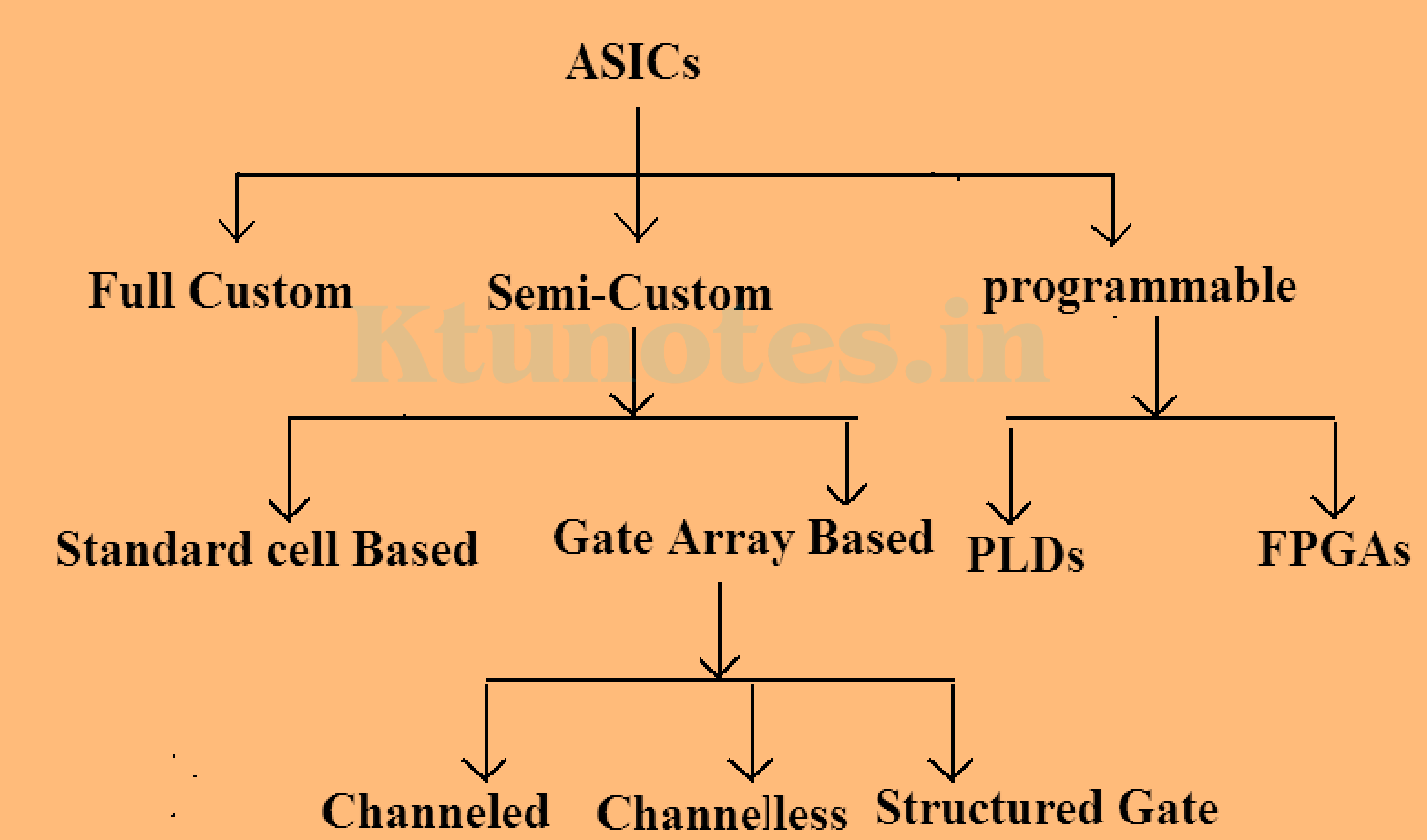
- The small size of ASIC.
- As a large number of circuits are built over a single chip, this causes high speed of application.
- ASIC has low power consumption.
- ASIC has no timing issues and post-production configuration.

Disadvantages of ASIC

- As these are customized chips they provide low level flexibility for programming.
- As the chips are designed from root level, they are of high cost per unit.
- ASIC have larger time to market.

Types of ASIC

- The classification of ASICs is shown below:



Types of ASIC

- So, as shown in the slide the ASICs are broadly classified into three types.

  I. Full-Custom ASICs

  II. Semi-custom ASICs

  III. Programmable ASICs

Full-Custom ASICs

- A Full custom ASIC is one which includes some (possibly all) logic cells that are customized and all mask layers that are customized.

- A microprocessor is an example of a full-custom IC . Designers spend many hours squeezing the most out of every last square micron of microprocessor chip space by hand.

- Customizing all of the IC features in this way allows designers to include analog circuits, optimized memory cells, or mechanical structures on an IC, for example. Full-custom ICs are the most expensive to manufacture and to design.

Full-Custom ASICs

- The manufacturing lead time (the time required just to make an IC not including design time) is typically eight weeks for a full-custom IC.
- These specialized full-custom ICs are often intended for a specific application so, we might call some of them as full-custom ASICs.

Full-Custom ASICs

- In a full-custom ASIC an engineer designs some or all of the logic cells, circuits, or layout specifically for one ASIC. This means the designer avoids using pretested and pre characterized cells for all or part of that design.
- This might be because existing cell libraries are not fast enough, or the logic cells are not small enough or consume too much power.

Full-Custom ASICs

- One has to use full-custom design if the ASIC technology is new or so specialized that there are no existing cell libraries or because the ASIC is so specialized that some circuits must be custom designed.
- Fewer and fewer full-custom ICs are being designed because of the problems with these special parts of the ASIC.
- The growing member of this family, now a days is the mixed analog/digital ASIC,

Features of Full-Custom ASICs

- Maximum performance.
- Smaller area.
- Offers highest speed.
- Increased design time.
- Higher design cost.
- Higher risk.
- Requires highly skilled designers.
- Mainly used in memory chips, high performance microprocessor chip.

Semicustom ASICs

- ASICs, for which all of the logic cells are predesigned and some (possibly all) of the mask layers are customized are called Semi-Custom ASICs.
- Using the predesigned cells from a cell library makes the design , much easier.
- There are two types of semicustom ASICs based on the type of logic cells taken from the library and amount of customization allowed for interconnects, they are:

  (i) Standard-cell–based ASICs

  (ii)Gate-array–based ASICs.

Standard-Cell Based ASICs

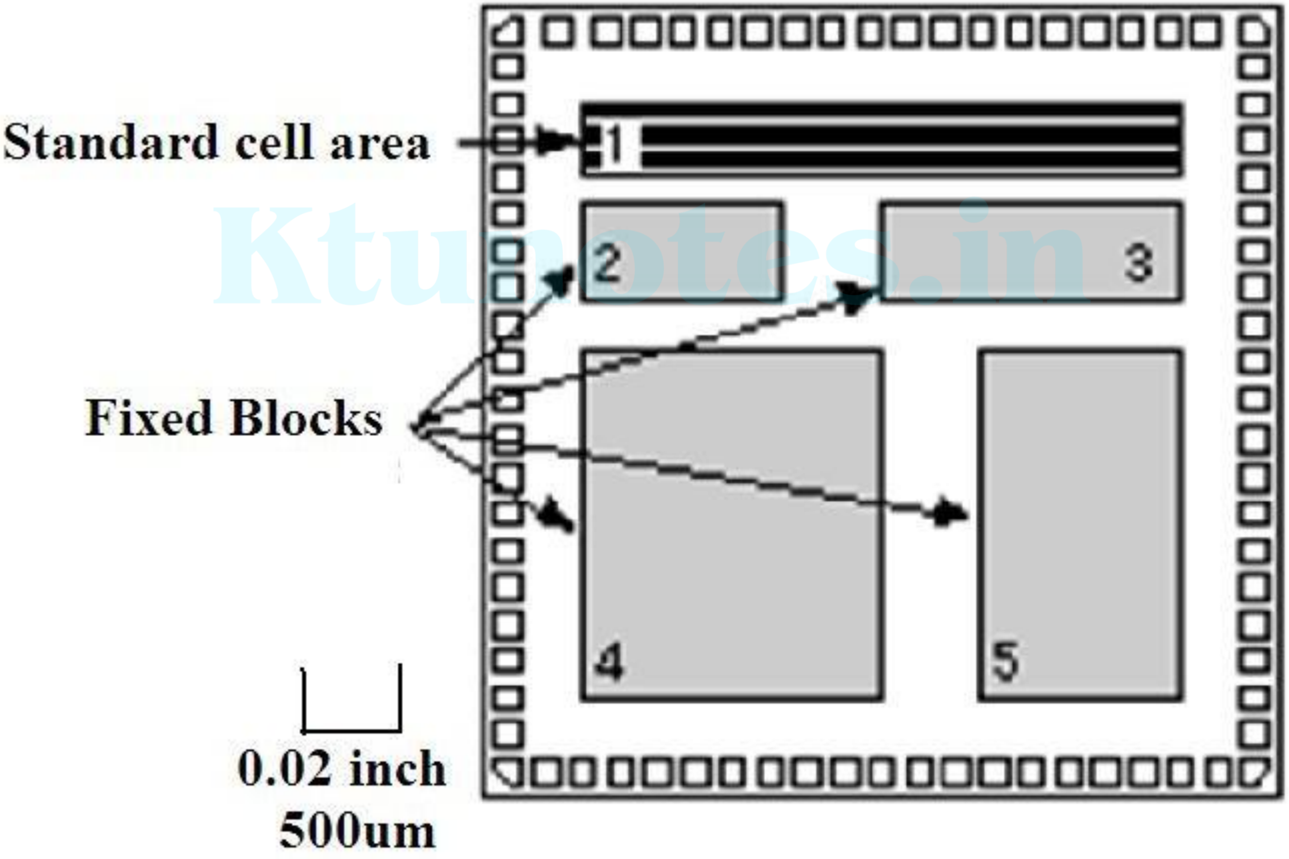
- A cell-based ASIC (cell-based IC, or CBIC pronounced sea-bick) uses predesigned logic cells (AND gates, OR gates, multiplexers, and flip-flops, for example) known as standard cells. They are stored in the form of library. This collection is known as standard cell library.
- One can apply the term CBIC to any IC that uses cells, but it is generally accepted that a cell-based ASIC or CBIC means a standard-cell based ASIC.

