ASIC Design Flow

Himanshu Patel

Space Applications Centre (ISRO)

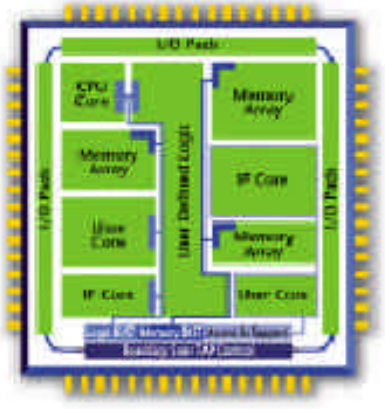
hnpatel@sac.isro.gov.in

Contents

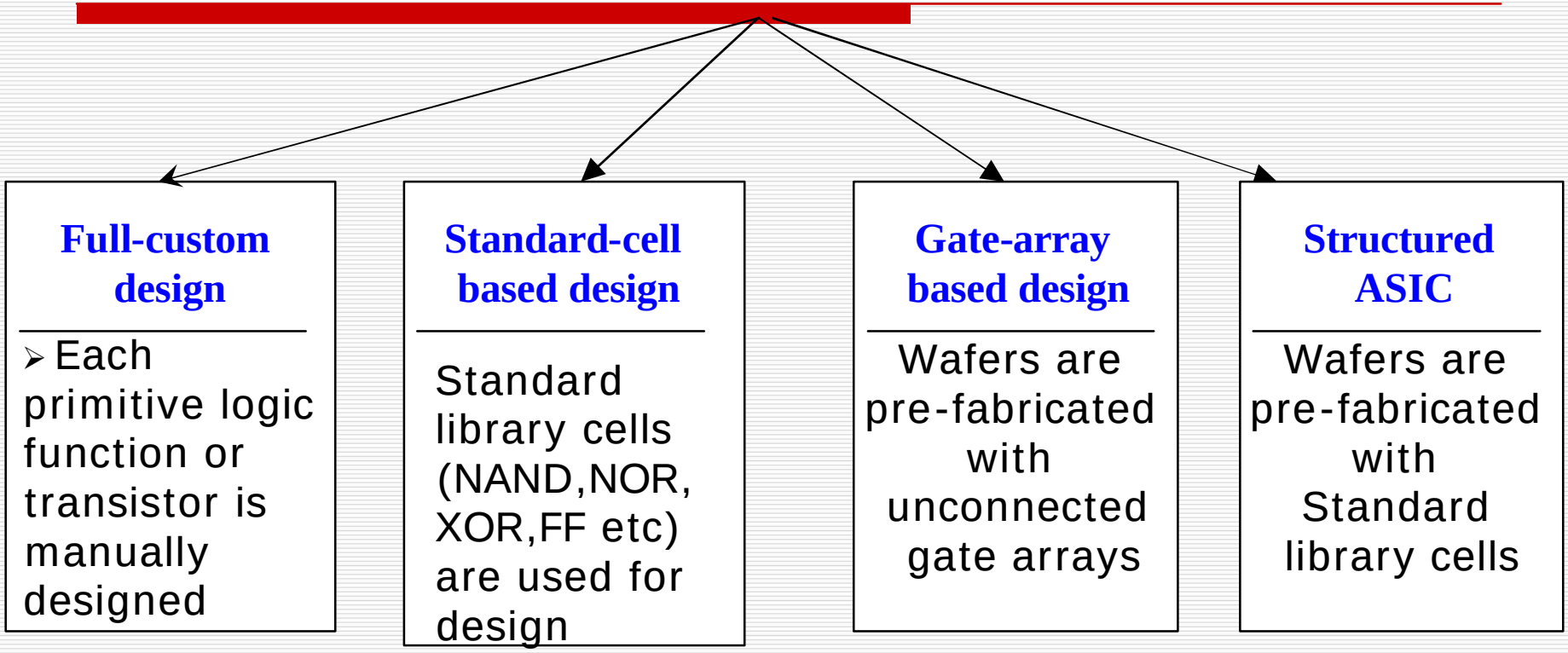
- **Introduction**
- **ASIC Design Methodologies**
  - Full custom
  - Standard Cell
  - Gate Array ASIC
  - Structured ASIC
- **ASIC Design Flow**
  - Design Entry
  - Functional Verification
  - Synthesis
  - Design For Test (DFT)
  - Place & Route
  - Timing Verification
  - Formal Verification
  - Proto ASIC Test
- **Mixed Signal ASIC**
- **Challenges for Deep Submicron ASIC**
- **CASE Study : OBC ASIC**

ASIC

- ASIC stands for **A**pplication **S**pecific **I**ntegrated **C**ircuits.
- It means an integrated circuit designed for a specific application.
- An application could be a microprocessor, cell phone, modem, router, etc.
- Nowadays, ASIC has a complete system on it, often called as System on a Chip (SOC)



ASIC Design Methodologies



Full custom ASIC

- Each primitive logic function or transistor is manually designed
- Manually manipulation of each transistor geometry is done, so also called *"polygon pushing"*
- Rarely used today, (except for very high volume products like microprocessor etc)

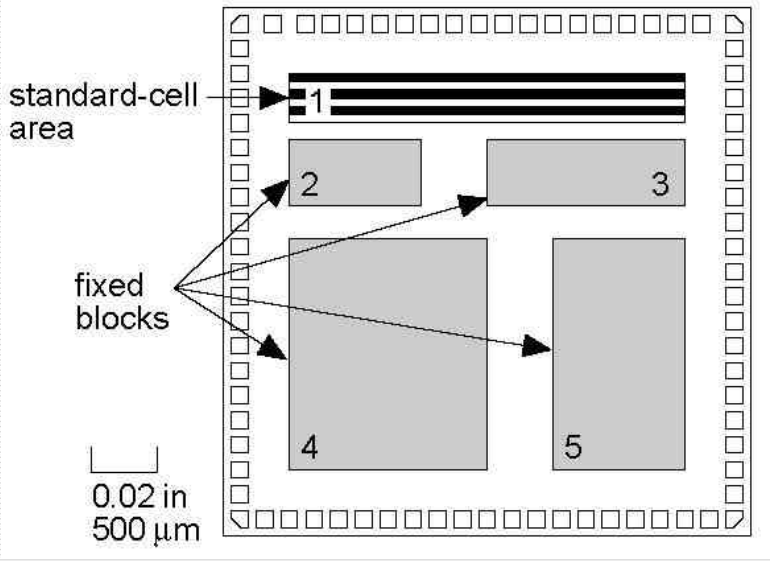
Advantages:

- It gives most optimized design : high speed, low power, small gate count

Disadvantage:

- Long design cycle
- Higher NRE (Non- Recurring Engineering) cost

Standard Cell based ASIC design



- Pre-defined library cells (NAND,NOR,FF,RAM, Hard macro cores etc) are used
- Designs are created using schematic capture or synthesis from Hardware Description Languages (HDL)
- All mask layers are customized —transistors and interconnect

Standard cell ASIC

Advantages:

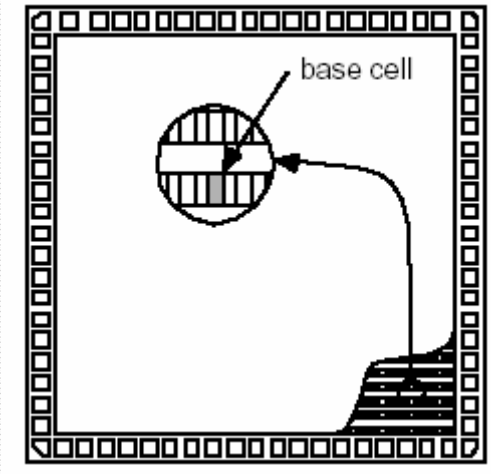
- Shorter design time compared to full custom style
- “Mega cells” or Hard IP cores (Microprocessor, MAC Memory etc) provided by vendor can be used easily

Disadvantage:

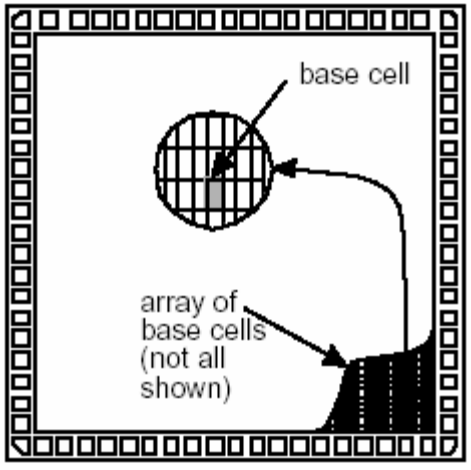
- Separate fabrication mask is required for each design : High NRE cost compared to gate array

Gate Array ASIC

- Wafers are pre-fabricated with unconnected gate arrays (so wafers are common for all design)
- Top metallization for connecting transistors is done according to different design at last stage



channeled gate array:
The interconnect uses predefined spaces between rows of base cells



channelless gate array:
Only some (the top few) mask layer are customized

Gate Array ASIC

Advantages:

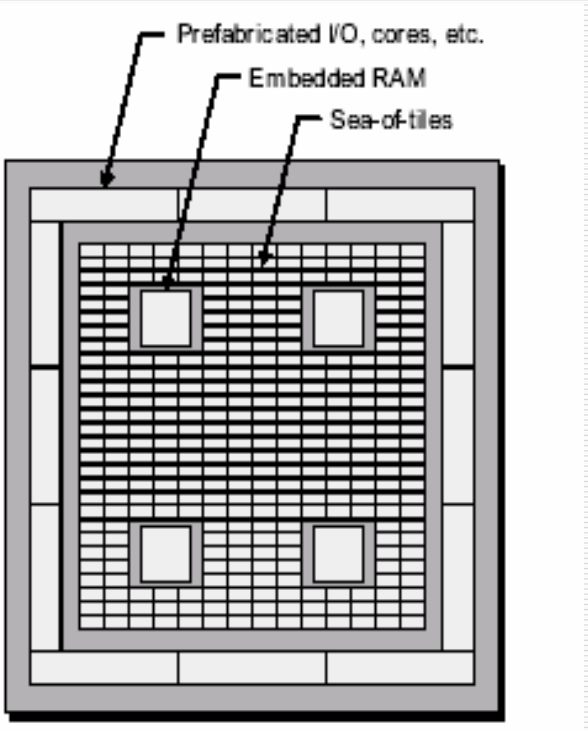
- Lower NRE cost as same base wafer is pre-fabricated for multiple designs
- Low turn around time

