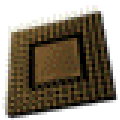
Introduction to ASICs

ni logic Pvt. Ltd., Pune



The Wonderful World of Silicon

About every two years, the number of transistors on a CMOS silicon chip doubles and the clock speed doubles… ..This rate of improvement will continue for the next 20 years.

- Technology Drivers:
  - Decreasing lithographic feature size, typically measured by the transistor gate length:

    0.35 μm … . 0.25 μm … . 0.18 μm 0.15 μm … etc.... 0.050 μm (?)
  - Increasing wafer size:

    6 inch diameter … .. 8 inch diameter … ..etc… .. 12 inches (?)
  - Increasing number of metal interconnect layers:

    4 … .. 6 … .. 8 … … 9 (?)

The Wonderful World of Silicon

- With the advent of VLSI in the 1980s engineers began to realize the advantages of designing an IC that was customized or tailored to a particular *system* or *application* rather than using standard ICs alone.

- This design paradigm shift was due to advancement in semi-conductor technology to satisfy the increasing complexity and performance needs of the applications.

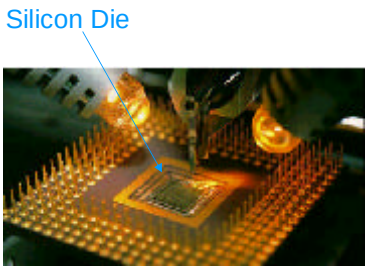
- Building a microelectronic system with fewer ICs allows you to reduce cost and improve reliability.

Agenda

- An Introduction
- ASIC Cell Library
- Types of ASICs and their comparison.
- Applications of ASICs
- ASIC Design Flow and Approach
- ASIC Vs FPGA
- ASIC Design Issues and Verification
- Backend Design and Issues
- FPGA to ASIC Conversion
- Packaging Technology
- Current Trend and Conclusion

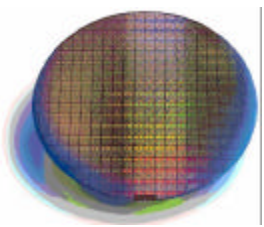
What are ASICs..?

- ASICs are silicon chips that have been designed for a specific application. Putting in other words, it is a chip designed to perform a particular operation as opposed to general purpose integrated circuits:
- An ASIC is NOT software programmable to perform different tasks.
- ICs that are not ASICs are :
  - DRAM
  - SRAM
  - 74xx series ICs
- ICs which are ASICs:
  - Baseband processor in mobile phone
  - Chipsets in PCs
  - MPEG encoders/ decoders
  - DSP functions in hardware, e.g. FFT



What Is In Them..?

- These are made on a thin circular wafer, with each wafer holding hundreds of dies.
- Each wafer consists of many mask layers, built on top of one another.
- Every mask layer corresponds to either a transistor
  - (semiconductor) mask or the interconnect (metal wires).
- The logic cells made up of transistors are fabricated on the semiconductor mask and the metal mask is used for the interconnection between them.



WHY USE ASICs?

Design Requirements

- Technology-driven:
  - Greater Complexity
  - Increased Performance
  - Higher Density
  - Lower Power Dissipation
- Market-driven:
  - Shorter Time-to-Market (TTM)
  - Cheaper with the competition

ASIC Cell Library

- Everybody has seen PCBs designed with the MSI or SSI discrete components..? How do they make it..?
- The PCB designer uses TTL, CMOS etc. component library for it.
- The libraries contains standard components with their mechanical, electrical and other specifications, which are fixed for the specific technology.
- The basic design principles are the same for designing on silicon as for using standard MSI or SSI parts on a PCB.

ASIC Cell Library

- The cell library is the key part of ASIC design.
- What is a Cell..?
  - An electronic functional unit normally defined in terms of its layout on silicon.
- Similar to PCB components, ASIC vendors have libraries build of Core Cells of the specific technology, viz 0.5 μ, 0.25 μ, or 0.18 μ.
- Each cell in an ASIC cell library must contain the following:
  - A physical layout
  - A behavioral model
  - A Verilog/VHDL model
  - A detailed timing model
  - A test strategy
  - A circuit schematic
  - A cell icon
  - A wire-load model
  - A routing model

FARADAY's ASIC CELL LIBRARY

A standard cell library includes the primitive cell library, the I/O cell library, RAM/ROM blocks, and Megacell functional blocks.

- Primitive Cell Library
  - Consists of basic primitive gates like, AND, NAND,XOR, Half & Full Adder, decoders, D F/Fs, Pull ups, Multiplexers, etc.
- I/O Cell Library
  - Divided into many groups.
  - True 2.5V programmable I/O
  - True 3.3V programmable I/O
  - 3.3V PCI I/O(66MHz)
- Megacells
  - 16-bit & 8-bit Micro Controller, MIPS R3000 compatible RISC, Programmable Peripheral Interface, Direct Memory Access Controller, etc.

The FS9000A is a 0.25 µm standard cell library from Faraday Inc.,USA

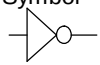
Inverter Cell

- Group Name : INV
- Function : Inverter
- Pin Order O I

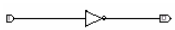
Truth Table

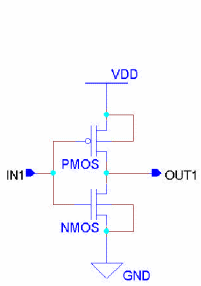
I	O
0	1
1	0

Symbol

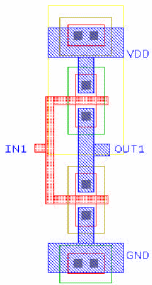


Schematic





Schematic



Layout

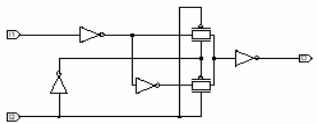
AC Characteristics (Temp=25˚C Vcc=2.5V Process=Nominal 1SL=0.0095PF)

Version	Cell Unit	Load (SL)											
		Path	1		2		4		8		16		32	
			tplh	tphl	tplh	tphl	tplh	tphl	tplh	tphl	tplh	tphl	tplh	tphl
INV1L	2	I-O	0.05	0.06	0.07	0.08	0.11	0.12	0.18	0.19	0.32	0.32	0.59	0.56

XOR Gate Cell

Symbol

Schematic



- Group Name : XOR2
- Function : Exclusive OR2

Truth Table

I1	I2	O
0	0	0
0	1	1
1	0	1
1	1	0

Pin Order O I1 I2

Input Capacitance & Maximum Loading (SL) & Power Consumption (uW/MHz)

Version	Input Capacitance		Maximum Loading	Power Consumption
	I1	I2	O	O
XOR2	1.2	2.4	48	0.13

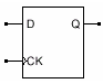
AC Characteristics (Temp=25˚C Vcc=2.5V Process=Nominal 1SL=0.0095PF)

Version	Cell Unit	Load (SL)											
		Path	2		4		8		16		32		64	
			tplh	tphl	tplh	tphl	tplh	tphl	tplh	tphl	tplh	tphl	tplh	tphl
XOR2	8	I2-O	0.17	0.18	0.19	0.21	0.24	0.26	0.33	0.36	0.51	0.53	0.87	0.86
		I1-O	0.16	0.17	0.19	0.20	0.24	0.25	0.33	0.35	0.51	0.52	0.87	0.85