Standard-Cell Based ASICs

- The standard-cell areas (also called flexible blocks) in a CBIC are built of rows of standard cells like a wall built of bricks.
- The standard-cell areas may be used in combination with microcontrollers or even microprocessors, known as mega cells.
- Mega cells are also called mega functions, full-custom blocks, system-level macros (SLMs), fixed blocks, cores, or Functional Standard Blocks (FSBs).

Standard-Cell Based ASICs

- A cell-based ASIC (CBIC) die with a single standard cell-area (a flexible block) together with four fixed blocks.



Standard-Cell Based ASICs

- The ASIC designer defines only the placement of the standard cells and the interconnect in a CBIC.
- However, the standard cells can be placed anywhere on the silicon; this means that all the mask layers of a CBIC are customized and are unique to a particular customer.

Standard-Cell Based ASICs

- The advantage of CBICs is that designers save time, money, and reduce risk by using a predesigned, pretested, and pre characterized standard-cell library.
- In addition each standard cell can be optimized individually. During the design of the cell library each and every transistor in every standard cell can be chosen to maximize speed or minimize area .

Standard-Cell Based ASICs

- The disadvantages are the time or expense of designing or buying the standard-cell library and the time needed to fabricate all layers of the ASIC for each new design.

Gate-Array Based ASICs

- In a gate array (sometimes abbreviated GA) or gate-array based ASIC the transistors are predefined on the silicon wafer.
- The predefined pattern of transistors on a gate array is the base array , and the smallest element that is replicated to make the base array is the base cell (sometimes called a primitive cell).
- Only the top few layers of metal, which define the interconnect between transistors, are defined by the designer using custom masks. To distinguish this type of gate array from other types of gate array, it is often called a masked gate array (MGA).

Gate-Array Based ASICs

- The designer chooses from a gate-array library of predesigned and pre-characterized logic cells.
- The logic cells in a gate-array library are often called macros . The reason for this is that the base-cell layout is the same for each logic cell, and only the interconnect (inside cells and between cells) is customized, which is similar to a software macro.

Types of MGA or Gate-array based ASICs

- There are three types of Gate Array based ASICs.
  - Channeled gate arrays.
  - Channelless gate arrays.
  - Structured gate arrays.

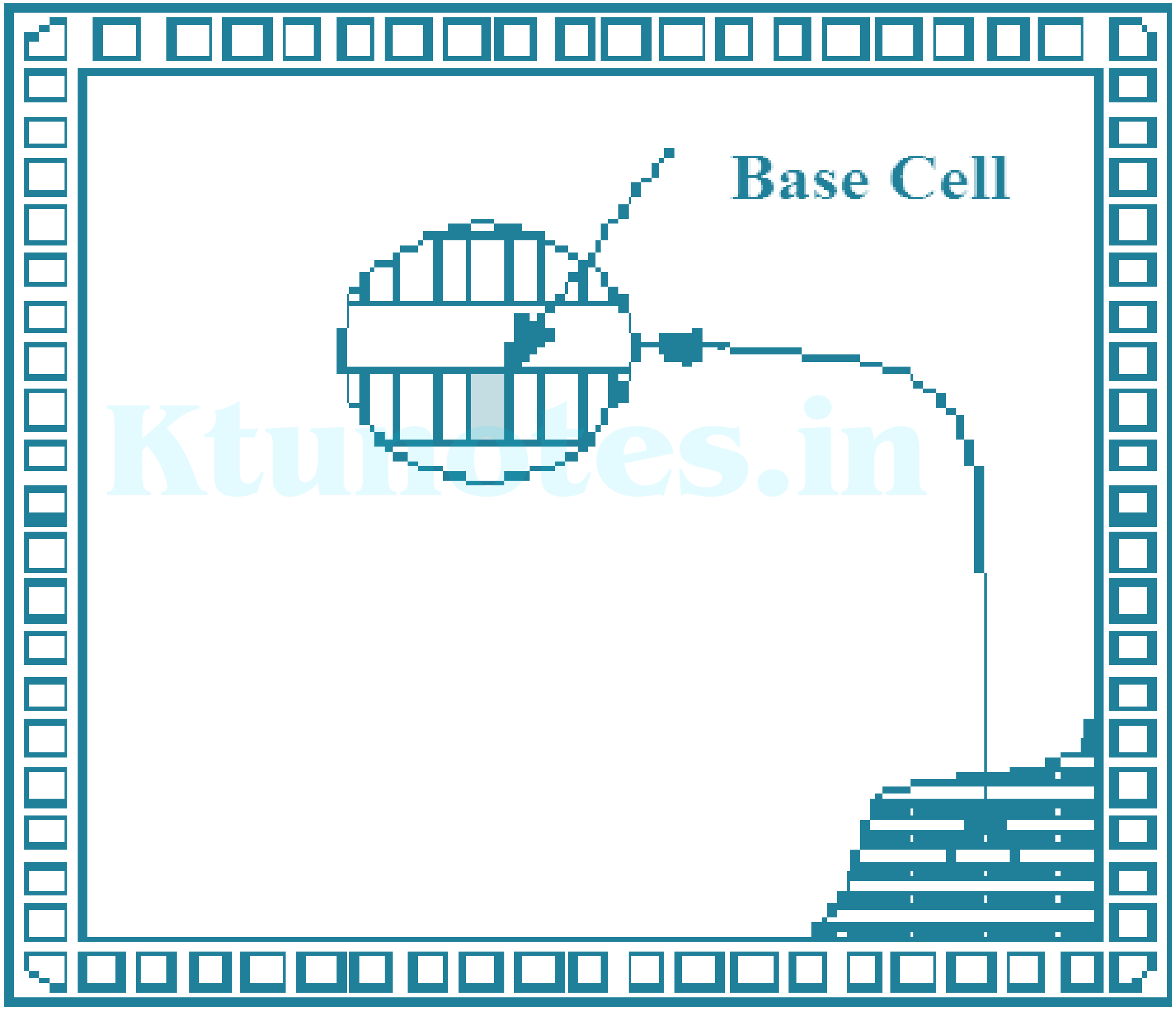
Channeled gate arrays

- The channeled gate array was the first to be developed .
- In a channeled gate array space is left between the rows of transistors for wiring.
- A channeled gate array is similar to a CBIC.
- Both use the rows of cells separated by channels used for interconnect.
- One difference is that the space for interconnect between rows of cells are fixed in height in a channeled gate array, whereas the space between rows of cells may be adjusted in a CBIC.

Channeled gate arrays

- The channeled gate array was the first to be developed .
- In a channeled gate array space is left between the rows of transistors for wiring.
- A channeled gate array is similar to a CBIC.
- Both use the rows of cells separated by channels used for interconnect.
- One difference is that the space for interconnect between rows of cells are fixed in height in a channeled gate array, whereas the space between rows of cells may be adjusted in a CBIC.

A channeled gate-array die



Features of MGA

- Only the interconnect is customized.
- The interconnect uses predefined spaces between rows of base cells.
- Manufacturing lead time is between two days and two weeks.