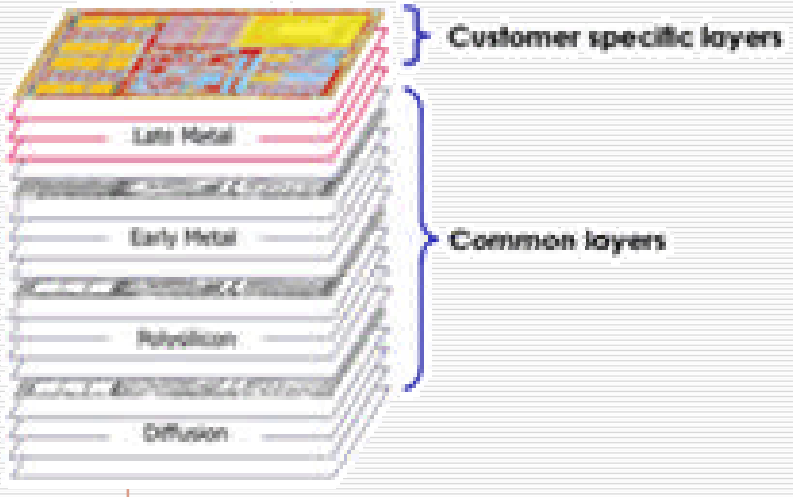
Disadvantage:

- Low layout density,
- less optimized: Low speed, High power consumption
- Suitable only for lower volume products

Structured ASIC

- A Structured ASIC falls between an Gate Array and a Standard Cell-based ASIC
- The design task involves mapping the design into a library of building block cells, and interconnecting them as necessary.
- Largely Prefabricated
  - Components are “almost” connected in a variety of predefined configurations
  - Only a few metal layers are needed for fabrication
  - Drastically reduces turnaround time





Structured ASIC

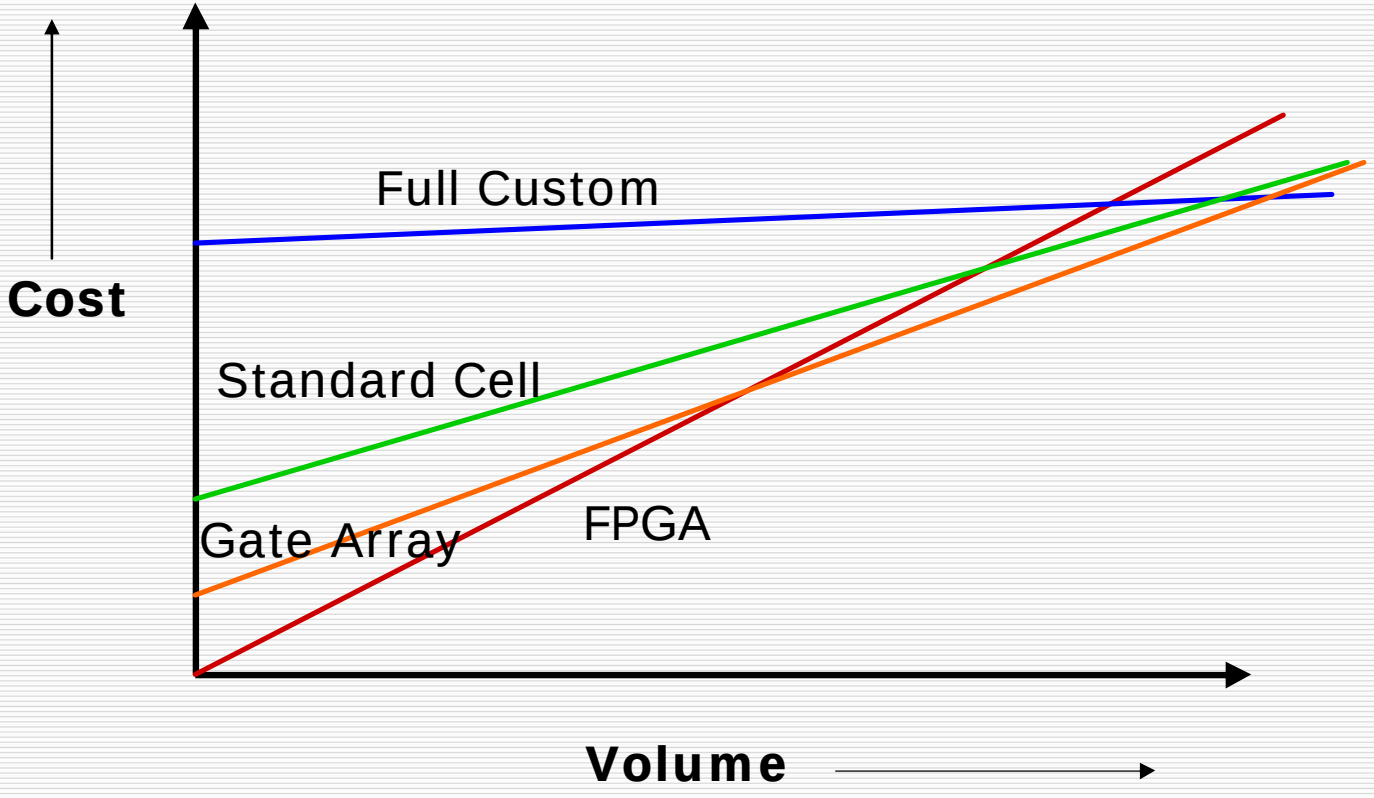
Advantages:

- Low NRE cost
- High performance
- Low power consumption
- Less Complex
  - Fewer layers to fabricate
- Small marketing time
  - Pre-made cell blocks available for placing

Disadvantages:

- Lack of adequate design tools
- Design constrained by pre-fabricated block available in library

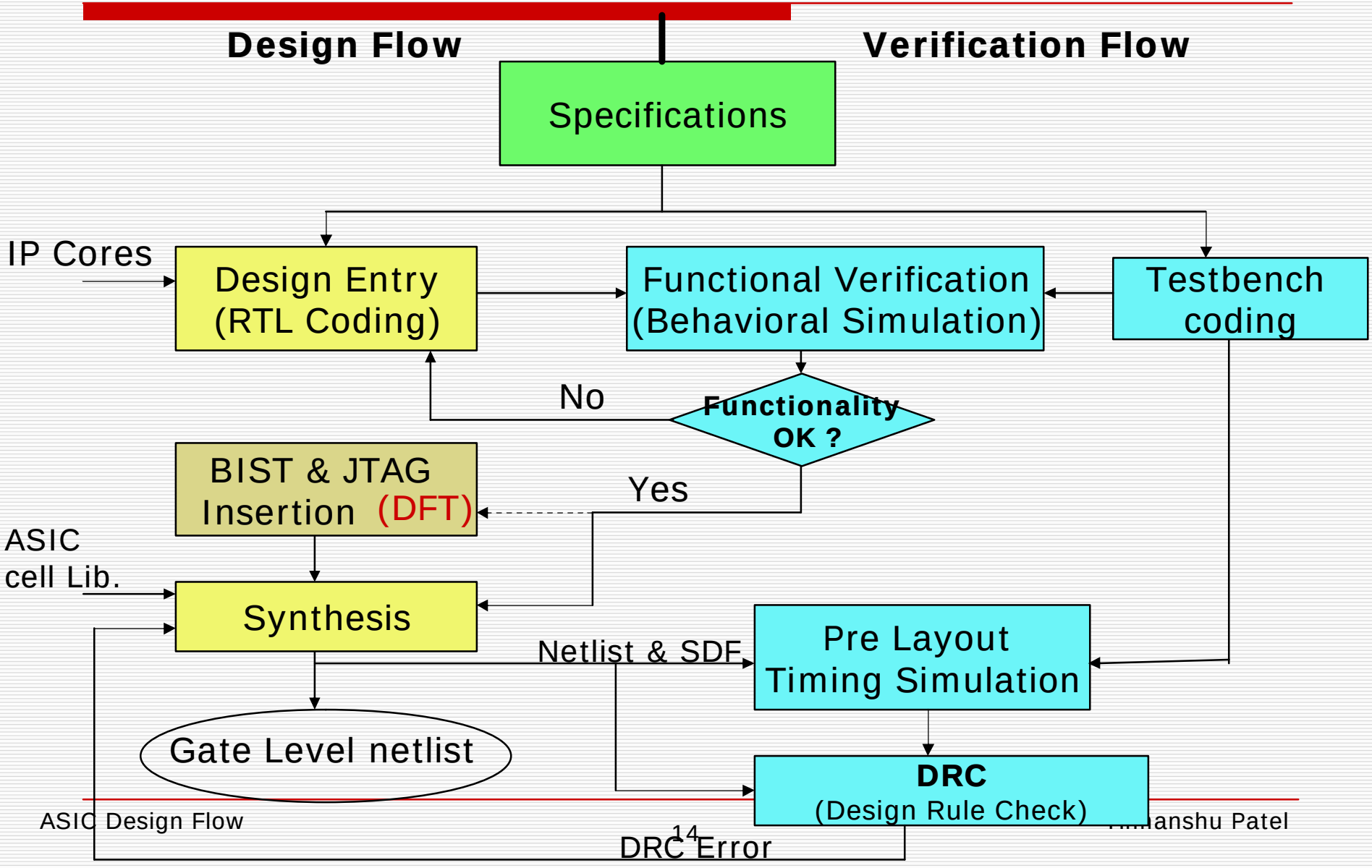
Comparison Graph



Digital ASIC Design Flow

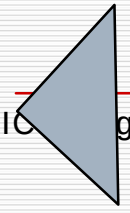
1. **Front End design :**
   - Specifications to Gate level netlist generation
   - Normally done by customer
2. **Back End design :**
   - From Gate Level netlist to GDS-II generation
   - Normally done by vendor or third party designer
3. **ASIC Fabrication:**
   - GDS-II to ASIC chip
   - Done by foundry

Front End Design



Specifications

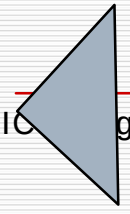
- The chip functionality is described in *"Requirement & Specification Document"*
- The targeted speed, power consumption, area are also specified
- System Engineer conveys requirement in plain English to Design team and Verification team
  - Design Team generates RTL code as per specs.
  - Verification team generates Test benches/test cases as per specs