D- F/F Cell

- Group name : QDFF
- Function : D Flip-Flop, Single Output

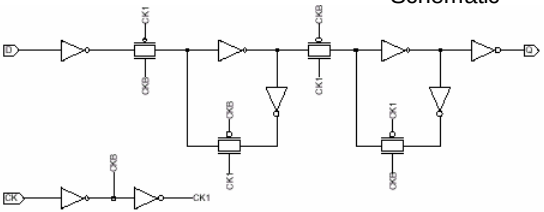
Symbol



Truth Table

CK	D	Q
↑	0	0
↑	1	1
↓	X	Q

Schematic



Pin Order: Q D CK

Tsu = 0.37 ns

Th = 0.11 ns

Input Capacitance & Maximum Loading (SL) & Power Consumption (uW/MHz)

Version	Input Capacitance		Maximum Loading	Power Consumption		
	D	CK	Q	Q	D	CK
QDFFN	0.7	0.8	47	0.31	0.09	0.16

AC Characteristics (Temp=25˚C Vcc=2.5V Process=Nominal 1SL=0.0095PF)

Version	Cell Unit	Load (SL)											
		Path	2		4		8		16		32		64	
			tplh	tphl	tplh	tphl	tplh	tphl	tplh	tphl	tplh	tphl	tplh	tphl
QDFFN	15	CK-Q	0.29	0.31	0.32	0.34	0.37	0.38	0.46	0.47	0.64	0.64	1.01	0.97

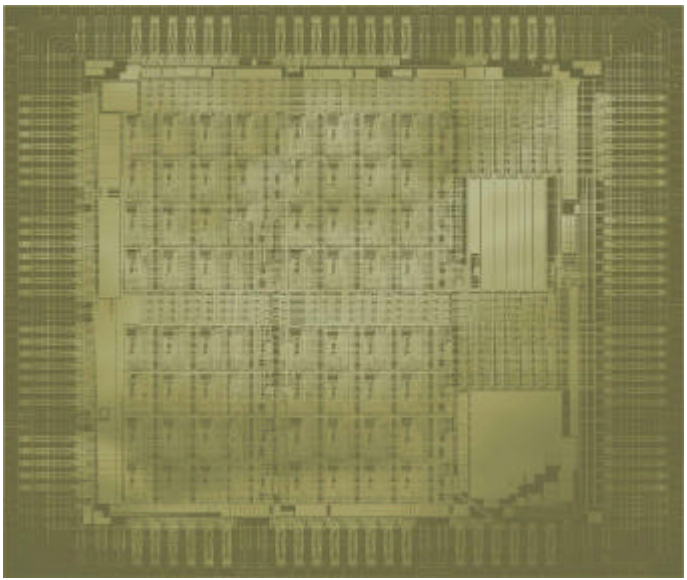
Types of ASICs

- ASICs are fabricated on a circular silicon wafer. The fabrication process remains the same, but the architecture makes ASICs to be divided into types.
- We will categorize the ASIC broadly in four types;
  - Full custom ASIC
  - Semi custom ASIC
  - Gate Array based ASIC
  - Programmable ASIC

Full Custom ASIC

- When engineers have a specific application to be designed and they are bothered about the performance, speed, power and cost, they go for designing Full Custom ASIC.
- The circuit is partitioned into a collection of sub-circuits according to some criteria such as functionality. Which are laid out specifically for one chip.
- Every transistor is designed and drawn by hand.
- Typically only way to design analog portions of ASICs.
- Usually used for layout of microprocessors.
- Full-custom design is very time consuming; thus the method is inappropriate for very large circuits, unless performance is of utmost importance

Full Custom ASIC



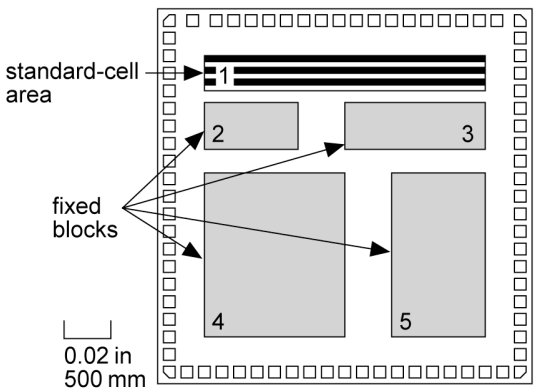
Actual 0.35? full-custom layout

Semi Custom ASIC

- A semi-custom ASIC, also known as a 'cell-based' ASIC, uses pre-designed logic cells (AND gates, OR gates, Multiplexers, Flip-flops etc.) known as standard cells.
- Simpler than full-custom design.
- Only the placement of the standard cells and the interconnection is done in a semi-custom ASIC. However, the standard cells can be placed anywhere on the silicon die.
- Possibly megacells , megafunctions , full-custom blocks , system-level macros(SLMs), fixed blocks , cores , or Functional Standard Blocks (FSBs). Eg. Microprocessor, multiplier, etc.

Semi Custom ASIC

- All mask layers are customized - transistors and interconnect
  - Automated buffer sizing, placement and routing
- Custom blocks can be embedded
- Manufacturing lead time is about eight weeks.

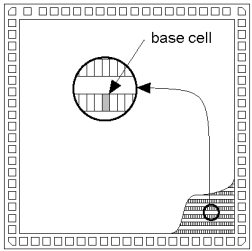


Gate-Array Based ASICs

- In a gate-array-based ASIC, the transistors are predefined on the silicon wafer.
- The predefined pattern of transistors is called the base array.
- The smallest element that is replicated to make the base array is called the base or primitive cell.
- The top level interconnect between the transistors is defined by the designer in custom masks - Masked Gate Array (MGA)
- Design is performed by connecting predesigned and characterized logic cells from a library (macros).
- After validation, automatic placement and routing are typically used to convert the macro-based design into a layout on the ASIC using primitive cells.
- Types of MGAs:
  - Channeled Gate Array
  - Channelless Gate Array
  - Structured Gate Array

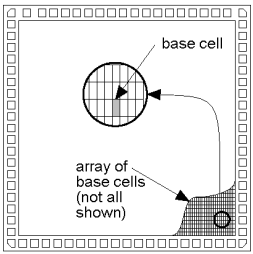
Gate-Array Based ASICs

- Channeled Gate Array
  - 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.



Channel gate-array die

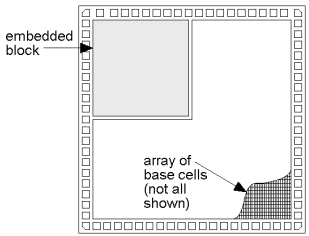
- Channelless Gate Array
  - There are no predefined areas set aside for routing, routing is over the top of the gate-array devices.
  - Achievable logic density is higher than for channeled gate arrays.
  - Manufacturing lead time is between two days and two weeks.



Sea-Of-Gates (SOG) array die

Gate-Array Based ASICs

- Structured Gate Array
  - Only the interconnect is customized
  - Custom blocks (the same for each design) can be embedded
    - These can be complete blocks such as a processor or memory array, or
    - An array of different base cells better suited to implementing a specific function
  - Manufacturing lead time is between two days and two weeks.

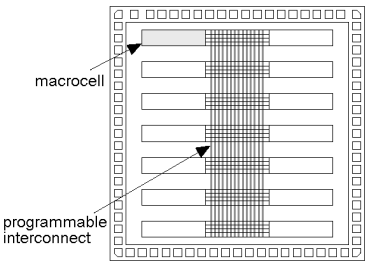


Gate array die with embedded block

Programmable ASICs

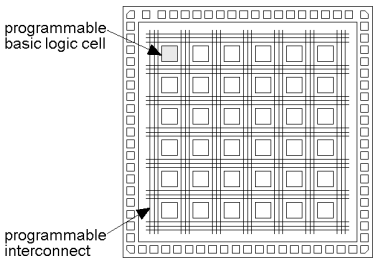
- Programmable Logic Devices
  - No customized mask layers or logic cells
  - Fast design turnaround
  - A single large block of programmable interconnect
    - Erasable PLD (EPLD)
    - Mask-programmed PLD
  - A matrix of logic macrocells that usually consist of programmable array logic followed by a flip-flop or latch.