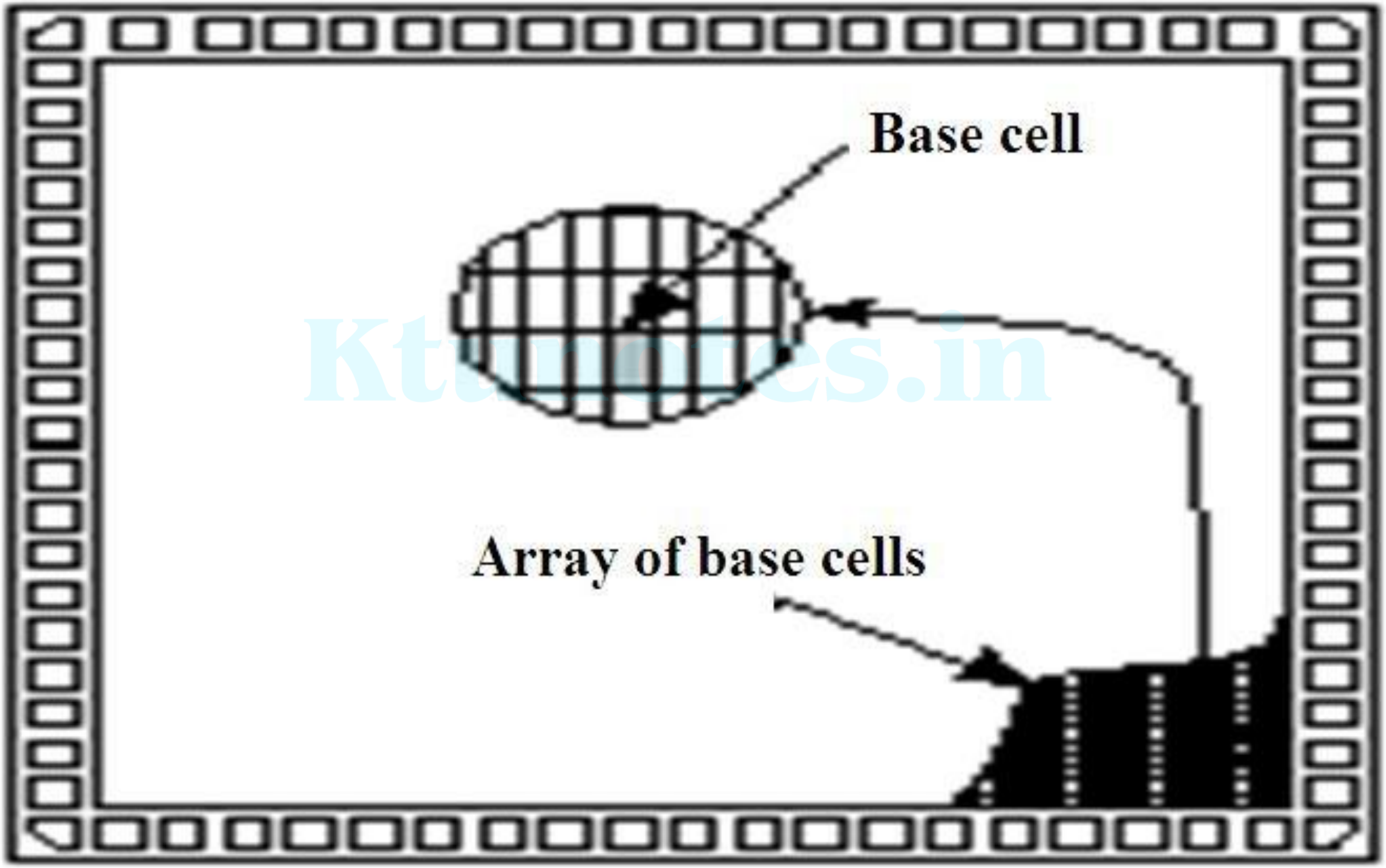
Channelless Gate Array

- The channelless gate-array architecture is now more widely used .
- The routing on a channelless gate array uses rows of unused transistors.
- The key difference between a channelless gate array and channeled gate array is that there are no predefined areas set aside for routing between cells on a channelless gate array.
- Instead we route over the top of the gate-array devices.
- We can do this because we customize the contact layer that defines the connections between metal 1, the first layer of metal, and the transistors.

Features of Channelless Gate Array

- Only some (the top few) mask layers are customised.
- Manufacturing lead time is around two days to two weeks.
- When we use an area of transistors for routing in a channel less array, we do not make any contacts to the devices lying underneath; we simply leave the transistors unused.

A channel less gate-array or sea-of-gates (SOG) array die.



Channelless Gate Array

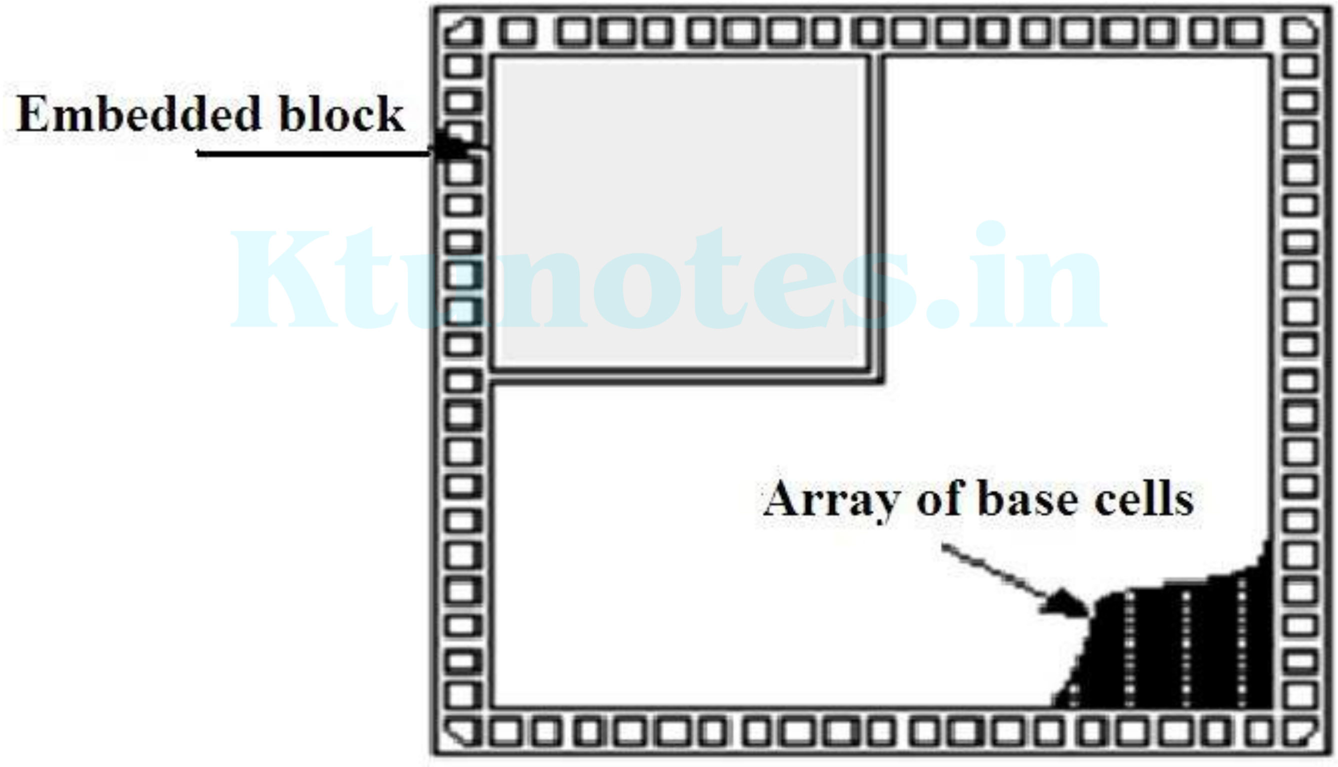
- The logic density (the amount of logic that can be implemented in a given silicon area) is higher for channel less gate arrays than for channeled gate arrays.
- This is usually attributed to the difference in structure between the two types of array.
- In fact, the difference occurs because the contact mask is customized in a channel less gate array, but is not usually customized in a channeled gate array.

Channelless Gate Array

- This leads to denser cells in the channel less architectures.
- Customizing the contact layer in a channel less gate array allows us to increase the density of gate-array cells because we can route over the top of unused contact sites.

Structured Gate Array

- A structured or embedded gate-array die showing an embedded block in the upper left corner.



Structured Gate Array

- A structured gate array or embedded gate array (also known as master-slice or master-image) combines some of the features of CBICs and MGAs.
- This makes the implementation of memory, for

Features of Structured Gate Array

- Only the interconnect is customized.
- Custom Blocks(same for each design can be embedded).
- Manufacturing lead time is between two days and two weeks.
- An embedded gate array gives the improved area efficiency and increased performance of a CBIC but with the lower cost and faster turn around of an MGA.