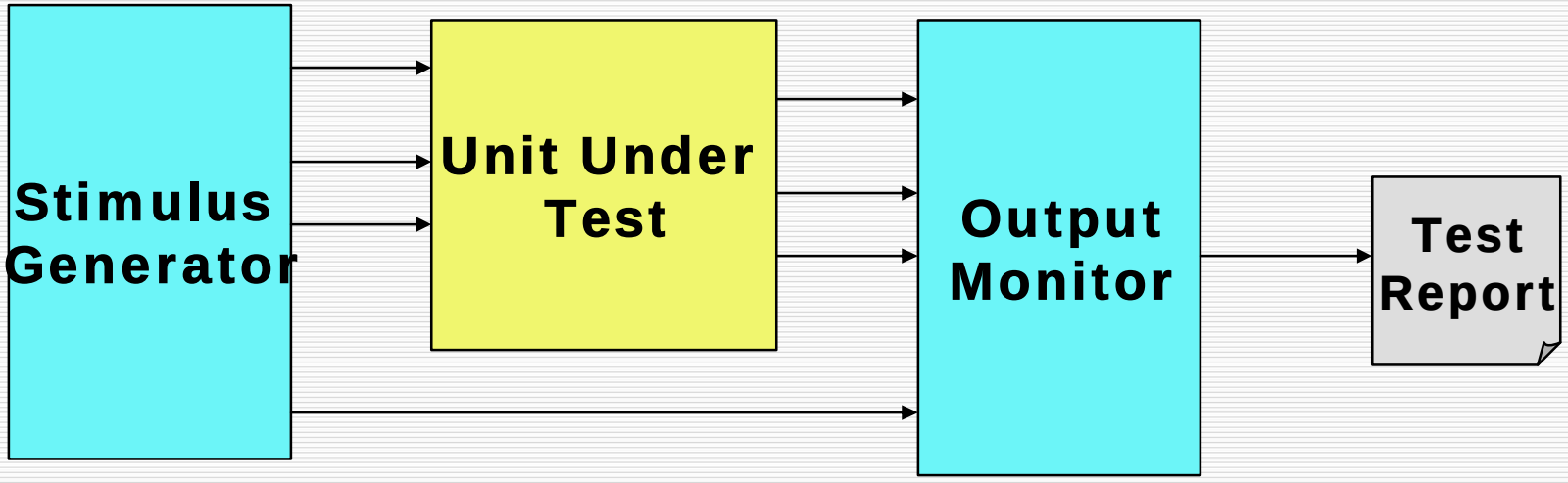
Back

Design Entry

- Either by Scematic Capture or through HDLs like VHDL,Verilog etc
- The quality of final chip depends largely on quality of RTL code
- There are some design guidelines which should be followed
  - Design should be synchronous
  - Clock gating should be avoided
  - Flip flops should be used instead of latches
  - Proper FSM coding styles (one hot, binary, etc)
- IP Cores or third party soft cores are used for standard blocks like processor, MAC, UART etc



Testbench



- First Test plan is worked out based on which different test cases are identified
- Assertion based testbenchs checks captured output with expected output and writes report

Back

Functional Verification

- The functionality of RTL code is verified using testbenchs it is also called "behavioral Simulation"
- Some of the popular simulators :
  - ModelSim®
  - NCSim®
- Code coverage indicates how much portion of RTL code is covered by testvectors
  - Statement coverage
  - Expression coverage
  - Branch Coverage
  - Toggle coverage
- Typically a "good" testbench achieves more than 95% code coverage

Design For Test

- Along with user logic, extra blocks are added for detection of manufacturing defects
- DFT is "structural test" (unlike, Dynamic simulation which is functional test)
- DFT methodology
  - Built In Self Test (BIST)
  - Boundary Scan chain (JTAG)
  - Internal Scan Chain

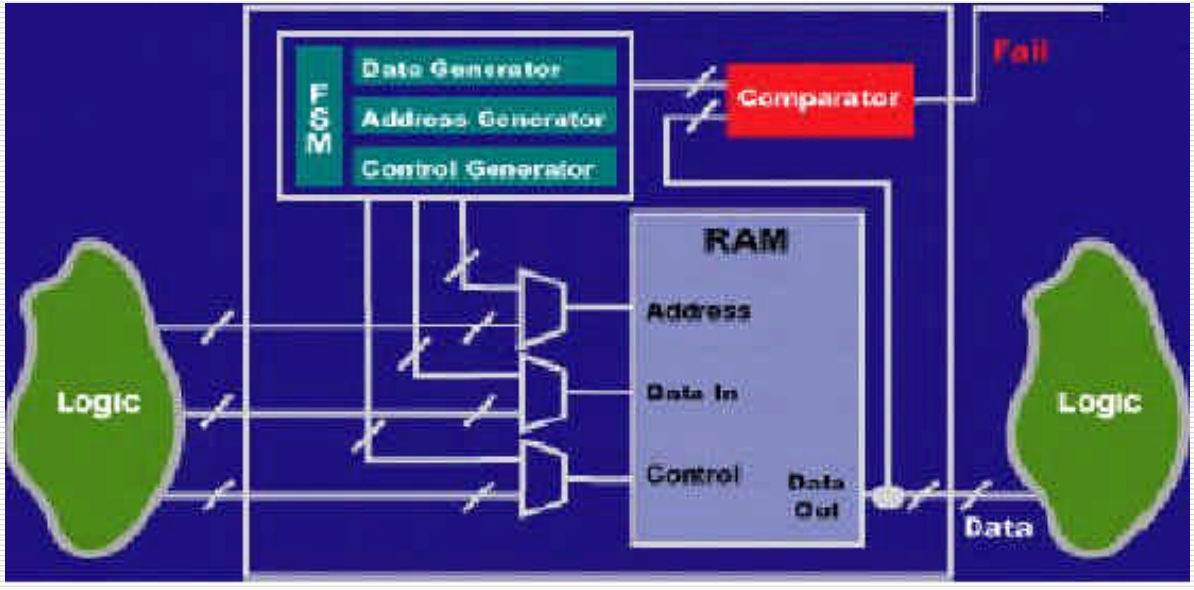
RTL level

Netlist level

- **DFT Advantages:**
  - Improve quality by detecting defects
  - Make it easier to generate vectors
  - Reduce vector generation time
- **DFT Disadvantages:**
  - Area overhead of 10-15%

BIST (Built In Self Test) Insertion

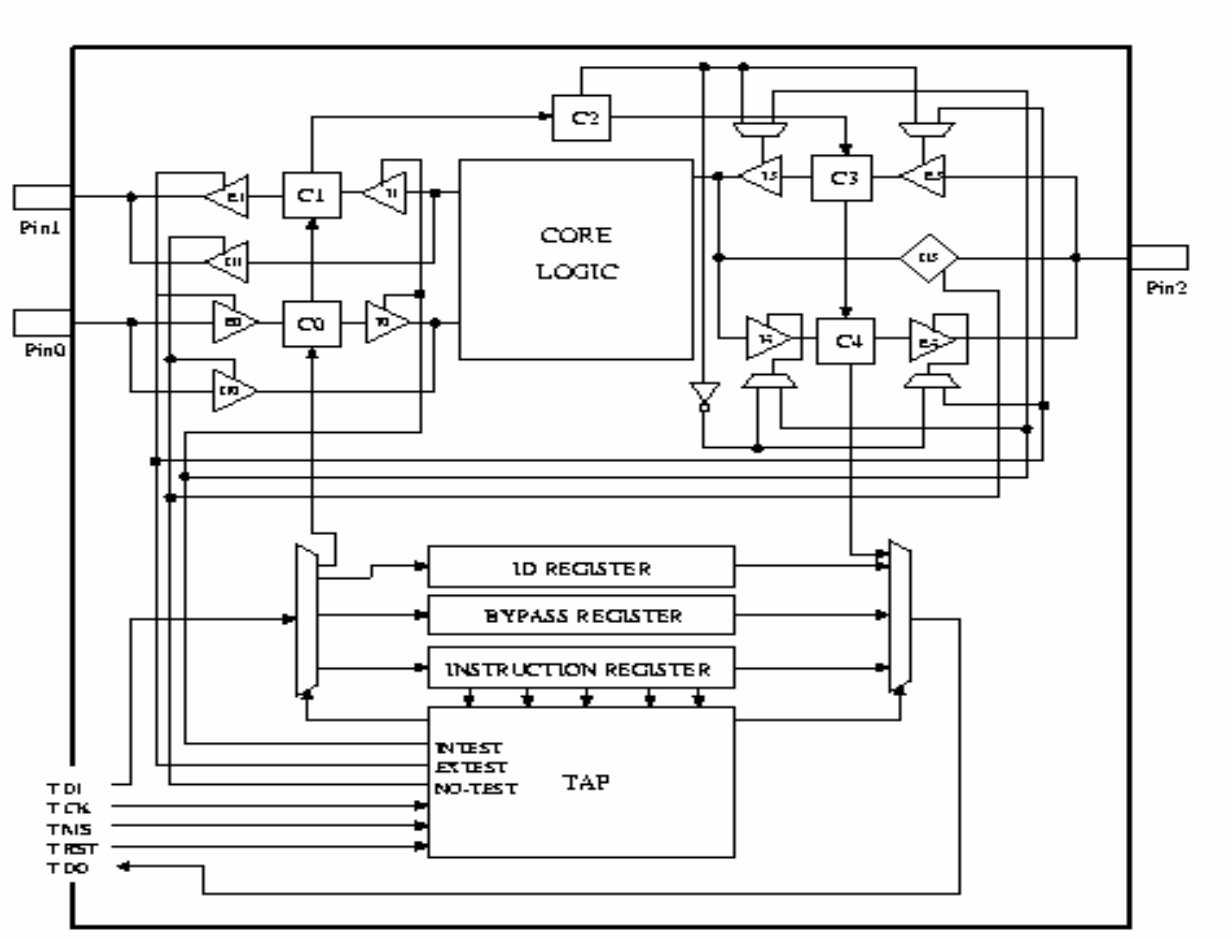
- Along with user logic, additional blocks are added for self test
  - Memory BIST (MBIST)
  - Logic BIST (LBIST)



MBIST

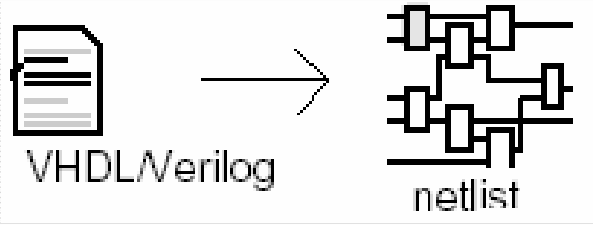
JTAG Insertion

IEEE 1149.1 standard for Boundary Scan test



JTAG Boundary Scan Interface Architecture

Synthesis



- Process of converting RTL code in to gate level netlist
- ASIC Vendor provides Cell library of basic gates (AND,OR,FF,RAM,FIFO..)
- Pre synthesized IP Core blocks (DesignWare,...) are treated as “Black Box”
- Some of the popular synthesizer :
  - Synopsys DC, Cadence Ambit BuildGates Synplify ASIC