Programmable Logic Device (PLD) die



- Field Programmable Gate Array
  - None of the mask layers are customized
  - 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



Comparison of different design styles

Architectural Difference

	STYLE			
	Full custom	Standard cell	Gate array	FPGA
Cell size	Variable	Fixed height	Fixed	Fixed
Cell type	Variable	Variable	Fixed	Programmable
Cell placement	Variable	In row	Fixed	Fixed
Interconnections	Variable	Variable	Variable	programmable

Comparison of different design styles

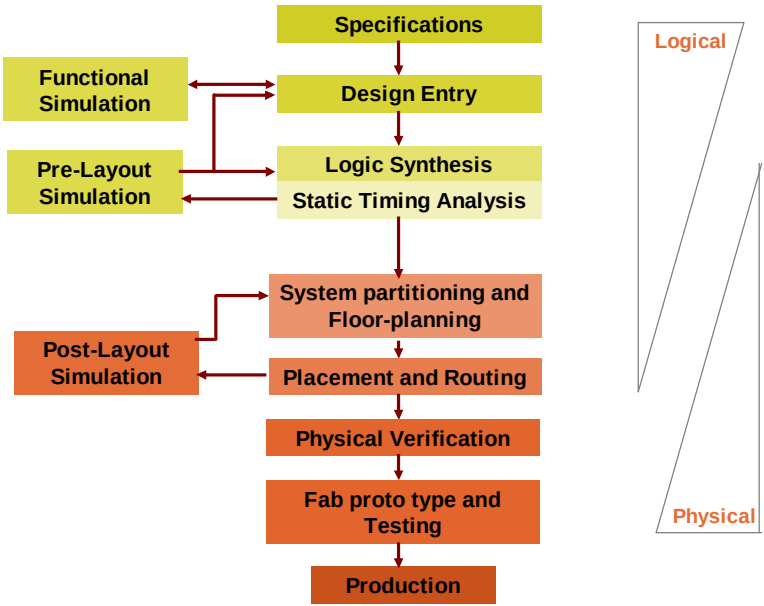
Comparison of Area, Performance, and Fabrication layers

	STYLE			
	Full custom	Standard cell	Gate array	FPGA
Area	compact	Compact to moderate	moderate	large
Performance	high	High to moderate	moderate	low
Fabrication layers	all	all	Routing layer	none

ASIC Applications

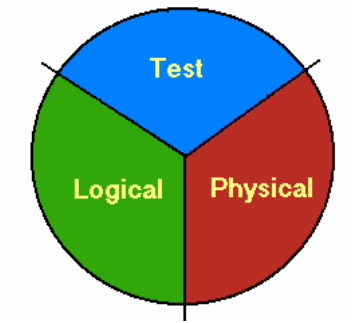
- The application field for ASICs could, in theory, be considered endless. Here are a few applications :
  - Aerospace subsystems and sensors
  - Wireless communication systems
  - Medical instrumentation
  - Telecommunications products
  - Consumer electronics, CDs, digital synthesizers, mini-discs
  - Computer products, graphics cards, MPEG technology.
  - Etc… … .

ASIC Design Flow



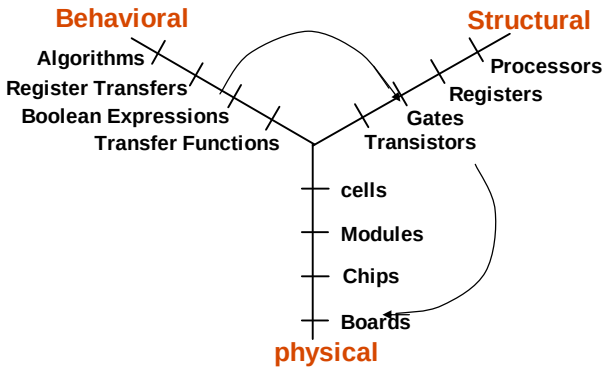
ASIC DESIGN METHODOLOGY

After going through the design flow, we can categorize it into three major phases;



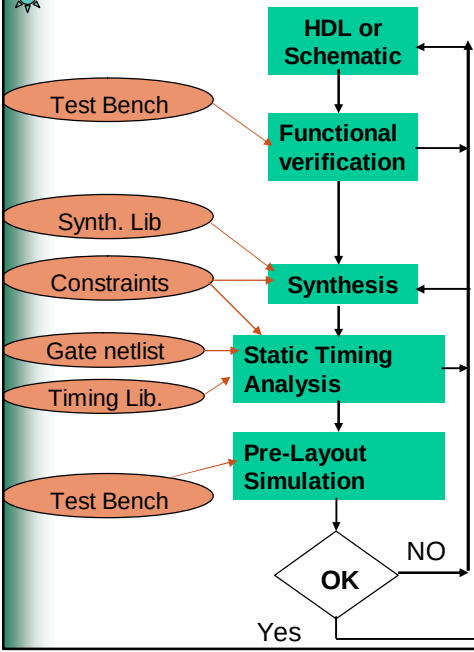
The ASIC Domains

Design Representation



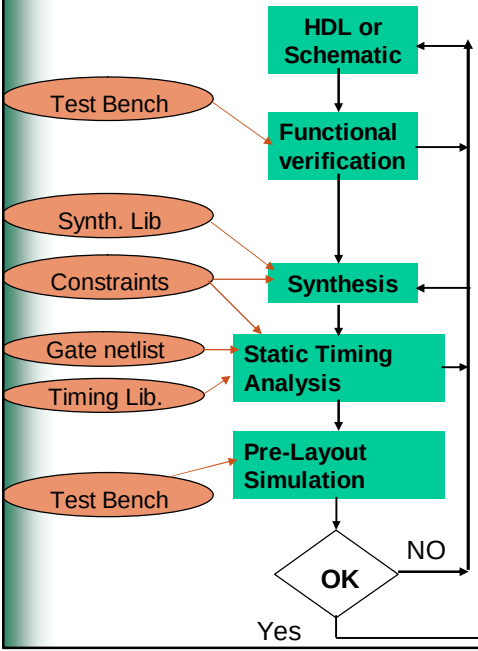
Gajski and Kuhn 'Y' chart

Logical Design, Phase - I



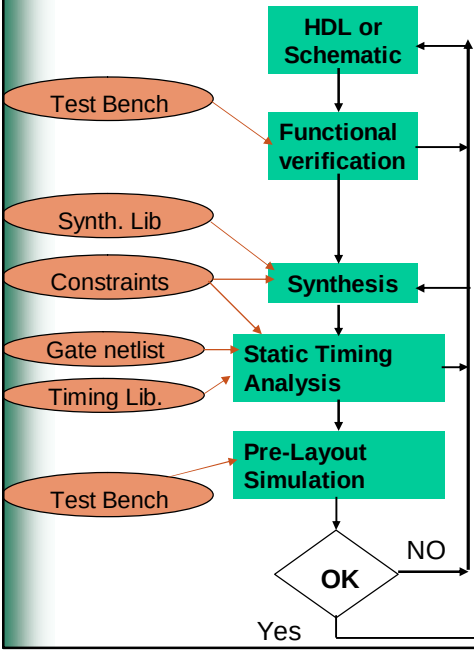
- HDL (Verilog, VHDL)
  - Synthesis from a HDL is the standard design entry method today.
  - Easier to update.
  - Easier to understand module level function.
- Schematic entry
  - Traditional method. Still quite often used.
  - Good system overview.
  - Simple to trace a signal flow.
  - What you see is what you get...