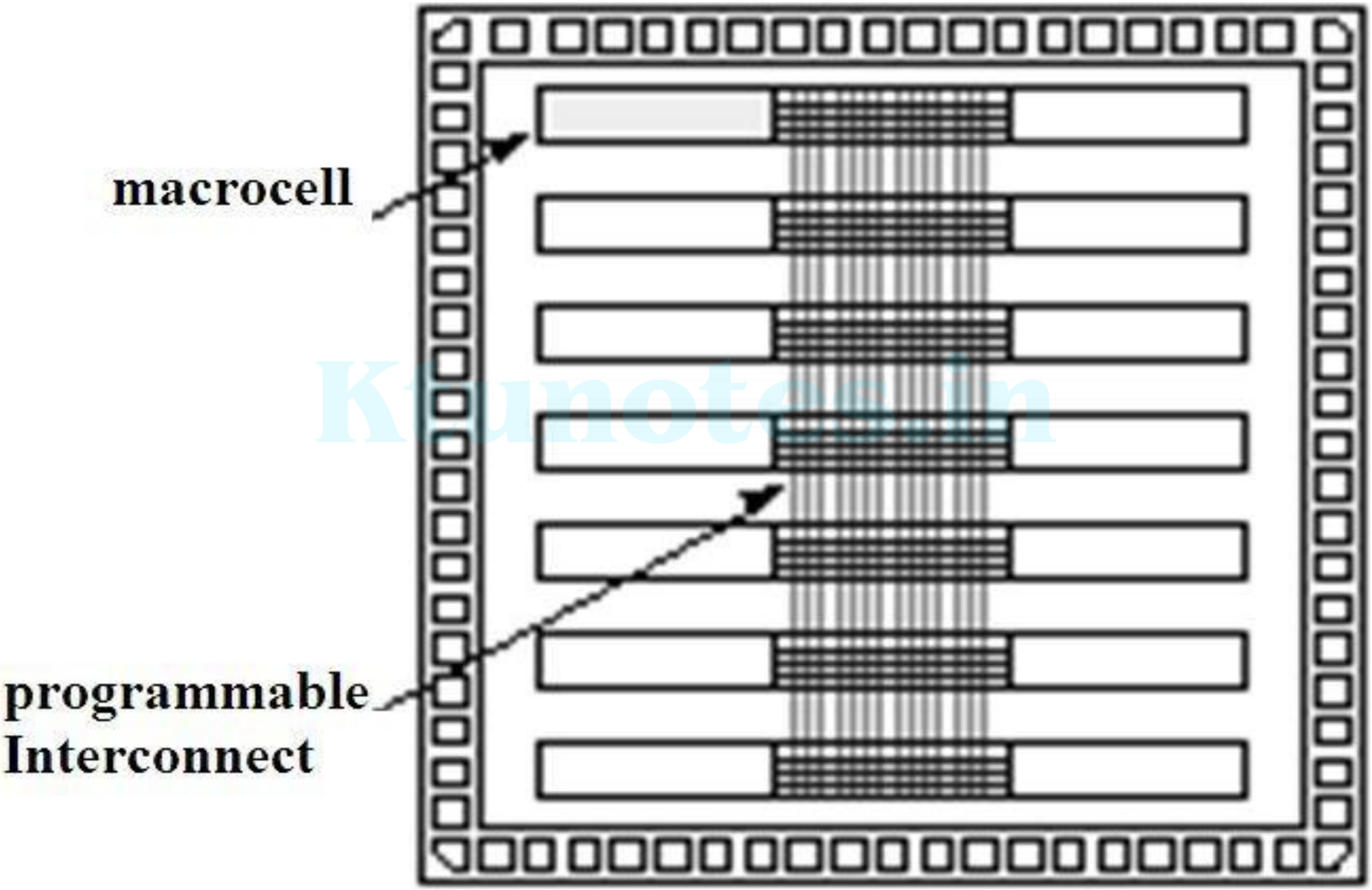
Disadvantages of Structured Gate Array

- The disadvantage of an embedded gate array is that the embedded function is fixed.
- For example, if an embedded gate array contains an area set aside for a 32 k-bit memory, but we only need a 16 k-bit memory, then we may have to waste half of the embedded memory function.
- However, this may still be more efficient and cheaper than implementing a 32 k-bit memory using macros on a SOG array.

Programmable Logic Devices

- Programmable logic devices (PLDs) are standard ICs that are available in standard configurations.
- However, PLDs may be configured or programmed to create a part customized to a specific application, and so they also belong to the family of ASICs.
- PLDs use different technologies to allow programming of the device.

A programmable logic device (PLD) die.



Features of PLDs

- No customized mask layers or logic cells.
- Fast design turnaround.
- A single large block of programmable interconnect
- A matrix of logic macro cells that usually consist of programmable array logic followed by a flip-flop or latch.

Programmable Logic Devices

- The simplest type of programmable IC is a read-only memory(ROM). The most common types of ROM use a metal fuse that can be blown permanently (a programmable ROM or PROM).
- An electrically programmable ROM, or EPROM, uses programmable MOS transistors whose characteristics are altered by applying a high voltage.
- One can erase an EPROM either by using another high voltage (an electrically erasable PROM, or EEPROM) or by exposing the device to ultraviolet light (UV-erasable PROM, or UVPROM).

Programmable Logic Devices

- There is another type of ROM that can be placed on any ASIC; a mask-programmable ROM (mask-programmed ROM or masked ROM).
- A masked ROM is a regular array of transistors permanently programmed using custom mask patterns.
- So, an embedded masked ROM is a large, specialized, logic cell.

Programmable Logic Devices

- A Programmable Array Logic (PAL) can also include registers (flip-flops) to store the current state information so that you can use a PAL to make a complete state machine.
- We can place a logic array as a cell on a custom ASIC.
- This type of logic array is called a programmable logic array (PLA).
- There is difference between a PAL and a PLA: a PLA has a programmable AND logic array, or AND plane, followed by a programmable OR logic array, or OR plane; a PAL has a programmable AND plane and a fixed OR plane.

Field-Programmable Gate Arrays(FPGAs)

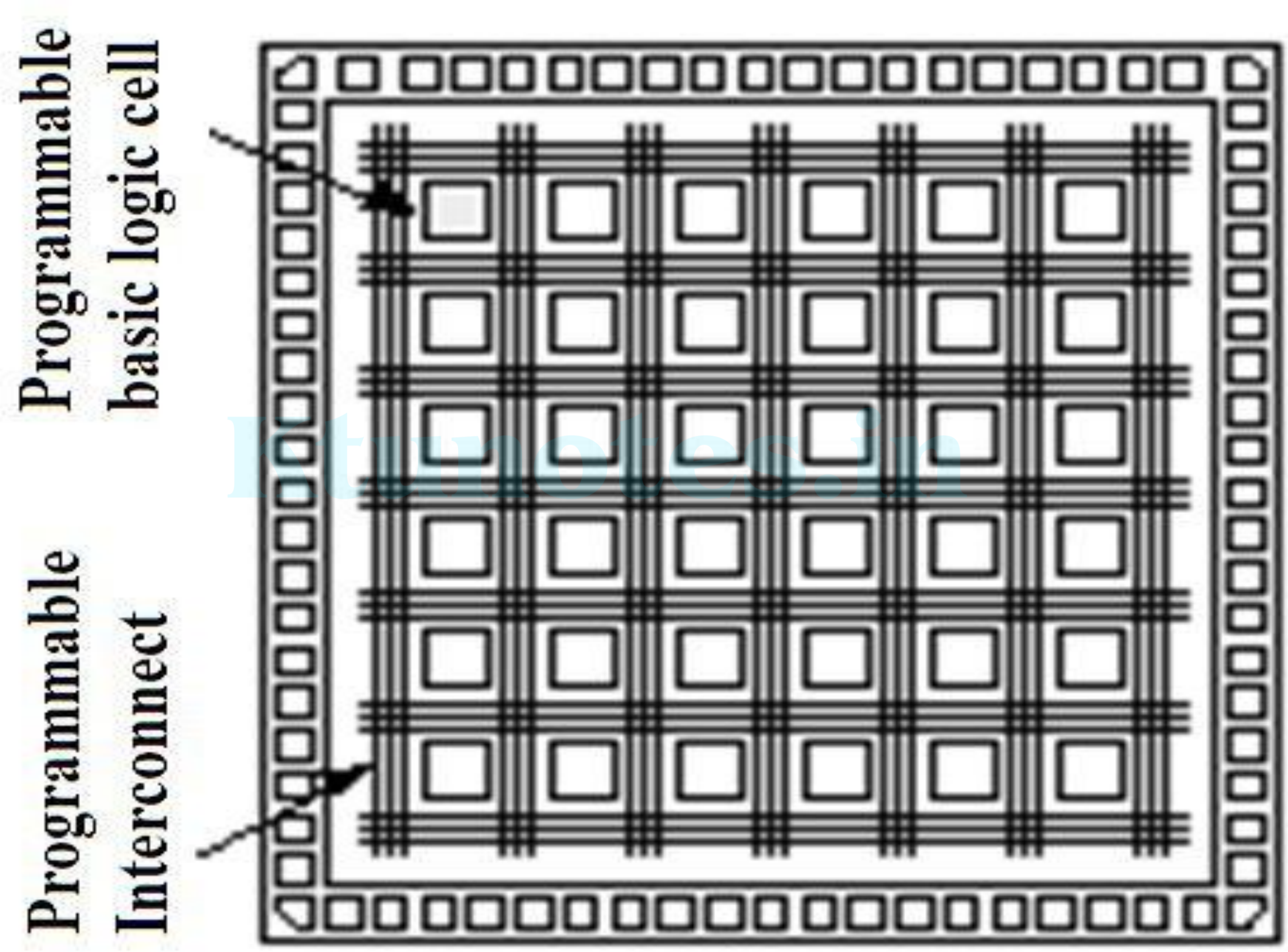
- FPGAs are the newest member of the ASIC family and are rapidly growing in , replacing TTL in microelectronic systems.
- Even though an FPGA is a type of gate array, we do not consider the term gate-array based ASICs to include FPGAs.

- There is very little difference between an FPGA and a PLD.
- An FPGA is usually just larger and more complex than a PLD.
- In fact, some vendors that manufacture programmable ASICs call their products as FPGAs and some call them as complex PLDs (CPLDs) .

Characteristics of an FPGA

- None of the mask layers are customized.
- There is a method for programming the basic logic cells and the interconnect.
- The core is a regular array of programmable basic logic cells that can implement combinational as well as sequential logic (flip-flops).
- A matrix of programmable interconnect surrounds the basic logic cells.
- Programmable I/O cells surround the core.
- Design turnaround is a few hours.

Field-programmable gate array (FPGA) die.



Field-Programmable Gate Arrays(FPGAs)

- The architecture consists of configurable logic blocks, configurable I/O blocks, and programmable interconnects.
- Also, there will be clock circuitry for driving the clock signals to each logic block, and additional logic resources such as ALUs, memory, and decoders may be available.
- The two basic types of programmable elements for an FPGA are Static RAM and anti-fuses.

FPGA – Configurable Logic Block (CLB)

- It is a regular array of basic logic cells that can implement combinational as well as sequential logic.
- CLB contain several modules such as look-up table (LUT), multiplexers, gates and flip-flops.
- LUT is a hardware that stores the truth table of a function in SRAM to function as a combinational circuit.
- SRAM can be reprogrammed to implement different functions.
- Typically CLD have 4-6 input variables.
- Larger designs are partitioned and mapped to a number of CLBs with each CLB programmed to perform a particular function.

FPGA – Input – Output Block (IOB)

- Programmable I/O buffers can be configured either as input or output.
- It provides programmable interface between CLB and device's external package pins.

FPGA – Programmable Interconnect (PI)

- Programmable Interconnect consists of horizontal and vertical routing channels that can be connected together if needed.
- Connection boxes are used to connect input and output pins of CLB to the channels.
- It consists of both wire segment and programmable switches.
- Each component has six switches, which are controlled by SRAM.
- Depending on the bit stored on SRAM, the connection is established between the horizontal and vertical interconnect wires.