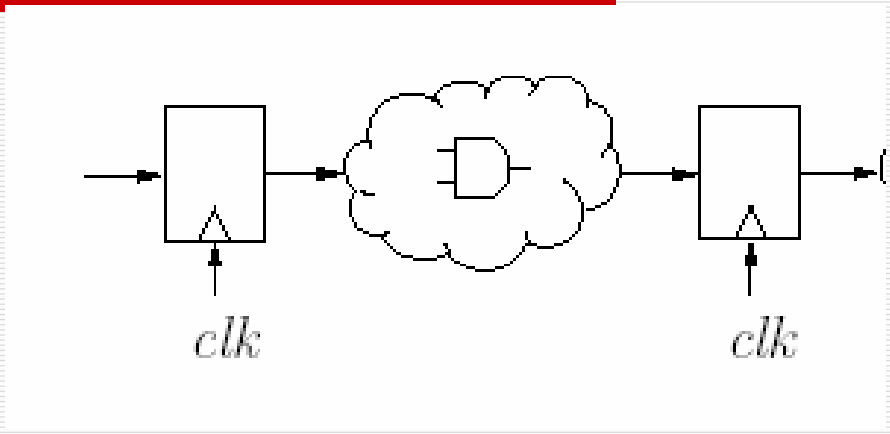
- Synthesis Script contains directives for synthesizer
  - Optimization goal : speed/area
  - Timing constraints
  - FSM encoding style etc

- Special care should be taken for *"high fanout nets"* like clock & reset
  - They are not synthesized at this level
    Set_dont_touch_network
  - During Clock Tree Synthesis process, layout tool creates optimized clock tree

Back

Pre Layout Timing Simulation

- Two types Timing Delays
  - Cell delay (input to output of a cell)
  - Propagation delay (o/p of cell1 to i/p of cell2)
- Delays are specified as (min, typ, max) depending on PVT (Process, Voltage, Temperature) condition
- *Wireload models* are used to estimate propagation delay based on fanouts because at this stage Layout is not done
- Setup violations must be addressed
  - Pipelining
  - Register retiming (balancing combi. Logic)
- Hold violations can be *ignored* at this stage



Setup violation: Data late, clock early

Max delays are considered

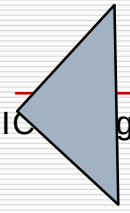
Decides maximum clock frequency

Hold violation: Data early, clock late

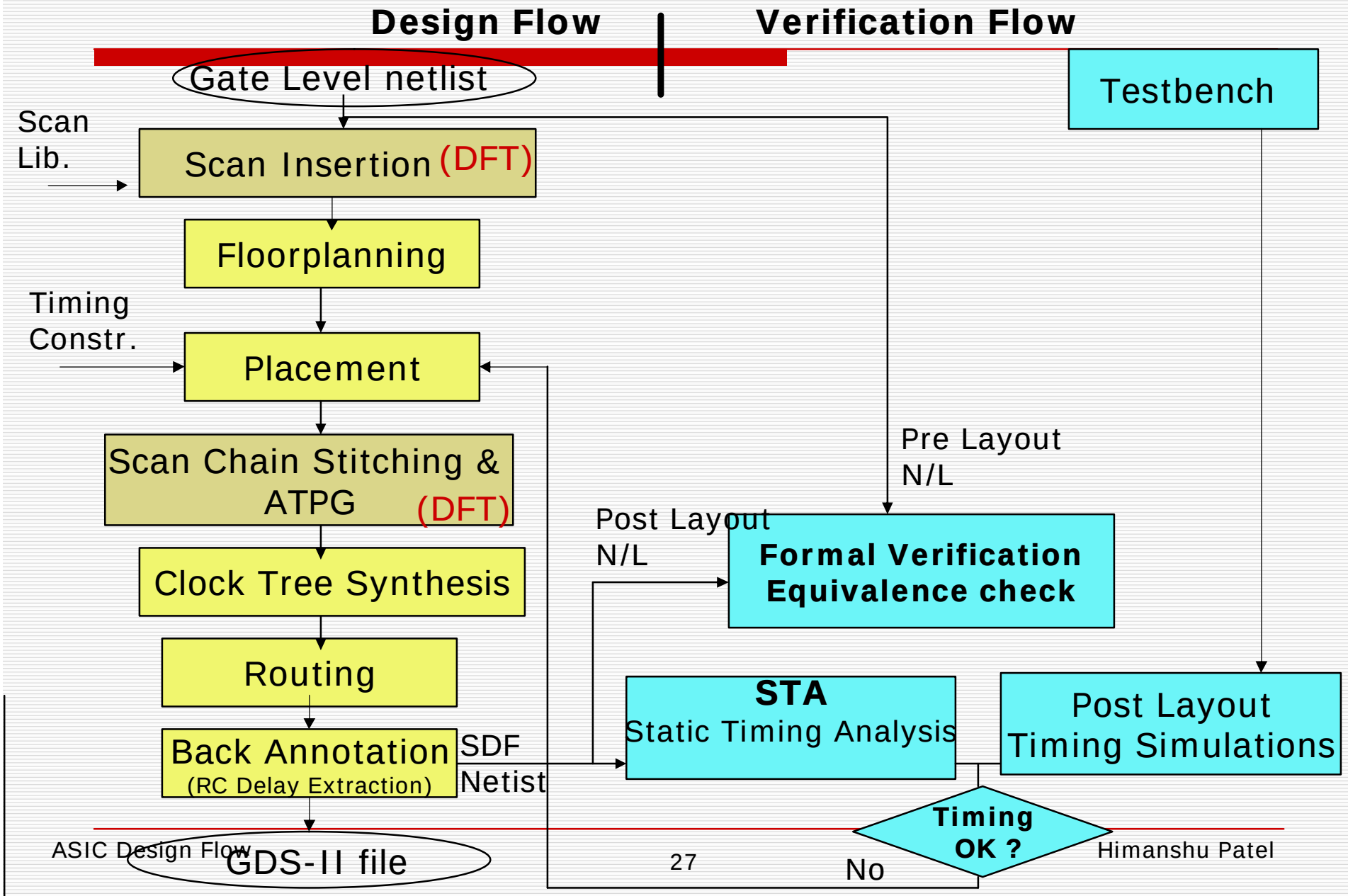
Min delays are considered

Design Rule Check (DRC)

- The gate level netlist must be checked for "design rules" before starting Back End design
- There are different DRCs/LRCs
  - Illegal net connections (two outputs shorted, etc)
  - Drive load limit violations
    - (fanout of driver < total load to output)
  - Undriven nets
  - Naming convention errors
- DRC tool kit is provided by ASIC foundry



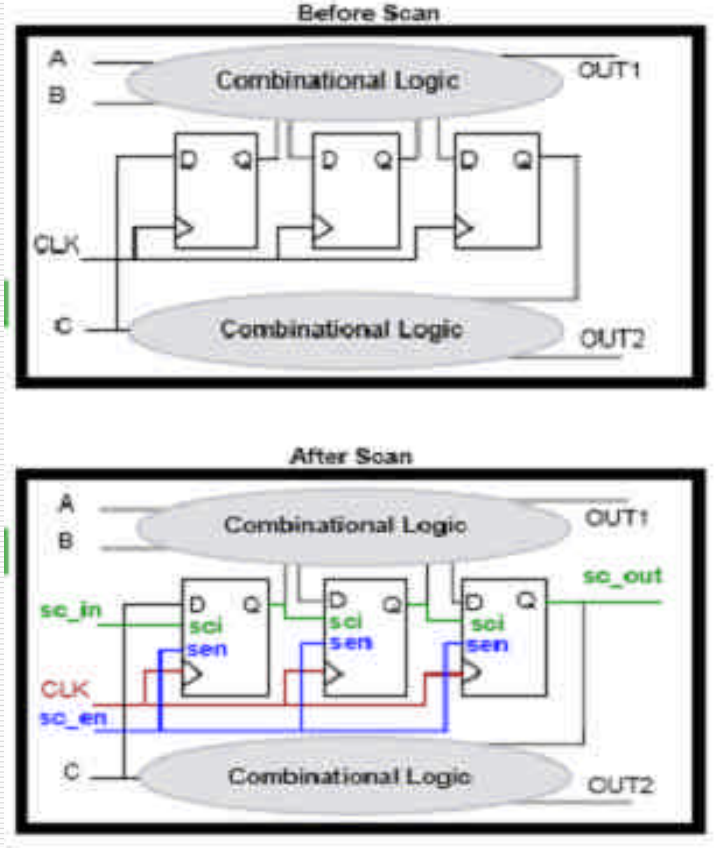
Back End Design



Scan Insertion (Design For Test)

- All internal flip-flops & latches are replaced by Scan Flip-flops (FF with MUX)
- Testability
  - Controllability
  - Observability
- Scan Pins
  - Scan In
  - Scan Out
  - Scan Enable
- scan cells are

NOT connected until placement is completed so 'chain' is not formed at this stage