Logical Design, Phase - I



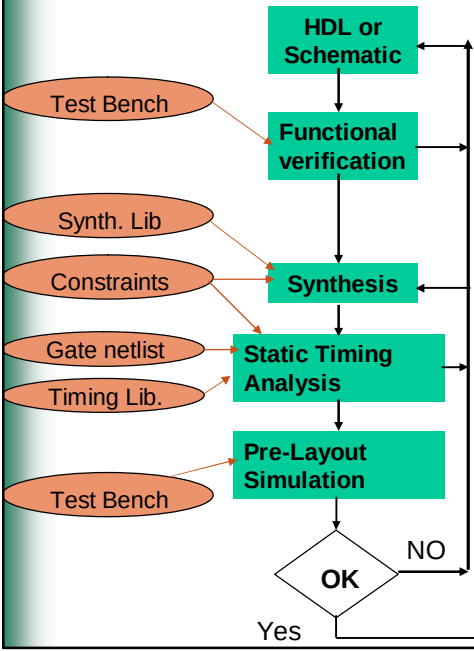
- **Functional Verification**
  - **Functional simulation of the design.**
  - **No timings are considered.**

Logical Design, Phase - I



- **Synthesis**
  - **Process for converting design specifications into gate level netlist.**
  - **Needs synthesis library containing target technology information.**
  - **Driven by the constraints.**

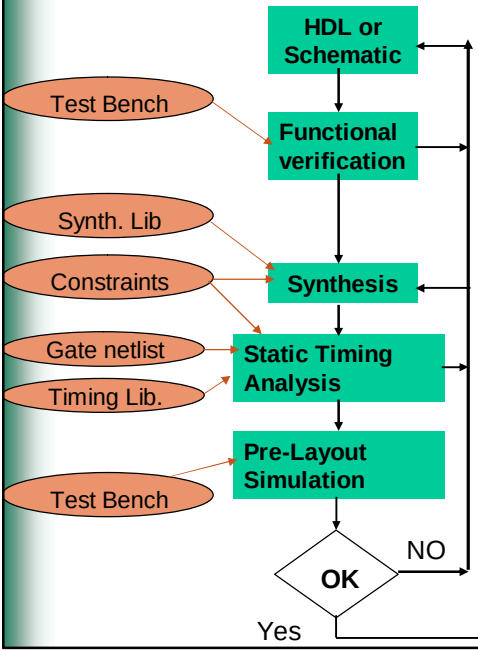
Logical Design, Phase - I



- **Static Timing Analysis**
  - Make sure the chip can run at specified frequency.
  - Detect and correct race conditions.
  - Independent of test vectors.
  - Helps synthesis into optimizing the logic.

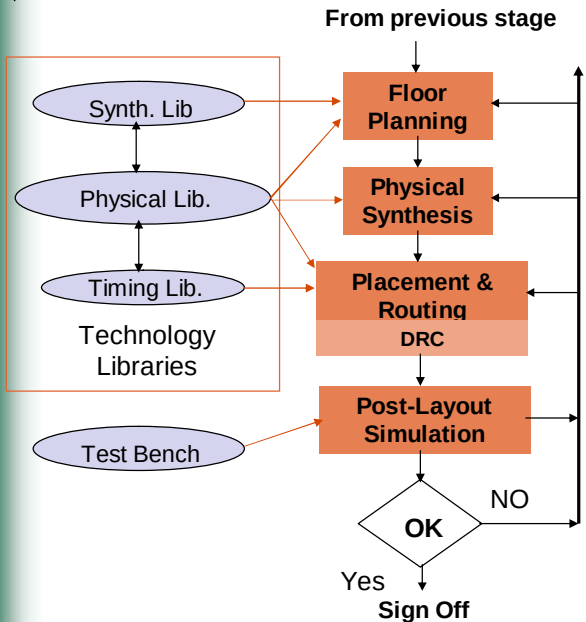
To next stage

Logical Design, Phase - I



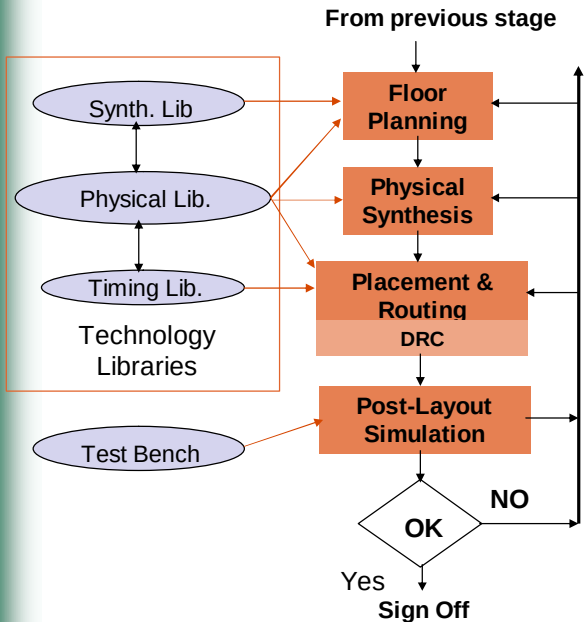
- **Pre-Layout Verification**
  - **Verification of design on the specified frequency, including the gate delays.**

Physical Design, Phase - II



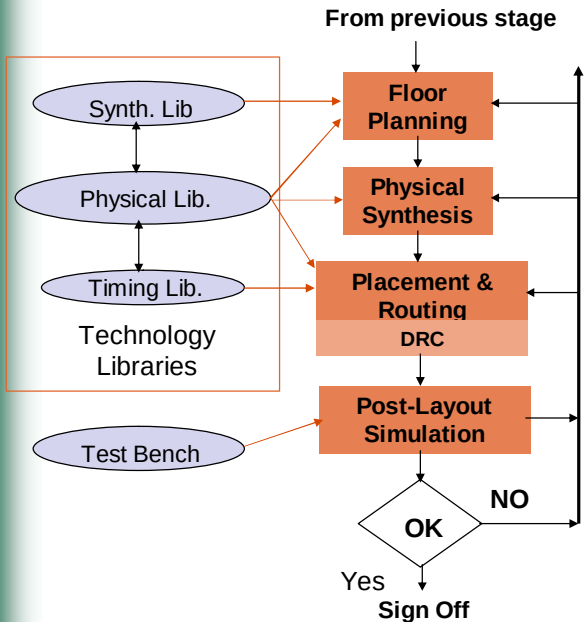
- **Floor Planning**
  - This stage of the design flow involves arranging the circuit blocks on the chip.
  - We have a netlist describing circuit blocks, the logic cells within the blocks and their connections.
  - What is now required is to arrange these circuit blocks intelligently on the silicon die.