FPGA – Advantages

- It can be programmed at logic level.
- It can implement faster and parallel processing of signals.
- FPGAs are programmable at software level at any to the time.
- Available faster to the market.
- Software takes care of routing, placement and timing. This makes lesser manual intervention.

FPGA – Disadvantages

- Programming requires knowledge of HDL programming languages as well as digital system fundamentals.
- More power consumptions.
- Slightly costly.

FPGA – Applications

- Testing and measurement.
- Aerospace and defence.
- Audio- video broadcasting.
- Medical imaging.
- Communication.
- Consumer applications.

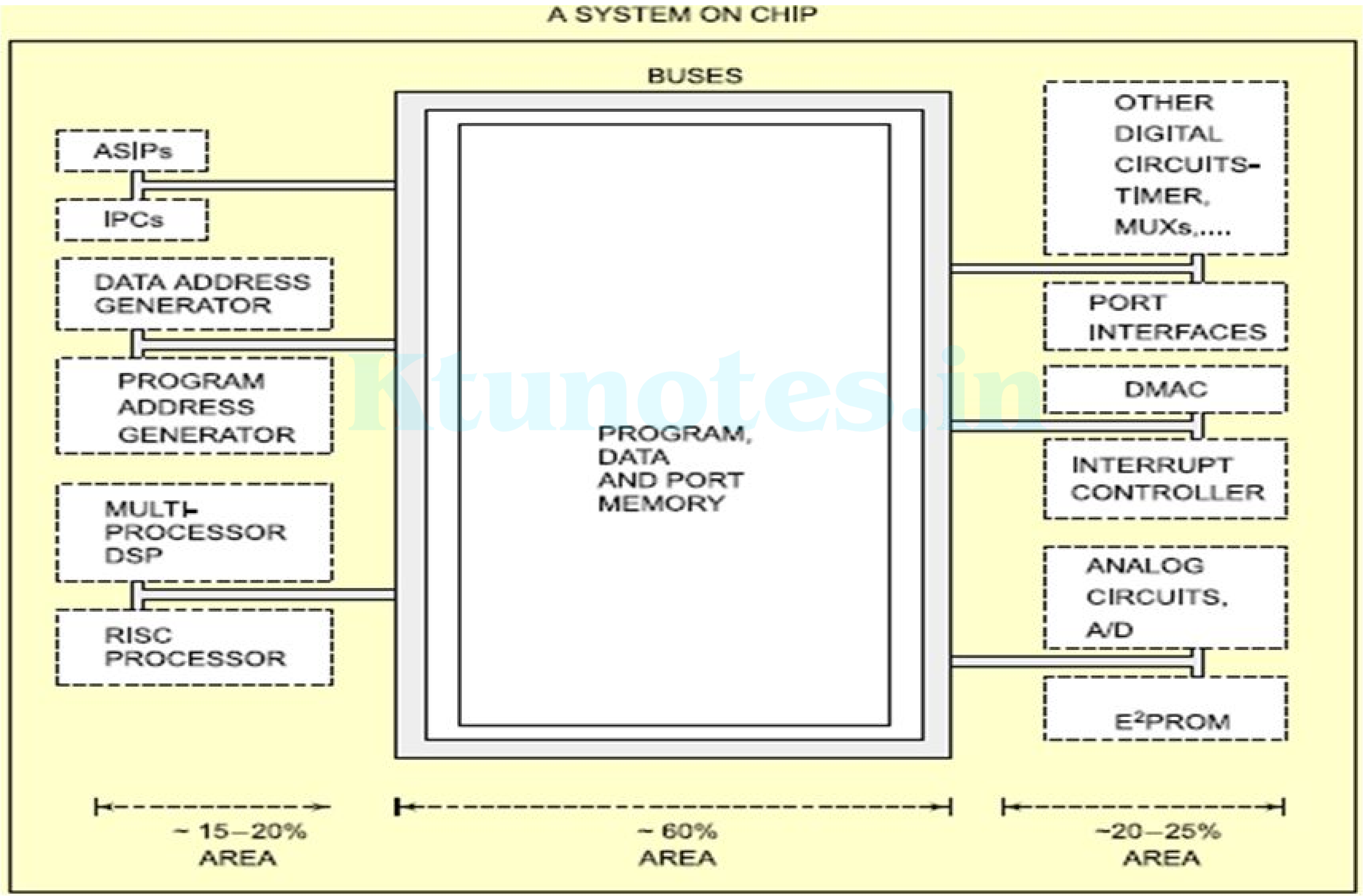
FPGA Vs ASIC

FPGA	ASIC
Field Programmable Gate Array	Application Specific Integrated Circuit
Purchased from vendor as a standard part, then programmed by the user.	Made to customer specification by the vendor.
No product set-up cost.	High production set-up cost.
Fast turnaround time (can be programmed in minutes).	Slow turnaround time (often at least 6 weeks)
Relatively high cost per unit and low capability per chip.	Lower cost per unit. Good for high volume production.
Design requires mostly writing HDL code.	Design often requires knowledge of physical layout of silicon inside the IC.
Reprogrammable.	Not reprogrammable.
Less energy efficient , requires more power.	Energy efficient, requires less power
Highly suitable for applications where the circuit has to be upgraded time to time.	Permanent circuiting that can’t be upgraded.

CPLDs Vs FPGAs

	CPLD	**FPGA**
Architecture	PAL-like	Gate-Array-like
Density	Low to medium	Medium to high
Speed	Fast, predictable	Application dependent
Interconnect	Crossbar	Routing
Power Consumption	High	Medium

System on Chip (SoC)



System on Chip (SoC)

- SoC is an Integrated Circuit, that integrates an entire electronic system onto it.
- The components includes a central processing unit, input and output ports, internal memory, analog circuits, on-chip interconnections, ASIC logics and software.
- Instead of a system that assembles several chips and components onto a circuit board, SoC fabricates all necessary circuits into one unit.

System on Chip (SoC)

- Depending on the kind of system that has been reduced to the size of a chip, it can perform a variety of functions including signal processing, wireless communication, artificial intelligence and more.

- ***CPU (Central Processing Unit) :-***
- Soc has a processor at it’s core which will define it’s functions.
- It can have more than one processor. i.e, Application Specific Instruction Processors (ASIPs), Intellectual Property (IP) Cores, Digital Signal Processors etc.

System on Chip (SoC)

- ***Memory:-***
- Memories allows the chip to store program and data.
- It may have RAM, ROM, EEPROM, flash memory etc.
- ***Connectivity module/ Interfaces:***
- SoC includes external interfaces for communication protocols.
- It includes wifi, USB, UART, Bluetooth etc.
- Also when needed it includes ADCs and DACs.

System on Chip (SoC)

- ***Advantages:***
- Low power.
- Low cost.
- Fast operation.
- Small size.
- High reliability.
- ***Disadvantages:***
- High fabrication cost.
- Difficulty in packaging.
- More heat dissipation.

ASIC Design Flow

- The sequence of steps to design an ASIC is known as the Design flow .
- The first step is defining the specifications of the product.
- It includes the top level functionality, computation algorithm, clock frequency, package type, power type, power supply, communication protocols etc.

ASIC Design Flow

1. Design entry :

- Entering the design into an ASIC design system, either using a hardware description language (HDL) or schematic entry.
- The schematic shows how all the components are connected together or even an HDL describes the circuit.

2. Logic Synthesis:

- Use an HDL (VHDL or Verilog) and a logic synthesis tool to produce a netlist.
- A netlist is a description of the logic cells and their connections.

ASIC Design Flow

3.System partinoning :

- Divide a large system into ASIC-sized pieces.
- More number of ASICs are used in case of large designs.

4. Pre-layout simulation:

- A simulation test is done to check whether the design contains any error or the design functions correctly.

ASIC Design Flow

5. Floor planning:

- Arrange the blocks of the netlist on the chip.
- Input to the floor planning step is the output of system partitioning and design entry (netlist).
- The entire area is divided into physical partitions and their shapes are modelled.

ASIC Design Flow

6. Placement:

- Decide the locations of cells in a block.
- The aim is to minimise the wire length, while ensuring optimal placement.

ASIC Design Flow

7. Routing:

- Make the connections between cells and blocks.
- It is split into two distinct steps, called ***global and local routing***.
- Global routing determines the interconnections between the logic cells and blocks.
- Local routing joins the logic cells with interconnection.