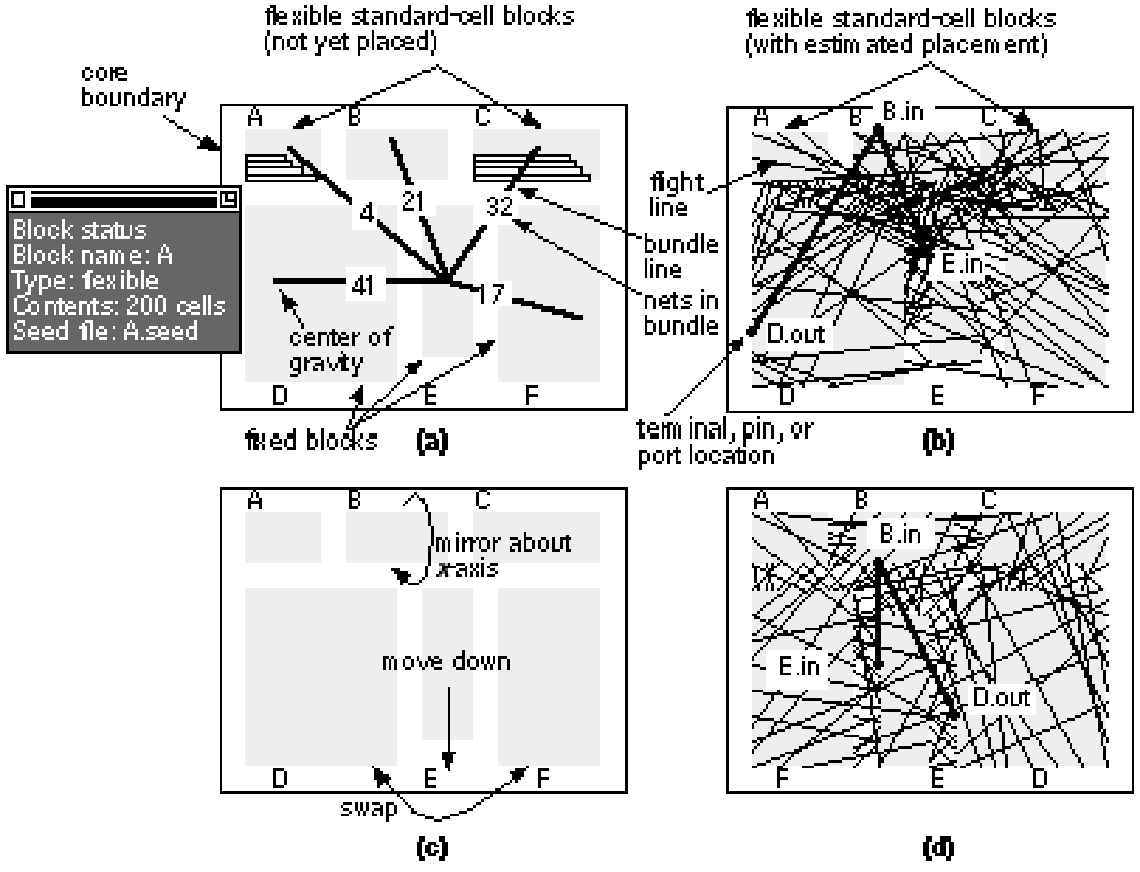
Floorplanning

- Floorplanning is a mapping between the logical description (Hierarchical Netlist) and the physical description (the floorplan).

The goals of floorplanning are to:

- arrange the blocks on a chip,
- decide the location of the I/O pads,
- decide the location and number of the power pads,
- to minimize the chip area and delay

Floorplanning



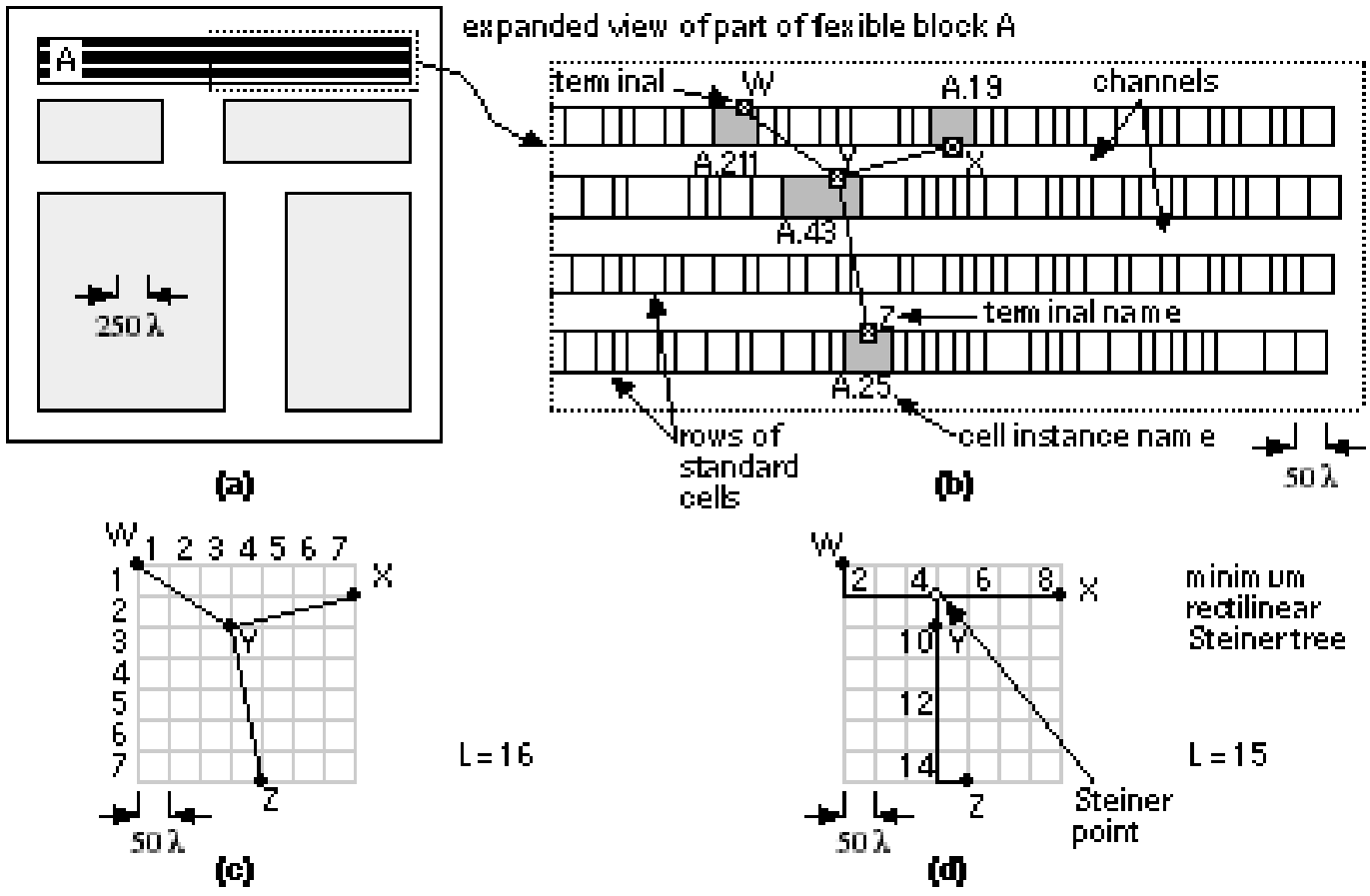
initial random floorplan generated by a floorplanning tool

Blocks are moved to reduce congestion

Placement

- Placement is arranging all the logic cells within the flexible blocks on a chip.

- objectives of placement
  - Guarantee the router can complete the routing step
  - Minimize all the critical net delays
  - Make the chip as dense as possible



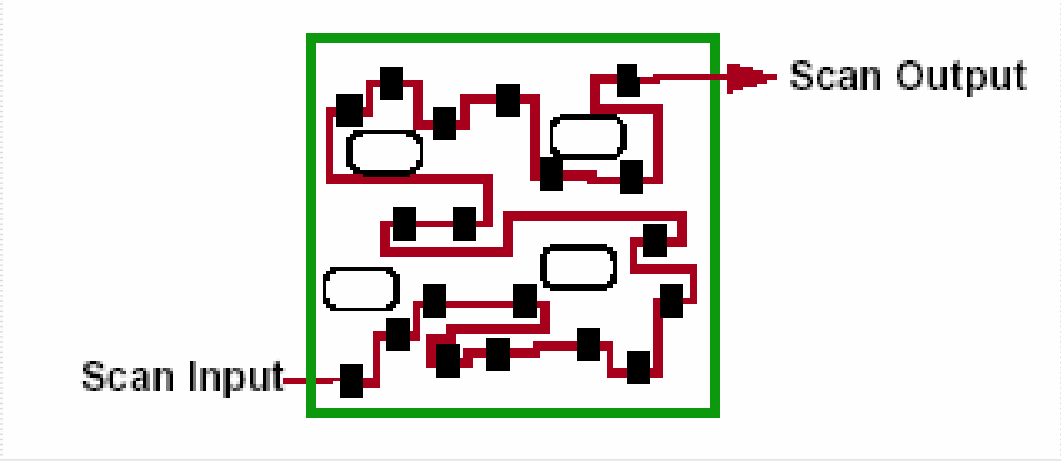
block A contains four rows of standard cells (A.211, A.19, A.43, A.25)

The Goal is to decide placement to achieve minimum distance between cells

Back

Scan chain stitching & ATPG

- After placement , Scan cells are stitched together to form a ‘scan chain’



- Normally Different scan chains are formed for different clock domain Flip Flops

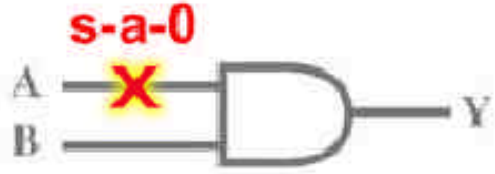
ATPG (Automated Test Pattern Generation)

- ATPG generates *Test patterns/vectors which are applied* to DUT for detection of manufacturing defects

- The goal of ATPG is to create a set of patterns that achieves a given (maximum) test coverage,

- ATPG consists of two main steps:
  - generating patterns
  - fault simulation.

Stuck at '0' fault



A	B	Y		
		Good	A s.a.0	A s.a.1
0	0	0	0	0
0	1	0	0	1
1	0	0	0	0
1	1	1	0	1

Fault models:

- **Stuck-at-fault**
- **Transition fault :** Propagation delay of cell
- **Path delay :** Sum of time delays in path
- **IDDQ :** Measurement of quiescent power supply current during the stable state

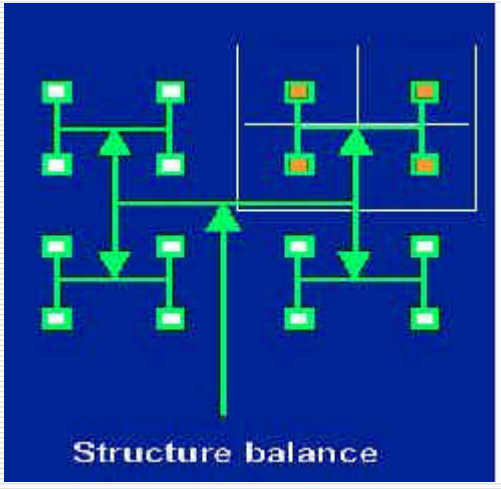
Back

Clock Tree Synthesis (CTS)

- Clock Tree is defined by its startpoint (source) and endpoints (sinks)
- During CTS, delay and skew from source to sinks are optimized.