Physical Design, Phase - II



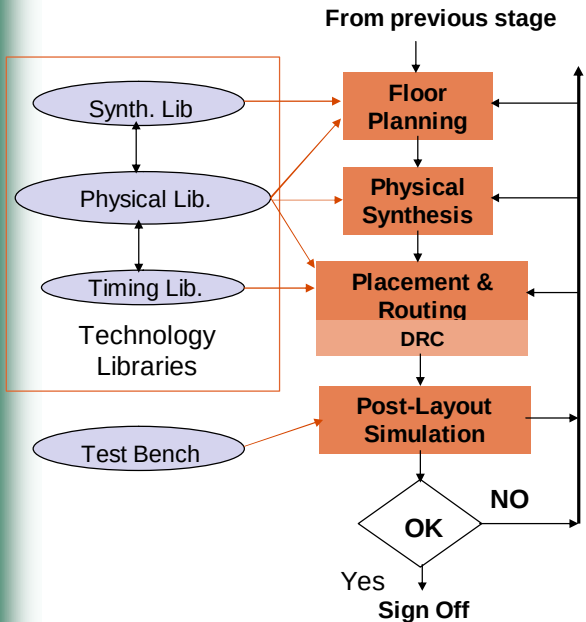
- **Physical Synthesis**
  - **Provide backend/physical information to the RTL synthesis tool to take care of timing coverage problem at the front end.**
  - ? **More powerful logic re-synthesis.**
  - ? **Need routing engine to provide accurate backend information.**
  - ? **Interconnect delays are longer than the logical delay.**

Physical Design, Phase - II



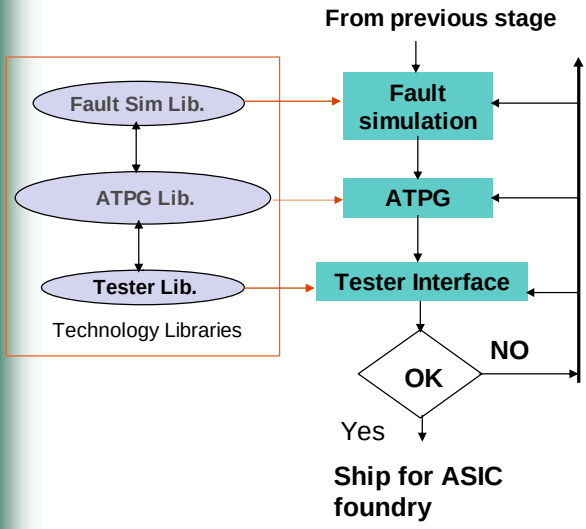
- **Placement and Routing**
  - Placement decides the exact location of cells in a circuit block.
  - Follows the floorplanning stage.
  - Finally connect the cells with interconnect material.
- **DRC(Design Rule Check)**
  - Checks for rules against the placement and routing of the logic cells.
  - To avoid interference, cross talk.

Physical Design, Phase - II



- **Post-Layout Simulation**
  - Verifying the design functionality considering the interconnect delay also.
  - Check that the design still works with added load of interconnects.

Testing, Phase-III



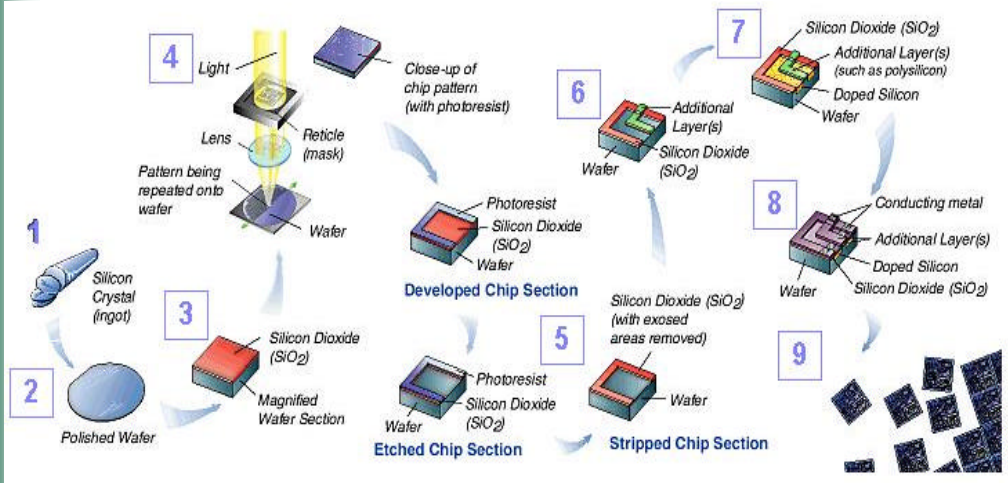
Testing aims at the detection of physical faults (production errors/defects and physical failures).

- **ATPG(automated test pattern generator)**

Semiconductor Manufacturing Process

- Silicon Wafers – The Chip's Foundation
  - Silicon wafers are preferred in today's chips because silicon is a natural semiconductor; it can conduct electricity or be converted to an insulator which prevents conduction.
  - Inexpensive and abundant; wafers are made of highly purified sand which has been refined to produce 99.9999% pure silicon.
- Semiconductor Fabrication-- Wafer Processing
  - Semiconductor manufacturers produce many kinds of ICs or chips. DRAMS, microprocessors, ASICs, and more. The precise process followed to make a chip varies according to the chip type and manufacturing company. However all wafer processing involves six basic steps:

Silicon Wafer Processing



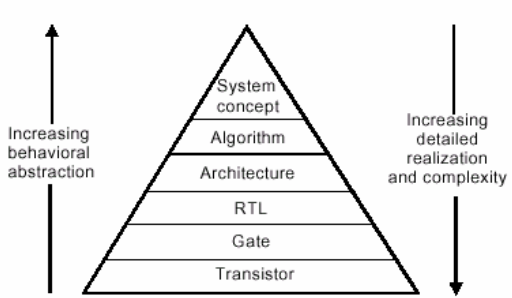
3-8 are the six basic steps involved in wafer processing.

Top-Down Approach

- In Top-Down design approach, the system is defined in ever-increasing levels of detail. Presumably, one then has everything defined completely before actually specifying a single gate.
- Usually possible when you start by having a very clear idea of the problem and how to solve it, which is better in most cases.
- Well suited for purely digital designs with relatively short turn-around times and moderate area performance requirements.
- Layout, place and route, differs between ASICs and FPGAs

Bottom-Up Approach

- Starts by defining the ‘low-level” design then moves up towards a more complex design using those that are already defined.
- Better suited for the design of very dense, high-performance digital blocks as well as analog and mixed-signal integrated circuits.
- Flexible at low level design issues (transistor size and parasitics optimization)



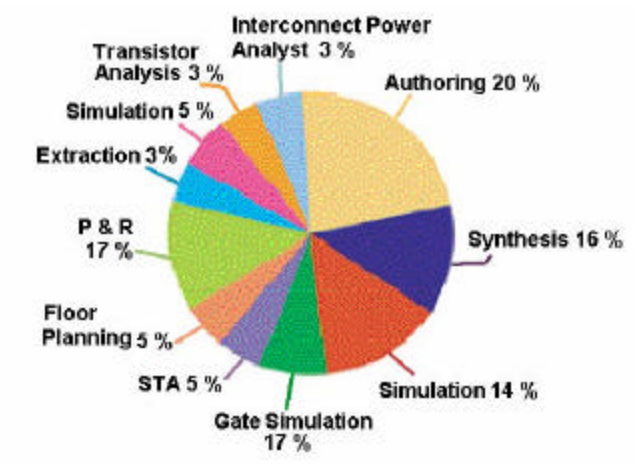
ASIC Vs FPGA

Biggest Difference - Designers mindset, because

- Waste is allowed.
- Cost per Gate is very Low in FPGA.
- No penalty for errors in FPGA Design.
- The experimental FPGA design mindset.
- Unsure about something? Try it and see what happens. Is dead wrong for designing ASICs.
- FPGA will always have extra resources then required.

Why Of FPGA?

- Why to choose an FPGA for the application?



Why Of FPGA?

- Why to choose an FPGA for the application **?**

ASIC Designers spend less and less time in the "window of innovation" (authoring, creating, and determining optimal feature sets) and too much time in the simulation and verification stages of the design cycle.