ASIC Design Flow

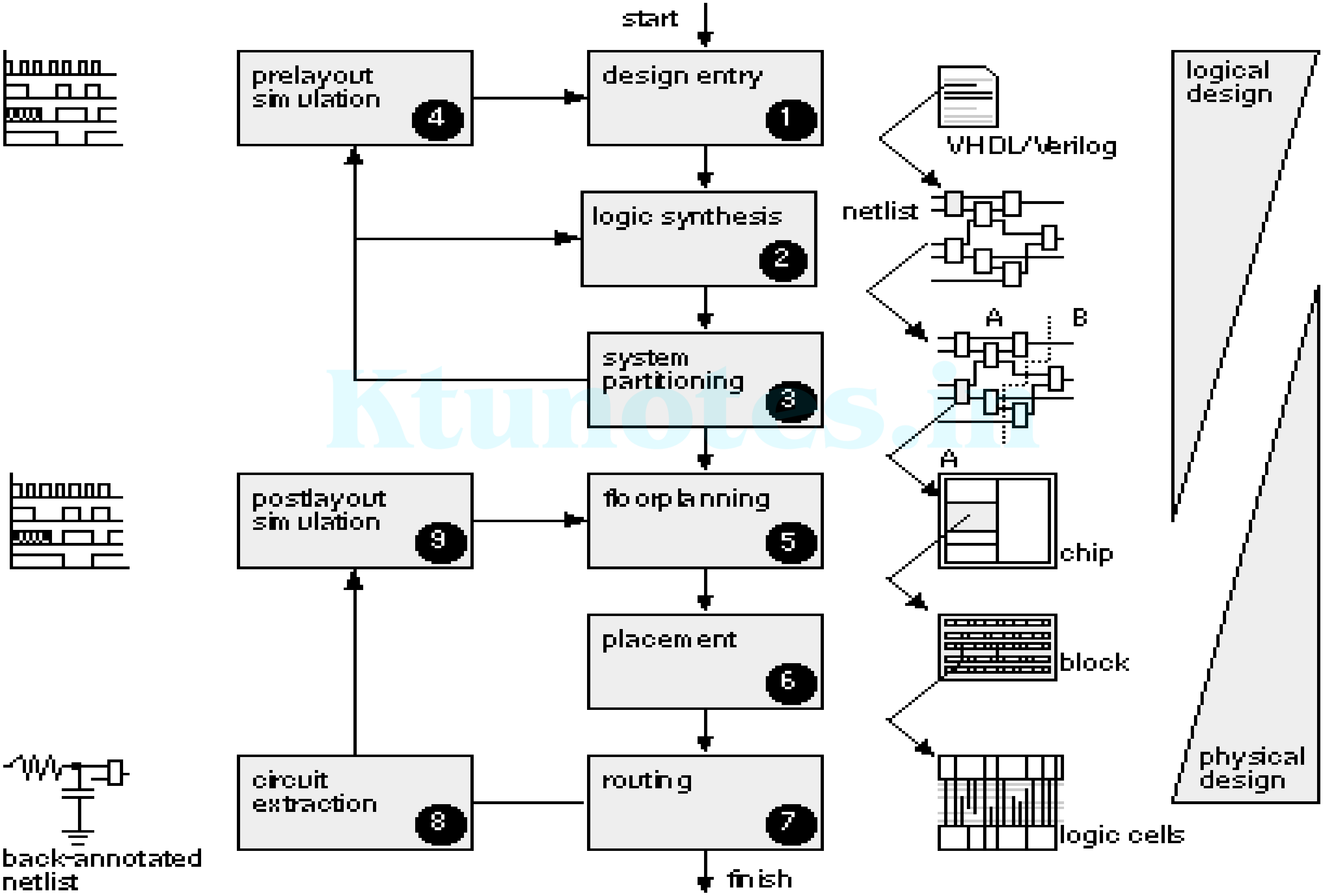
8. Extraction:

- Determine the electrical properties like resistance value and capacitance value of interconnect.

9.Post layout simulation:

- Before the submission of the model for manufacturing, simulation is done to check whether the system functions properly along with a load of interconnect.

ASIC design flow diagram



ASIC Design Flow

- In the flow diagram the steps from 1 to 4 are part of logical design ,and steps from 5 to 9 are part of physical design.
- When we are performing system partitioning we have to consider both logical and physical factors.

FPGA Design Flow

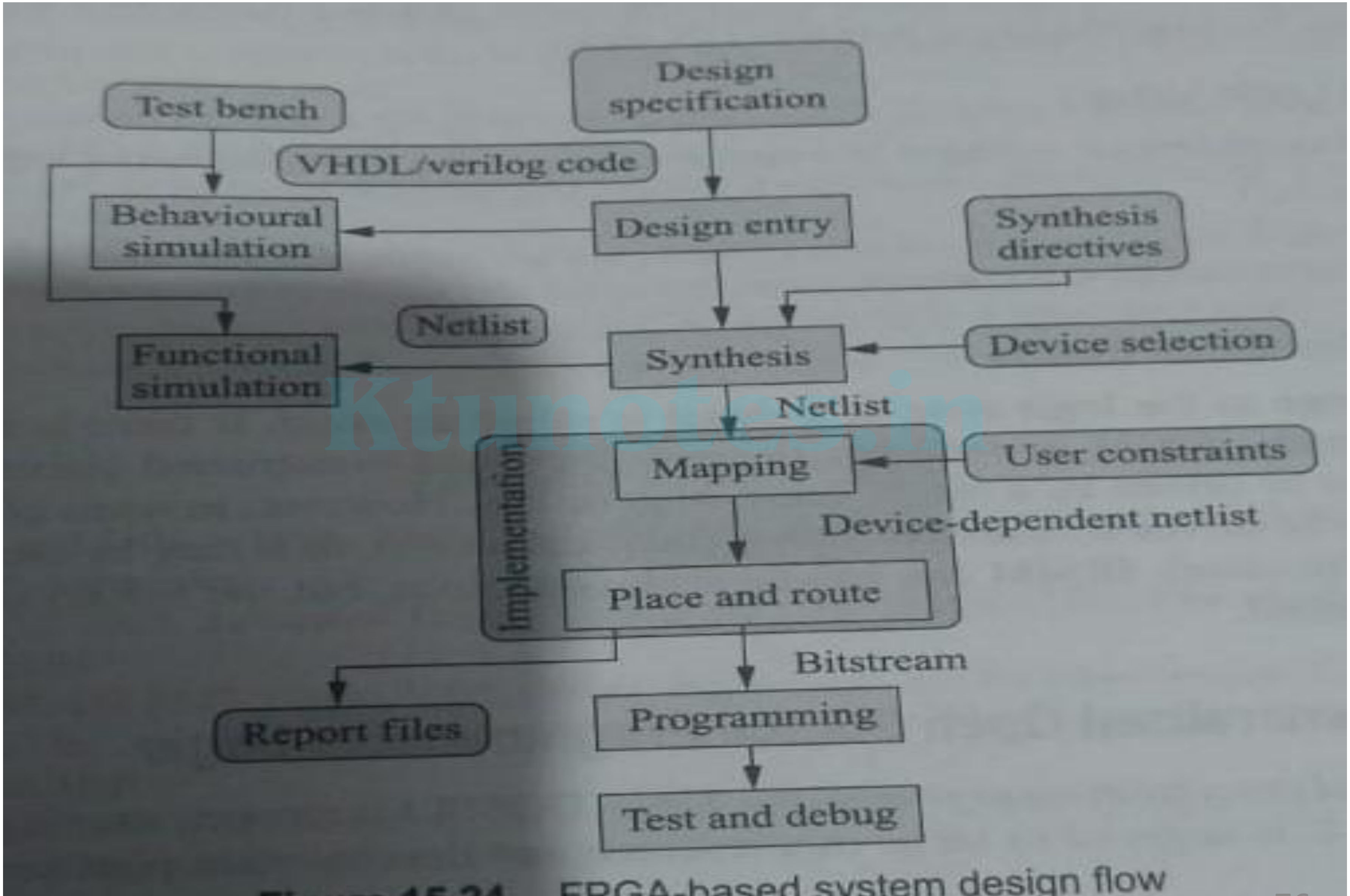


Figure 15.24 FPGA-based system design flow

FPGA Design Flow

- The design flow starts with the design specifications.
- The functional description of the system is written in a hardware description language (VHDL or Verilog) in the behavioural modelling style.
- The functionality is checked by performing behavioural simulation using a set of test vectors.
- The synthesis steps translates the behavioural netlist, the selected device family name and other synthesis directives.
- The gate level netlist is again checked for functionality.

FPGA Design Flow

- The user constrains are to be specified for timing , power etc.
- Then using user constrains and gate level netlist, the implementation step is performed.
- In the implementation step, the mapping of the logic gates are done to the available functional blocks in the FPGA and the placement and routing are done to complete the implementation.
- Next, the bitstream file is generated which contains the programmable data. It is downloaded through the JTAG cable onto the FPGA device. It is called FPGA programming.

FPGA Design Flow

- The final step is to test the FPGA device in the system and debug for any problem in functionality.