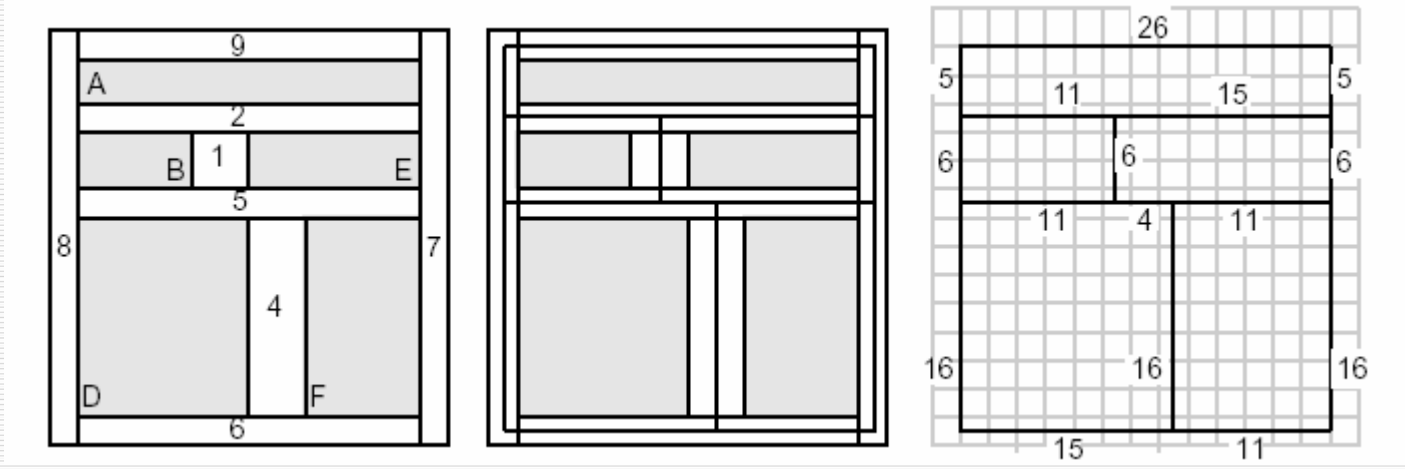
Step 1: Generate a clock tree

Step 2: Tune the clock tree to meet Skew & Slew target



Routing

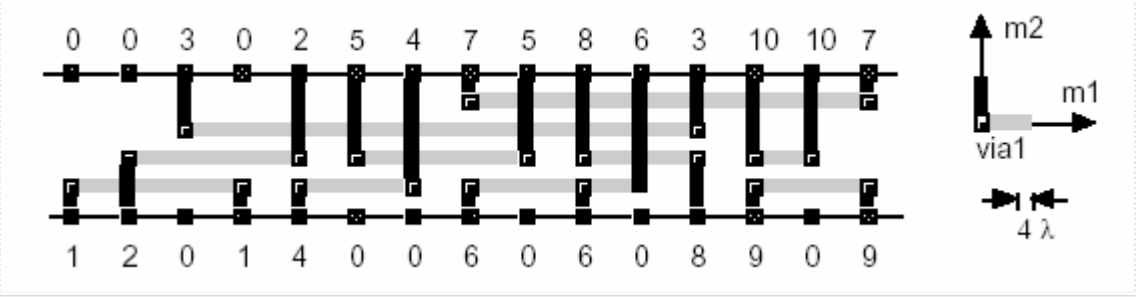
- Routing is done in 2 steps
  - **Global Routing** : plans channels for routing between blocks, Its goal are:
    - Minimize the total interconnect length.
    - Maximize the probability that the detailed router can complete the routing.
    - Minimize the critical path delay.



Global routing for a cell-based ASIC formulated as a graph problem.

Detailed Routing :

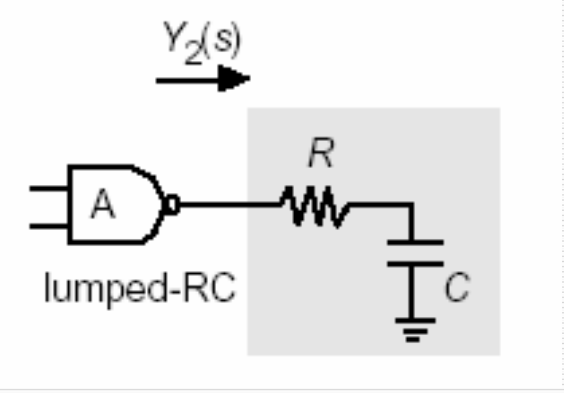
- complete all the connections between logic cells
- exact location and layers for each interconnect are determined



Completed channel route (2 metal layers m1 and m).

Back Annotation & RC Extraction

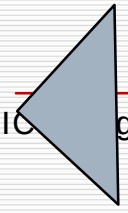
- Delays are extracted from physical & RC information in Standard Delay Format (SDF)
- Back annotated SDF file is used during post layout timing simulation and STA.



The lumped-RC interconnect model.

Formal Verification

- Equivalence check between pre-layout and post layout
- Mathematical models are made to check functionality equivalence at each node of netlist
- FV can also be done between RTL & Netlist
- EDA Tool
  - Formal Pro (Mentor)
  - Formality (synopsis)



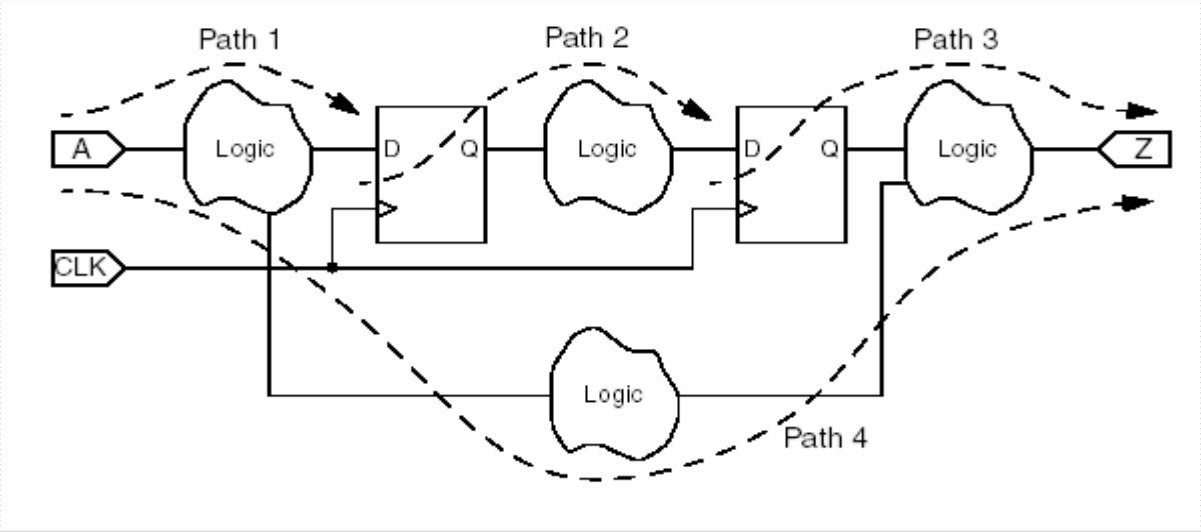
Back

STA (Static Timing Analysis)

- Static timing analysis is a method of validating the timing performance of a design by checking all possible paths for timing violations.
- STA tool breaks the design down into a set of timing paths, calculates the signal propagation delay along each path

- Compared with dynamic simulation, STA is much faster because it is not necessary to simulate the logical operation of the circuit.

STA - Timing Paths



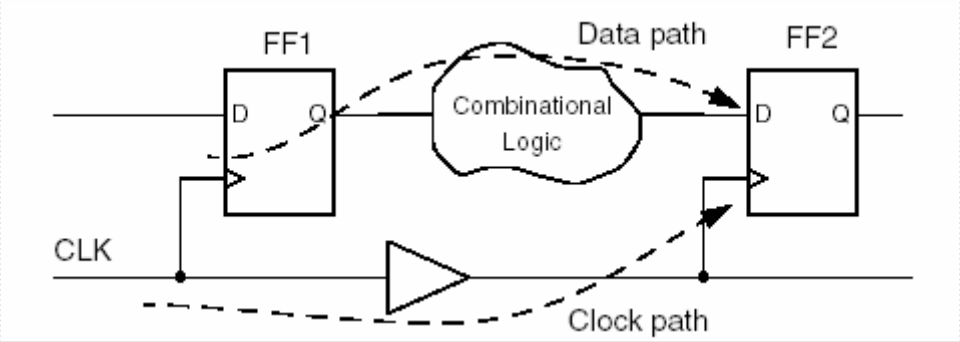
Timing Paths :

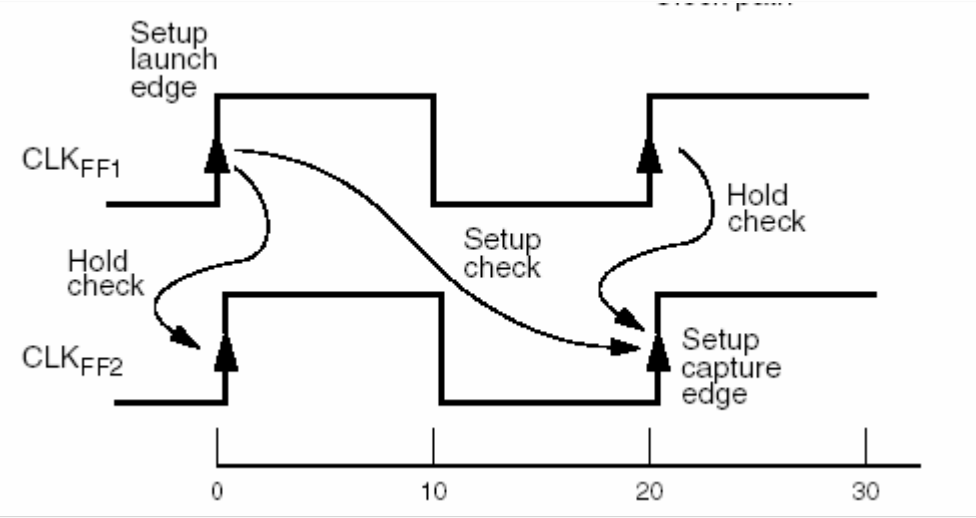
- Input path (I/p pad to FF)
- Sequential path (FF to FF)
- Output path (FF to o/p)
- Combination path (i/p to o)

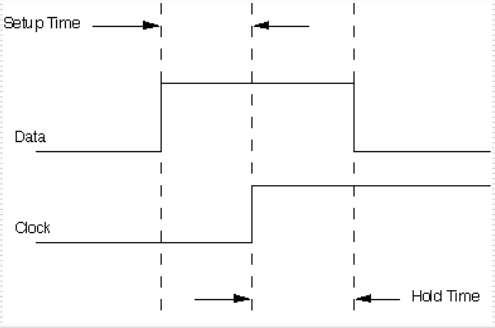
Post Layout Timing Simulation

- Back annotated SDF with post layout netlist is simulated at min, typ and max condition
- All interface timing (Ex. ROM/RAM access timing, PCI bus timing etc) should be modeled as Bus Functional Model (BFM)
- The simulation should be free from all Setup and hold violation
- Whenever data is crossing clock domain, metastable conditions should be checked.