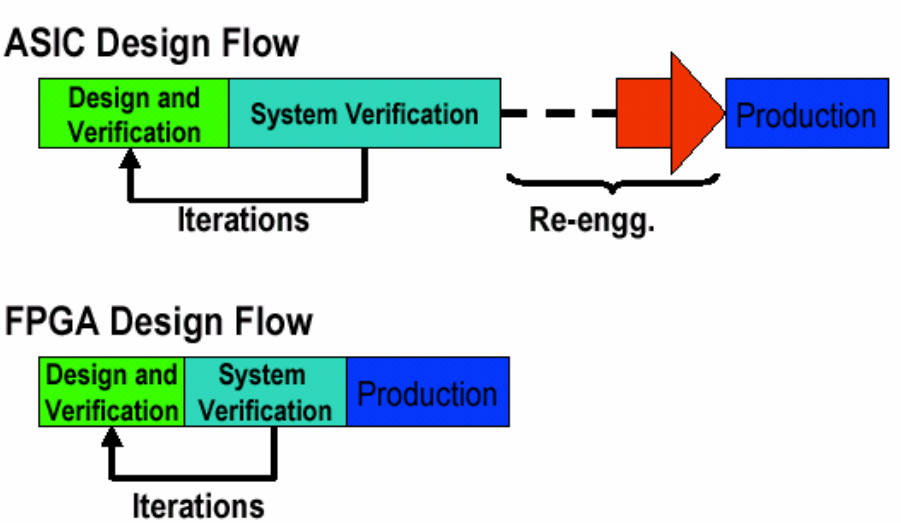
In fact, studies show that only 20% of the typical ASIC design process is spent in the innovation stage.

ASIC Vs FPGA

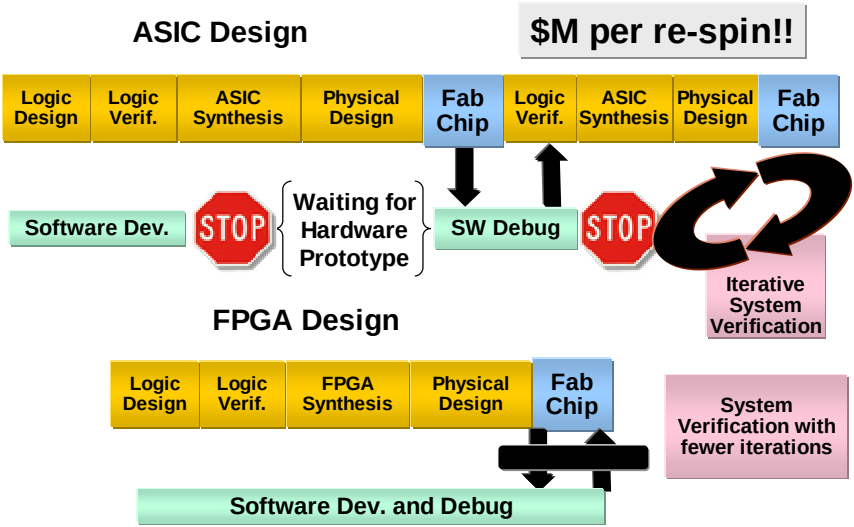
Basic differences in FPGA and ASIC Design strategies.

- FPGA have low-skew global networks for Clock and Reset. These have to created by the ASIC designer.
- There is a huge cost to making an error in an ASIC in terms of foundry charges and lead-time. This requires a careful, cautious, and conservative design approach with extensive testing.
- ASIC will have only the resources demanded by the design thus it will be smaller, use less power and operate at higher speed.
- High NRE (non recurring engineering) Costs.
- Test program development cost
  - Is the cost for generating test vectors & test programs for production testing.

FPGA vs ASIC Design Cycle



ASIC vs FPGA Design



Software Complexity

- ASIC designers are logic designers
  - Risk- averse and methodical
  - Spend lot of time verifying.
  - Don t want to spend time in physical design.
  - Separate engineers for physical design.

- FPGA designers
  - Would rather debug on the bench.
  - Realize must spend time in physical design.
  - Expect physical design to be "hands- off".

When to choose what? ASIC or FPGA

- Highly application dependent. What kind of algorithm are you trying to implement?
  - ASICs are used for specific application.
  - FPGAs are used for custom applications.
- Performance. High data rates?
  - ASIC have highest performance.
  - FPGAs can outperform in many applications.
- Flexibility, can the function of the device easily be altered?
  - FPGAs can be reconfigured.
  - Function in an ASIC is fixed.

When to choose what? ASIC or FPGA

- Is power consumption an issue?
  - ASICs have the lowest power consumption. Silicon is tailored for the specific application.
- Time to market
  - FPGAs, short.
  - ASICs, very long.
- Volume, cost
  - ASICs have high initial costs. Best choice for high volume designs makes them cheaper at the end.
  - FPGAs are most expensive, but low initial cost.

ASIC Design Issues & Verification

- Factors to consider before design
  - Performance
  - Functionality
  - Design techniques
  - Physical dimensions
  - Fabrication technology
- Specifications
  - Area
  - Speed
  - Power

What is RTL?

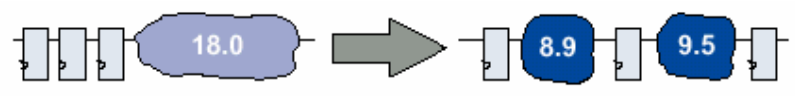
- RTL description: A design description in which the system is specified by:
  - Combinational logic blocks
  - Sequential Elements
- RTL description incorporates:
  - Specific architecture
  - A Clocking scheme
- An RTL design can be partitioned into two parts:
  - Data path design
  - Controller design.

RTL Designs

Pipelining [pre-computation]

- Pre-compute the output logic value one cycle before they are required and use this pre-computed value to reduce circuit switching in succeeding cycle.

- For a huge combinational Logic embedded within register, stages create large delays, thus effecting speed.
  - Solution is Split the combinatorial blocks and balance them using registers .



The Role of Synthesis

- What is logic synthesis?
  - It is the process of mapping a design specification into a detailed logic design based on a given library technology.
- Why logic synthesis?
  - Engineering Time
  - Improve Design Quality
  - Design Trade- Offs
- How logic synthesis is performed?
  - Transformation
  - Optimization
  - Technology Mapping

Clocking Strategy

- A lot of things must be considered when choosing clocking strategy:
  - Nature of the design
  - Power consumption
  - Chip area
  - Performance
  - Tool limitations
  - Design simplicity
  - Block reuse
  - Electromagnetic interference (EMI)
- Therefore, clocking strategy must be considered early in the project,

Clocking Methods

- Single phase clocking
  - "Industry Standard" method.
  - Well supported by synthesis and timing tools.
  - Clock skew is critical, requires clock tree synthesis.
- Two phase clocking
  - Clock skew not critical.
  - Lower peak power, reduced EMI.
  - Lower power consumption. (Not always true!)
- Multi phase clocking
  - May be needed in special cases (e.g. CCD control).

Low Power Design

- Total power consumption in an ASIC is given as
- $P_{Total} = P_{SW} + P_{INT} + P_{LEAK}$
- P_{SW} - Switching Power is 70%-90% in active circuits, also referred to as Capacitive Power. (dynamic dissipation)
- P_{INT} - Internal Power is 10-30% in active circuits.
  It is due to Short circuit power and any power consumed internal to gates (dynamic dissipation)
- P_{LEAK} - Leakage Power(<<1 % in active circuits)
  Sub-threshold leakage, some due to substrate leakage (static dissipation)

Power Optimization - Themes

- Power optimization can be done at
  - System level
    - Operate the circuit at lower supply voltage.
    - Operate the circuit at lower Frequency
      e.g. Laptop.
  - RTL level
    - Gated Clocks
  - Gate level
    - Buffer Insertion [To increase fanout]
    - Pin Swapping

$$P = FCV^2$$

RTL Level Tradeoff's

Using gated clocks

- Pros
  - Reduces capacitance on the clock network
  - Reduces internal power in the affected registers
  - Reduces need for muxes.
  - Large opportunity for power reduction exists as no. of registers in a design is large.
- Cons:
  - Testability
  - Complicates Clock skew balancing and Clock tree Synthesis.