VLSI Design Methodology

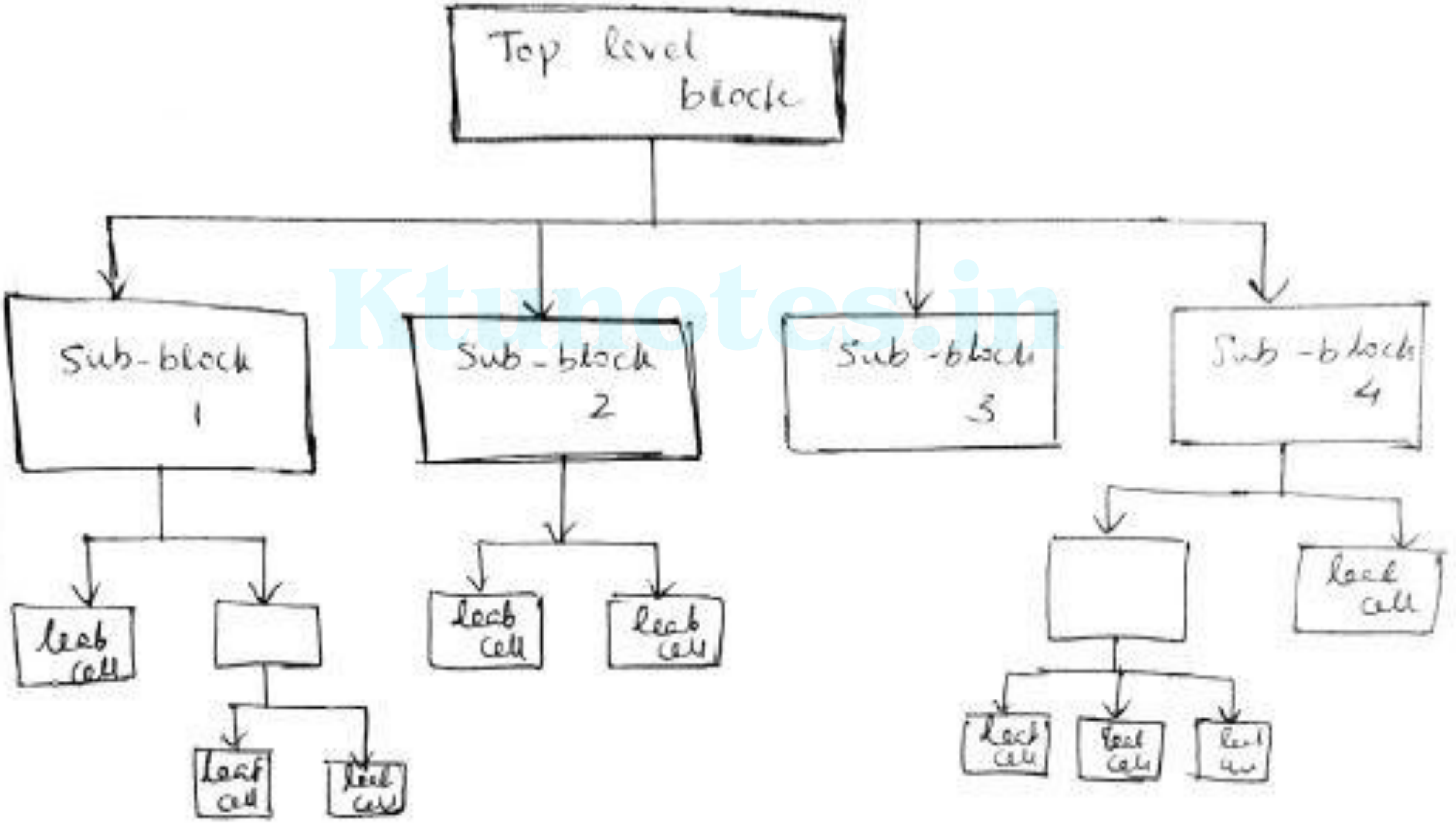
- VLSI design is a sequential process of generating the physical layout of an IC, starting from the specifications of that circuit.
- It can be fully or semi-automated using numerous software's called Electronic Design Automation (EDA) or Computer Aided Design (CAD) tools.
- The complexity of the VLSI circuits is usually large.
- The design complexity is handled using several approaches.
- There are two design styles used in VLSI design:
  - Top-down approach
  - Bottom-up approach

VLSI Design Methodology

- In top-down approach, the system is built starting from the top to the bottom.
- In bottom-up approach, the basic building blocks are built first, and they are combined to build the entire systems.
- Both approaches have their merits and demerits.
- ***<u>Top-down design methodology:</u>***
- In this method, we define the top-level blocks and identify the sub-blocks necessary to build the top-level blocks.
- The sub-blocks are further subdivided up to leaf cells.
- Leaf cells are the cells that cannot be sub divided further.

VLSI Design Methodology

- ***Top-down design methodology:***



VLSI Design Methodology

- ***Top-down design methodology:***

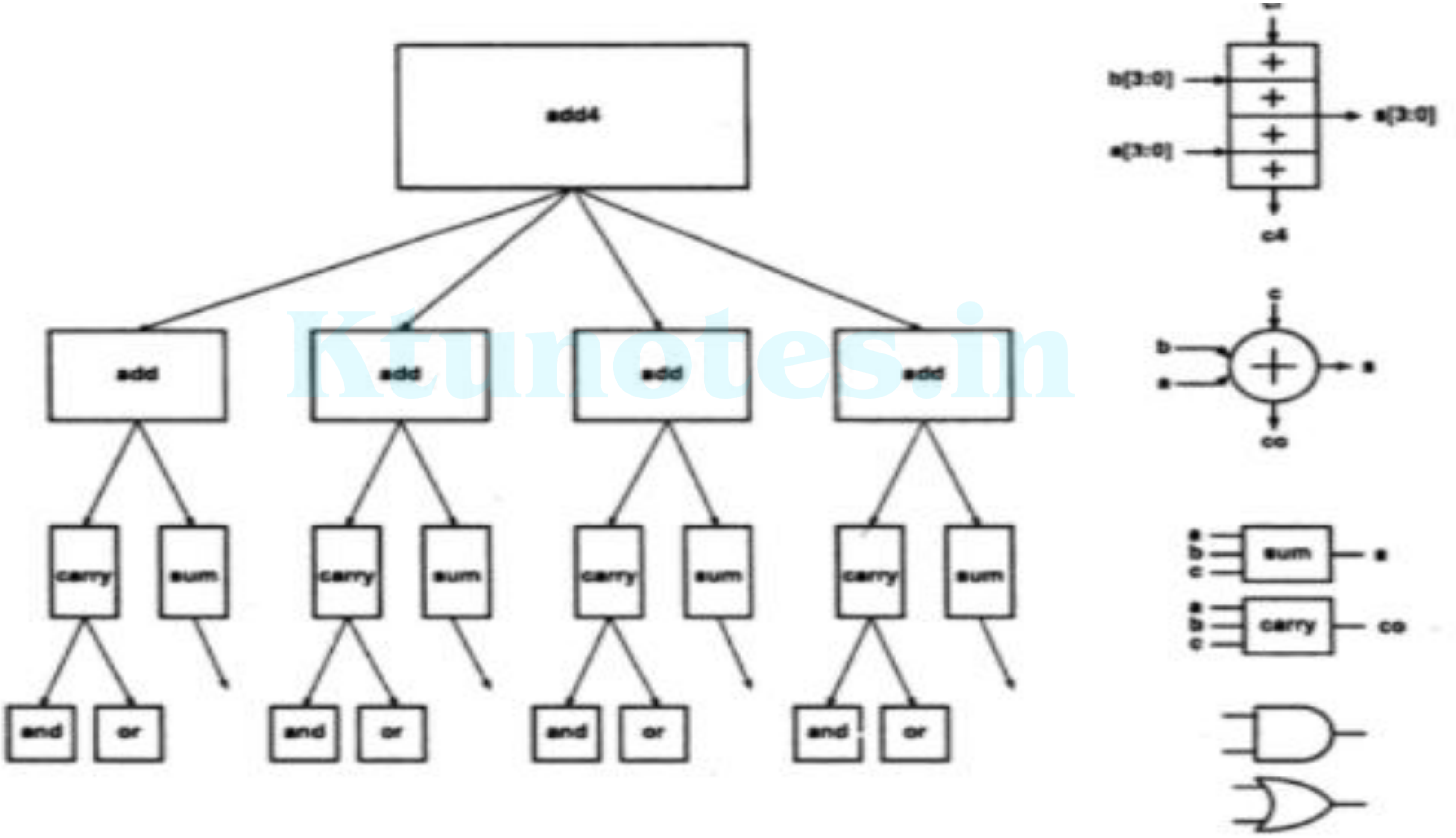


Figure 1.22 Structural decomposition of a 4-bit adder, showing the levels of hierarchy.

VLSI Design Methodology

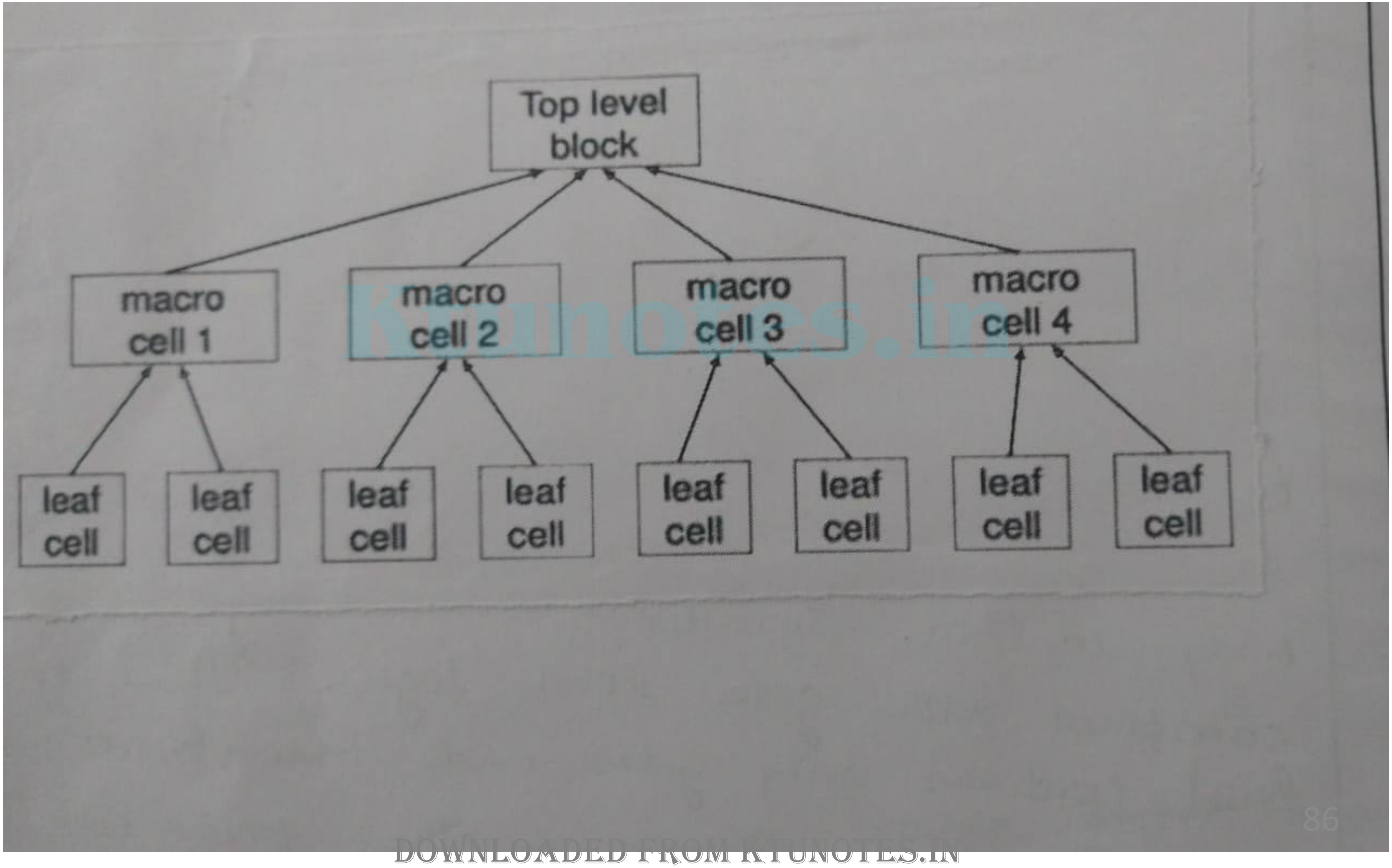
- ***Top-down design methodology:***
- The top-level block represents the entire chip.
- Each lower level blocks represent major functions of the chip.
- Intermediate blocks may contain smaller functionality blocks combined with gate level logic.
- The bottom level contain only gates and macro-functions.
- Top-down design allows more than one engineer to design the chip and simplify the design task.
- It also allows flexibility in design.