Setup and Hold Checking

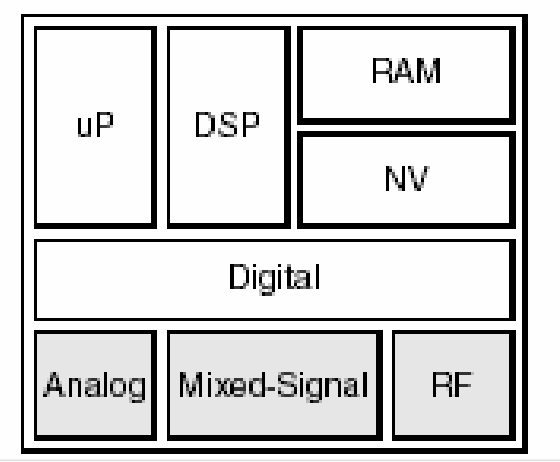




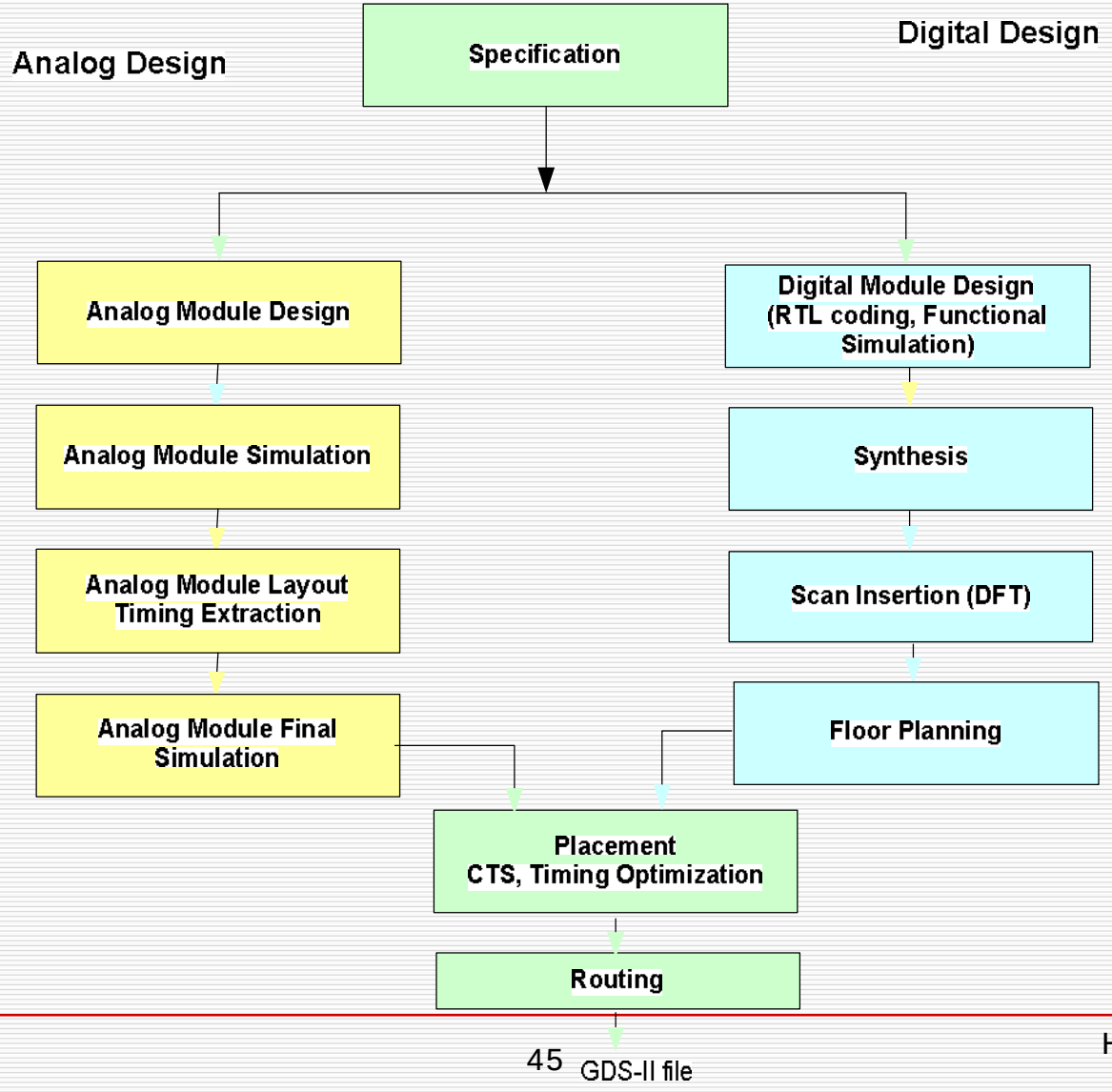


Mixed Signal ASIC

- Digital + Analog blocks
- Analog blocks
  - ADC, DAC
  - Amplifiers, comparators etc
  - RF Modulators & Demodulators

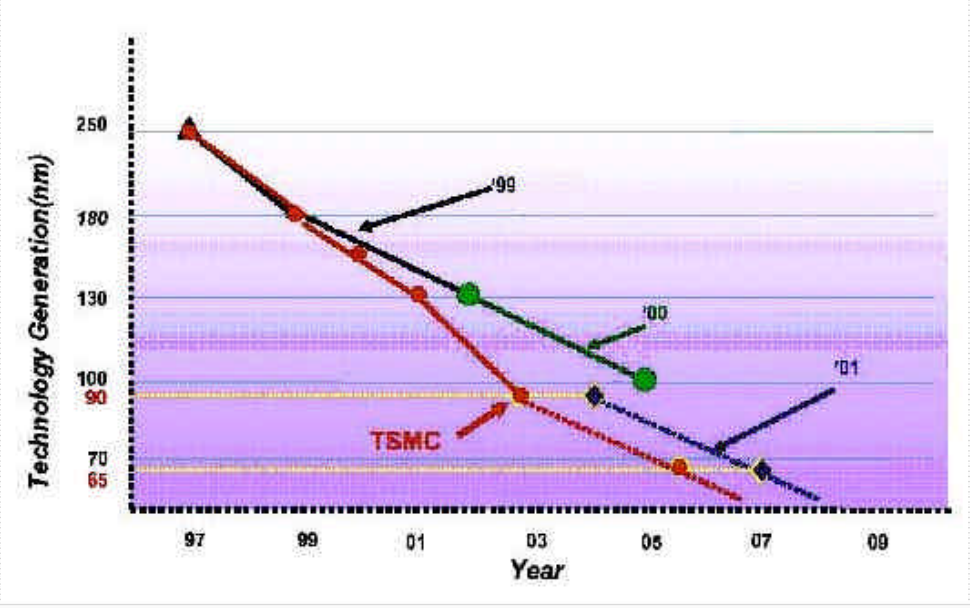


Mixed Signal ASIC Design Flow



Deep Submicron ASIC : Challenges

- As technology advances towards submicron (below 0.13um/0.9um) issues like signal integrity, power etc become prominent



Deep Submicron issues

- **Power**
  - Increased DC power due to leakage current
- **Signal Integrity**
  - Lower supply voltage reduces noise margin
  - Smaller geometries induces coupling noise
  - Higher current density causes EM issues
- **Design Complexity**
  - Transistor Density doubles every 18 months (moor's law)
  - Clock frequency increases above 5 GHz

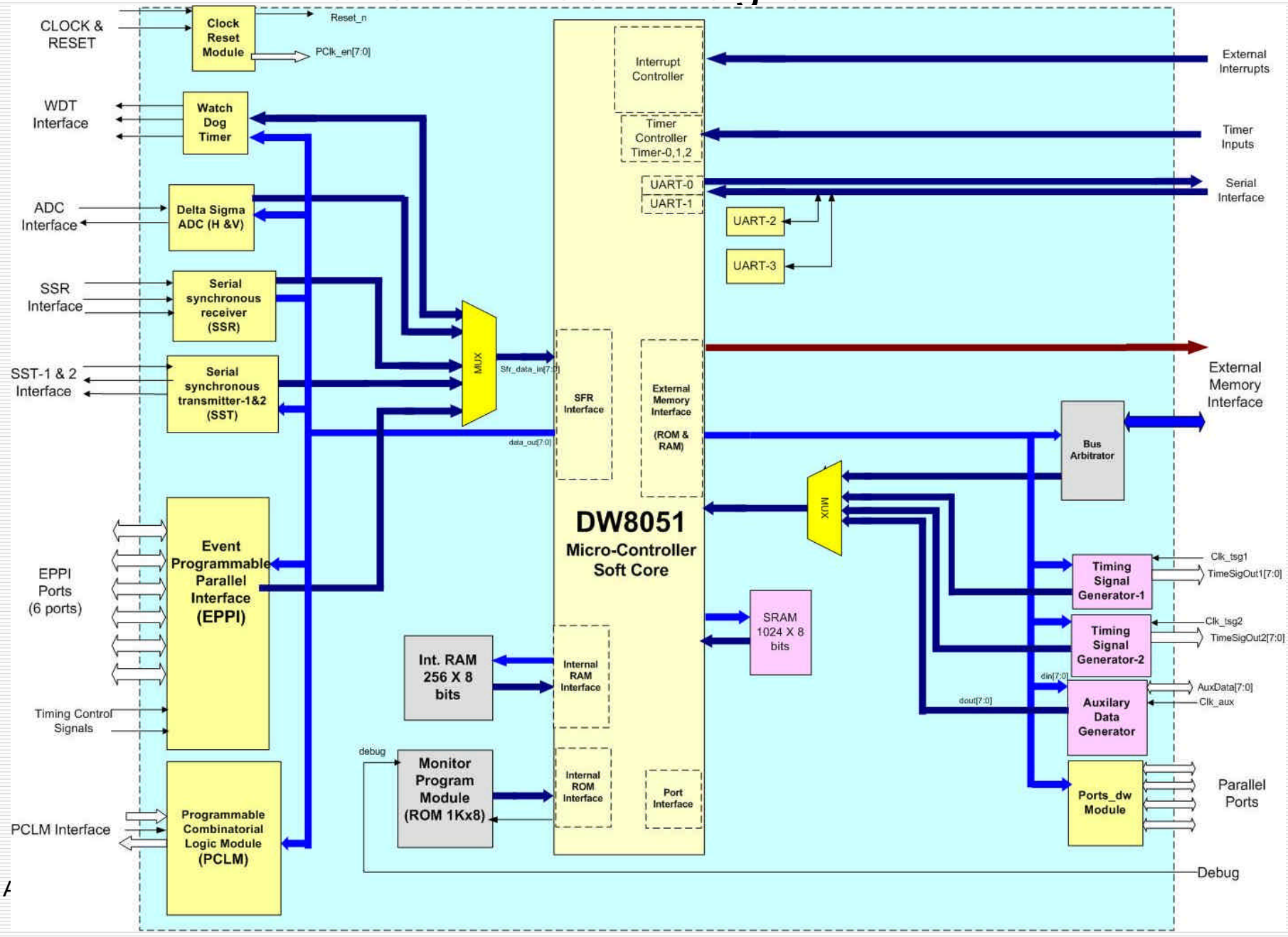
OBC ASIC : A Case Study

- OBC (On Board Controller) ASIC is designed for different ISRO's Satellite missions having on board Distributed controllers