Design Verification

- Is the design consistent with the original specification?
- Design is verified at different stages during the flow.
- Software Simulation
  - Application of simulation stimulus to model of circuit
- Hardware Accelerated Simulation
  - Use of special purpose hardware to accelerate simulation of circuit
- Emulation
  - Emulate actual circuit behavior - e.g. using FPGA's
- Rapid prototyping
  - Create a prototype of actual hardware
- Formal verification
  - Model checking - verify properties relative to model
  - Theorem proving - prove theorems regarding properties of a model

Test Requirements

- A number of things have to be considered when selecting the test methods:
  - Engineering time.
  - Fault coverage.
  - Tester limitations.
  - Manufacturer requirements.
  - Impact on chip area and timing.
  - Portability.
  - Compatibility with design styles.

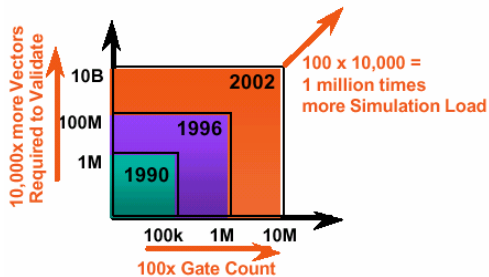
Test Generation

- Purpose:
  - To create stimuli patterns for production test.
- Automatic test pattern generation (ATPG) :
  - The creation of signals by computer-run algorithm for use with an integrated circuit tester.
  - ATPG signal sets (often called test vector sets) are commonly developed in conjunction with a specific Design- for-test technique, such as Scan design.
- Approaches:
  - Functional test vectors.
  - Scan chains.
  - Built In Self Test (BIST).
  - "Nonfunctional" tests.

  Note this step is done by the FPGA Vendor when he ships the FPGA, The designer has to do nothing

Status of Design Verification

- Software Simulation
  - Too slow
  - Moving to higher levels is helping – but not enough
- Hardware Accelerated Simulation
  - Too expensive
- Emulation
  - Even more expensive
- Rapid prototyping
  - Too ad hoc
- Formal verification
  - Not robust enough
- Intelligent Software Simulation
  - Symbolic simulation – not robust enough
  - Coverage metrics – useful, but not useful enough
  - Automatic vector generation – not robust enough



Frontend Design

- Starts from the *system level* with a top-down synthesis approach, which is technology independent (i.e., at this level, it is "irrelevant" whether the design will be implemented in CMOS or bipolar technology).
- The design is then transformed into the *circuit level*, in which the logic functionality, timing delays, speed and power, etc., are the primary concerns. This level is technology dependent but relatively process independent.
- If more detailed study on the transistor performance is needed, it can be supplemented at the *device level* based on the device physics which, in general, requires process information.

Backend Design

- At the backend, the final design must be translated into the physical layout representation, which is to be used to implement in wafer fabrication.
- In the "conventional" hierarchy, technology development (manufacturing) is relatively independent of the design.
- The "feedback" only occurs at the *circuit level* where the fab provides the circuit designer a set of SPICE parameters for the particular process through electrical measurement and parameter extraction of the fabricated transistors.
- I.e., Backend designing is not only technology dependent but also process dependent.

ASIC & FPGA Backend

- Logical designing for both remains the same up to the synthesis level, but the approach changes after the design is proceeded for the further stages.
- For an ASIC, there is a team for backend design, they prepare library, and convert the netlist , floor plan with front end designers, place the cells , design clock tree, routes the blocks ,estimates power consumption, etc.
- In FPGA design, the same team can lead for the backend process, as because the fabricated FPGA has less overheads than the ASIC.

Floorplanning

- This stage of the design flow involves arranging the circuit blocks on the chip, intelligently.
- The goals of Floorplanning are:
  - To arrange the circuit blocks on the chip.
  - To decide the location of the I/O pads
  - To decide the location and number of power pads (already done on an FPGA)
  - To decide the location of clock distribution nets.
  - To minimize the chip area.

Placement

- In FPGAs
  - Placement problem is very similar to ASICs
    - Fewer movable objects
    - 10M FPGA ~ 300,000 movable elements
  - Estimating delays during placement
    - Easier than ASICs
    - Finite set of routing resources
- In ASICs
  - Each block is movable.
  - Guarantee the router, for completing the routing step.
  - Better timings are achieved.

Delay Estimation In Placement

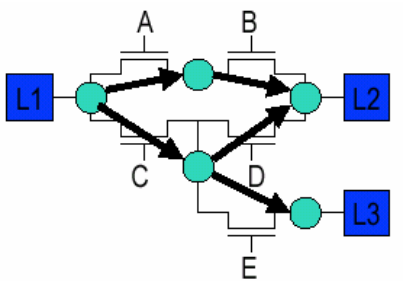
- FPGAs
  - Fixed set of most likely routes
  - Pre- computed delays for routes
  - Architecture makes delay predictable
- ASICs
  - RC tree analysis for proposed route
  - Computationally expensive
  - Very little pre- computation

Routing

- Interconnects placed instances
- Objective - minimize wire length and wire congestion within given constraints
- Use of channel-routing, over-the-cell routing, and routing layers
- Provides the most accurate delay data
- Routing Stages
  - Global routing - identifies general path for each net relative to other nets
  - Detailed routing - implements specific paths based on the general paths obtained from global routing

Routing

- For ASICs
  - Routing algorithms works according to the placement reports.
  - All interconnections are fixed.
  - Special nets are build for interconnecting the logic cells.
  - More tedious.
- For FPGAs
  - Routing algorithms are fixed, as the placement of the CLBs is predefined and fixed.
  - Interconnection is flexible.
  - Depends on the architecture.
    - Conductor segments are nodes
    - Programmable point are arcs
    - Architecture represented as a routing connectivity graph.



Engineering Change Orders (ECOs)

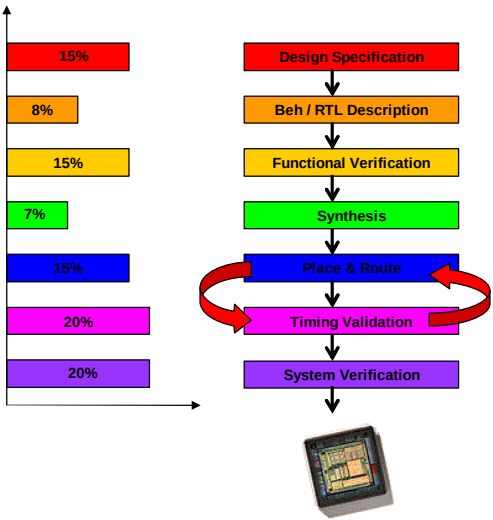
- Minor incremental changes in a design that make only changes to deal with a specific problem.
- Affect only small part of netlist.
- Applications:
  - Problem fixes late in the design process.
  - Changes in versions after chip deployed.
- Examples:
  - Timing failure - add or delete logic to solve.
  - Design rule violation - change to faster drivers.

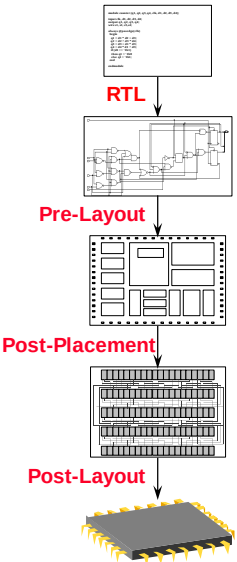
State-of-the-Art : Failures

Failures	%
Logical	55
Slow Path	13
Clocking	10
Power	6
Race Condition	4
Yield	4
Misc	3
IR drops	2
Mixed signal interface	1

With increasing competition it is better to SUICIDE than to commit mistake

A Designer's Life





Why this Slide..?