VLSI Design Methodology

- ***Bottom-up design methodology:***
- In this methodology, the designer first identify the building blocks, that are available.
- The bigger cells are obtained from these building blocks.
- These cells are then used for higher level blocks until we build the top-level blocks in the design.

VLSI Design Methodology

- ***Bottom-up design methodology:***



Concept of Regularity, Modularity & Locality

- The large complexity of VLSI design is handled by dividing the large system into several sub-modules.
- ***Regularity:***
- It indicates that the decomposition process must not produce a large number of blocks and the blocks need to be similar as much as possible.
- An array structure normally has a good regularity.
- Regularity avoids a number of different blocks to be designed and verified and can be maintained at all levels of abstraction.

Concept of Regularity, Modularity & Locality

- ***Modularity:***
- It means that the functional blocks must have well-defined interfaces and functionality.
- It allows each block can be designed independently from each other concurrently.
- It also enables design reuse.

Concept of Regularity, Modularity & Locality

- ***Locality:***
- The concept of locality ensures that connections are mostly between neighbouring modules, avoiding long-distance connection as much as possible.
- The decomposition should be such that the blocks, exchanging signals frequently must be close to each other.

Logical and physical design

- ***Logical Design :***
  - Design entry
  - Logic synthesis
  - System partitioning
  - Pre-layout simulation
- ***Physical Design:***
  - Floor planning
  - Placement
  - Routing
  - Circuit extraction
  - Post-layout simulation

Logical Design

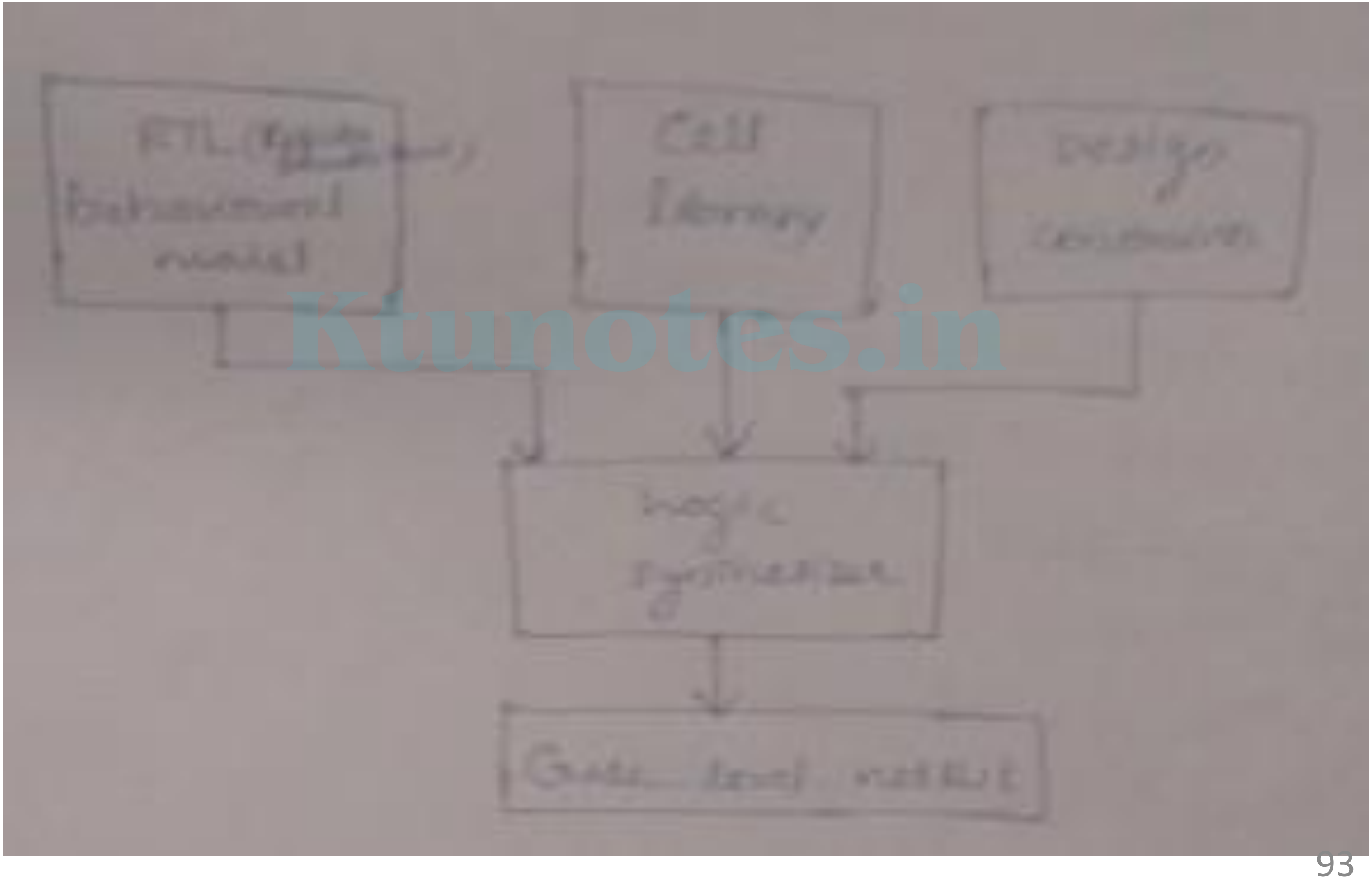
- ***<u>Design Entry & Logic Synthesis :</u>***
- The purpose of design entry is to describe a microelectronic system to a set of electronic design automation tools.
- Design entry usually consists of drawing a schematic which shows how all the components are connected together.
- Otherwise hardware description language can be used for the design entry.
- The output of a schematic entry tool is a binary version of the schematic that we call a netlist.

Logical Design

- ***Design Entry & Logic Synthesis :***
- Netlist contains a description of all the components in a design and their interconnections.
- Circuit schematics are drawn on schematic sheets.
- Components in an ASIC schematics are chosen from a library of cells.
- Most ASIC companies provides a schematic library of primitive gates to be used for schematic entry.

Logical Design

- ***Design Entry & Logic Synthesis :***



Logical Design

- ***<u>Design Entry & Logic Synthesis :</u>***
- Logic synthesis is the process of converting a high-level description of the design into an optimal gate-level representation, given a standard cell library and certain design constrains.
- Standard cell library is a database containing logic gates such as AND, OR, NOT etc. It can have macro cells such as adders, multiplexers and flip-flops.
- Design constraints are set of constraints such as maximum chip area , operating speed, maximum power dissipation etc.

Physical Design

- Physical design is essentially a sequence or set of steps that converts a circuit netlist into the final layout, which is a set of rectangles.
- Or converts a circuit description into a geometric description.
  - This description is used for the fabrication of the chip
- Basic steps in the physical design cycle are:
  - Partitioning, floor planning and placement.
  - Routing
  - Static timing analysis
  - Signal integrity and cross talk analysis
  - Physical verification and signoff

Speed, Power and Area Considerations in VLSI Design

- ***Speed***:
- Increasing the operation speed will normally require a large area.(for e.g., duplicate the hardware in order to parallelize the computation) .
- The design process should, therefore, always carefully consider the trade-off between speed and area.
- Often the operation speed is part of the specification and the area should be minimized without violating this specification.
- Speed is then a design constraint rather than an entity to optimize.

Speed, Power and Area Considerations in VLSI Design

- ***Power disspation***:
- When a chip dissipates too much power, it will either become too hot and cease working or will need extra (expensive) cooling.
- Besides, there is a special category of applications, viz, portable equipment powered by batteries, for which a low power consumption is of primary importance.
- Here again there are trade-offs; designing for low power may for e.g., lead to an increase in the chip area.

Speed, Power and Area Considerations in VLSI Design

- ***Area***:
- Minimization of the chip area is not only important because less silicon is used but also because the yield is in general increased .
- Not all circuits that are manufactured function properly; the yield is the percentage of correct circuits.
- Cause of failure, like crystal defects, defects in the masks, defects due to contact with dust particles, etc. are less likely to affect a chip when its area is smaller.