FPGA Prototyping

- Need for FPGA to ASIC Requirement comes from the fact that FPGA is an excellent prototyping platform.
- Most of the time is consumed in verification and testing for ASICs.
- But once the Design is finalized, then all the advantages of FPGA become a Overhead.
- E.g. requirement of configuration at power up and configuration memory.

FPGA to ASIC Conversion

List of Conversion Requirements.

- Netlists
  - Gate netlist, Simulation results
- Architectural/Design Information
  - A list of clocks and clock frequencies.
  - Gate-count estimate
  - No. of I/Os with drive strength, tristate, latched.[all I/O's in ASIC do not have the same specification]
  - ASIC technology cell library
- Physical specifications
  - Temperature grade
  - Pin list and locations. This includes power, ground, configuration, and unused pins.
  - Type of package(decided by the amount of pins, eg. BGA, PBGA)
- Others
  - Critical timing paths, RAM, FIFO's and other special logic modules.

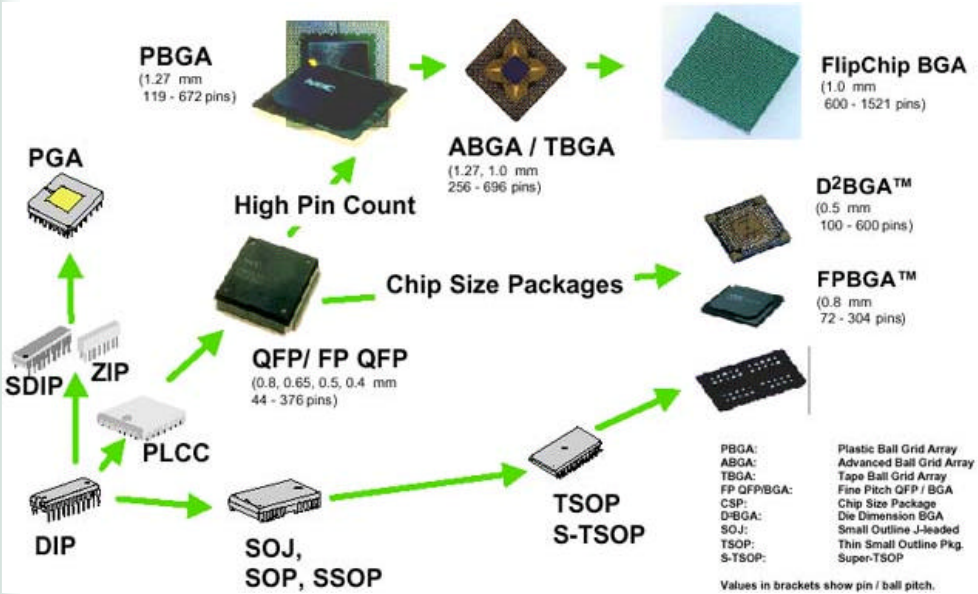
Packaging Technology

- Today's products can feature more than 10 million gate count. Having such greater number of functions on a die of silicon, manufacturers & users faces new and increasingly challenging electrical interconnect issues.
- At the same time, electronic equipment designers are shrinking their products, increasing complexity, and setting higher expectations for performance. To meet these demands, package technology must deliver higher lead counts, reduced pitch, reduced footprint area, and significant overall volume reduction.
- While packaging cannot add to the theoretical performance of the device design, it can have adverse effects if not optimized.

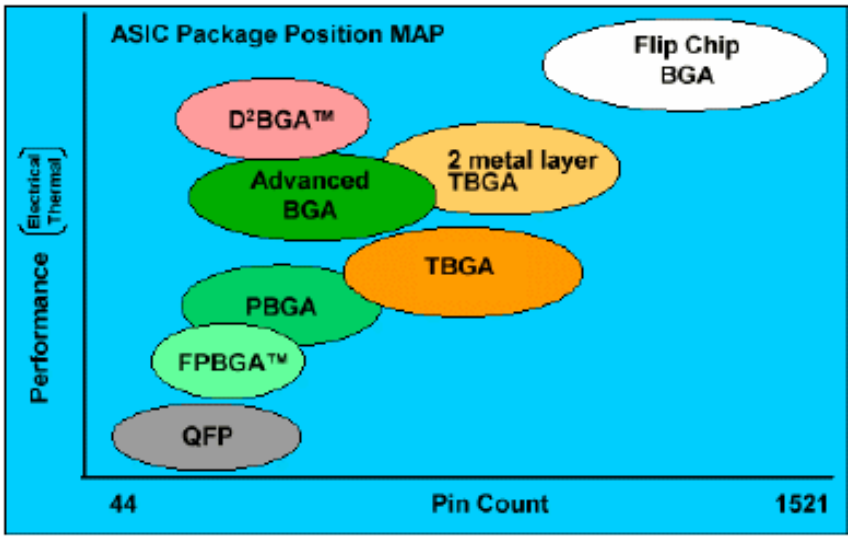
Package Family

- Actually, two mainstreams can be seen in packaging trend, which is going towards high performance, high pin count and the other in direction of small dimension packages to reduce the form factor of the device.
- New package types keep on adding to the family, few of the standard types are:
  - Ball Grid Array (BGA)
    - Has solder balls at the bottom of the device for lead connection.
    - High density
  - Chip Size Package (CSP)
    - Package with a small outline and high density.
  - Quad Flat Pack (QFP)
    - Cheap solution for low pin count.

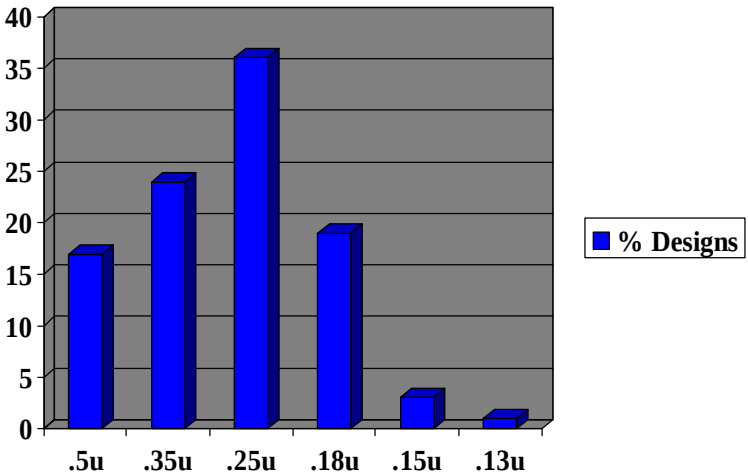
Package Family



Packaging Map



State-of-the-Art : *Technology*



Current Trends

- SOC(System On Chip)
  - ASICs & FPGAs are rapidly evolving to SOCs.
- IP(Intellectual Property)
  - Fueling the SOC paradigm.
- Design Planning
  - Merging Logical/ Physical/ Timing.
- VDSM(Very Deep Sub Micron)
  - Designs moving towards nanometer technology.
  - Interconnect delays are dominating gate delays.
- Power Analysis & Optimization.
  - With the extensive growth in gate count the power consumption is also increased.
  - Separate tools and engineers for power analysis of the chips.

Conclusion

- ASIC is a great solution for mass production and high-tech products. But, it requires non-significant investment in terms of cost, time, and human resources.
- If the complexity of the microelectronics technology will continue to grow, the migration towards higher abstraction level will continue.
- Technology is changing rapidly. It took 21 years to get to a 1Ghz processor, after it took 1 year to get to a 2Ghz processor, soon 3 Ghz processors would be unleashed.