- OBC ASIC Features
  - On chip 8051 micro controller soft core
  - 4 UARTs, 3 Timers, 6 Interrupts
  - 3 synchronous serial receivers & transmitters
  - 10 ports for parallel I/Os
  - On chip 1KB ROM containing monitor program
  - 16 programmable timing signal generator

- ASIC Features
  - CMOS Gate Array ASIC
  - 256 pin package , 224 user I/Os
  - 5 V core & 5V I/Os
  - Radiation Hardened process
  - Testability features like SCAN and ATPG with logic Fault Coverage of > 95%

OBC Block Diagram

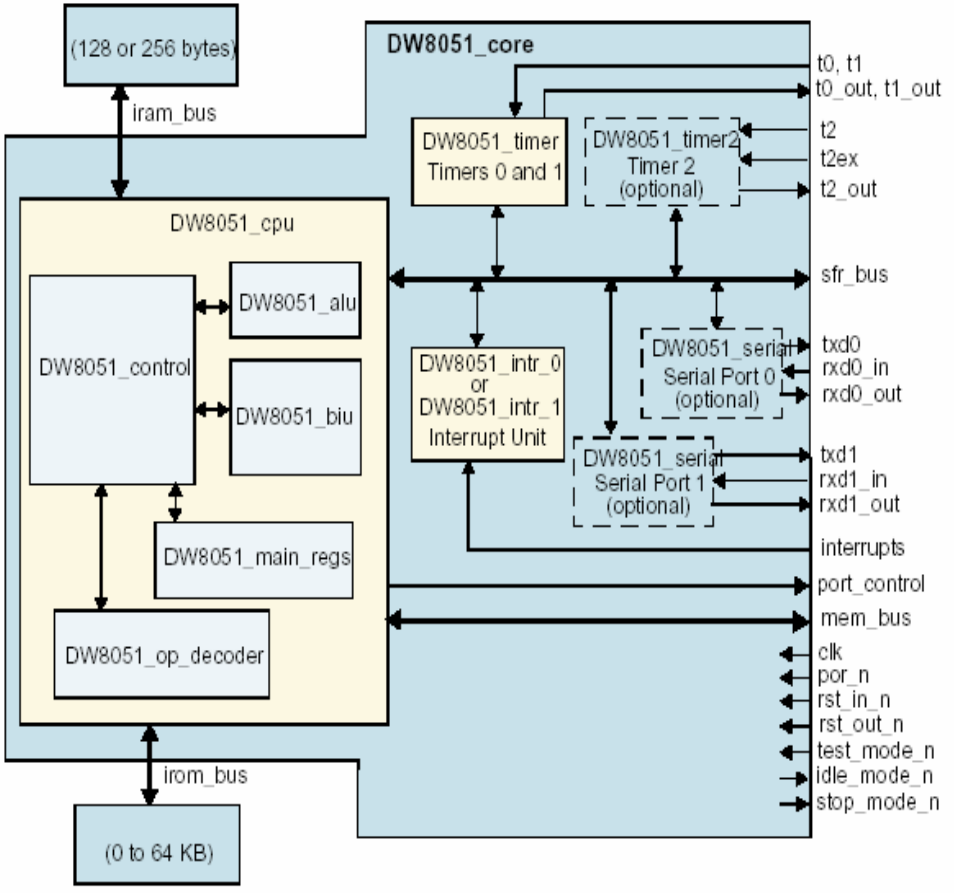


DW8051

Synopsys® DesignWare 8051 Soft IP Core

Features

- **High–speed architecture :4** clocks per instruction cycle 2.5X improvement over the standard 8051
- Dual data pointers
- 3 Timers, 2 UARTs
- Extended Interrupts (7 nos)
- Variable length MOVX to access fast/slow RAM peripherals
- Fully static synchronous design

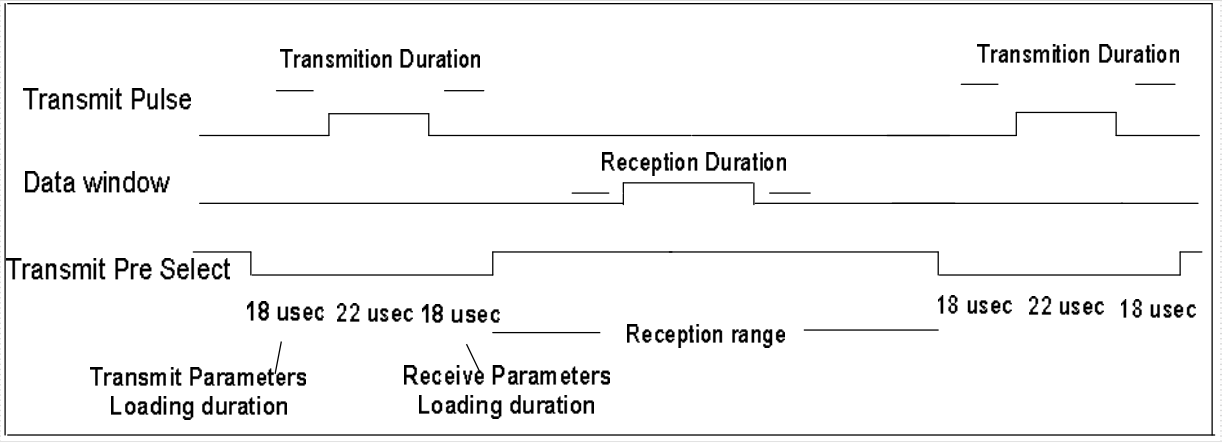


OBC Peripheral Modules

- **Watch Dog Timer** : generates reset/interrupt whenever the software hangs
- **Delta Sigma ADC:** 8 bit digitization of low-rate analog data
- **Programmable Combinatorial Logic Module:** A small FPGA CLB inside ASIC...!

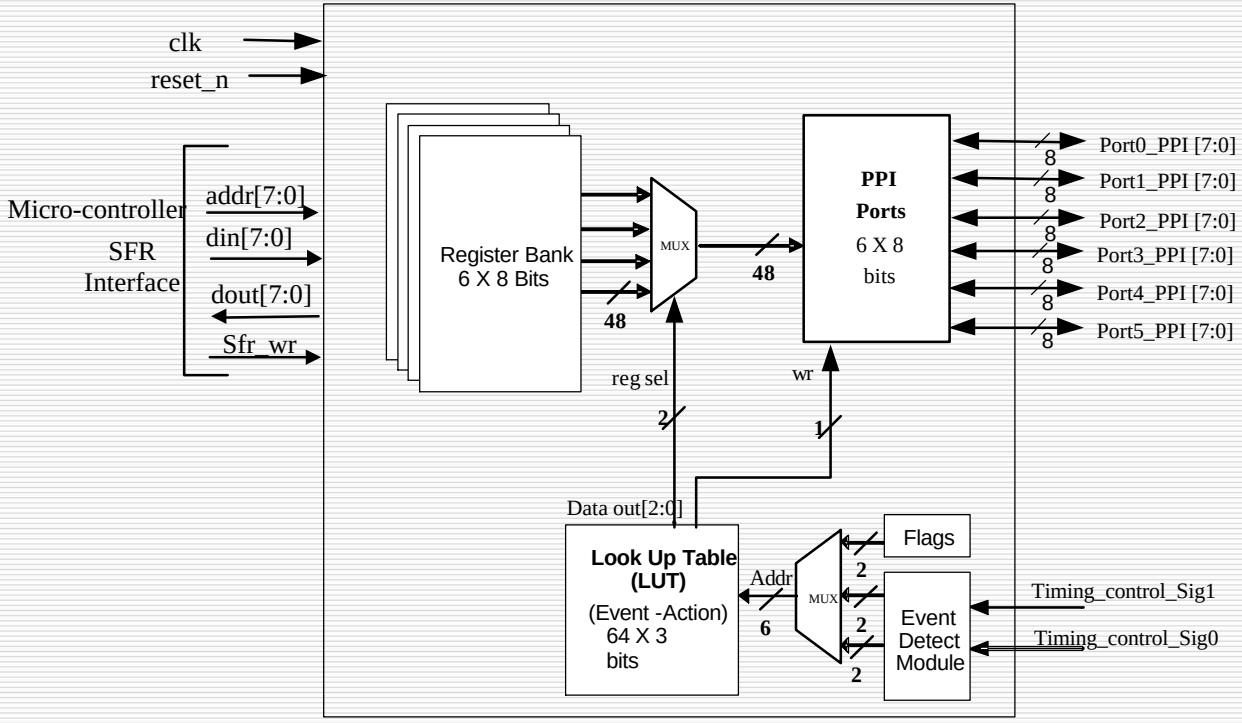
- **Timing Signal Generator:** This is generic timing signal generator, which can generate up to 16 programmable pulses
- **Auxiliary Data Interface:** This is parallel/serial auxiliary interface with built in dual port RAM.
- **Serial Synchronous Transmitter/Receiver** This is 3-wire (clock, strobe, data) synchronous tx/rx
- **Monitor Program :**OBC ASIC contains 1K Bytes of on chip ROM which holds Monitor program firmware.

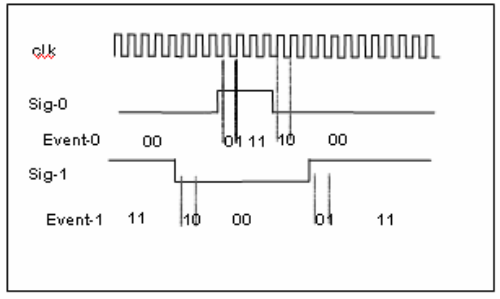
Event Programmable Parallel Interface



- In Phased array distributed controller it is required to load Transmit & Receive characterization data within time constraint as shown above
- This task was earlier implemented in software as "Interrupt Service Routine", but due to variable interrupt latency it was not meeting timing constraint
- So a Hardware module was implemented

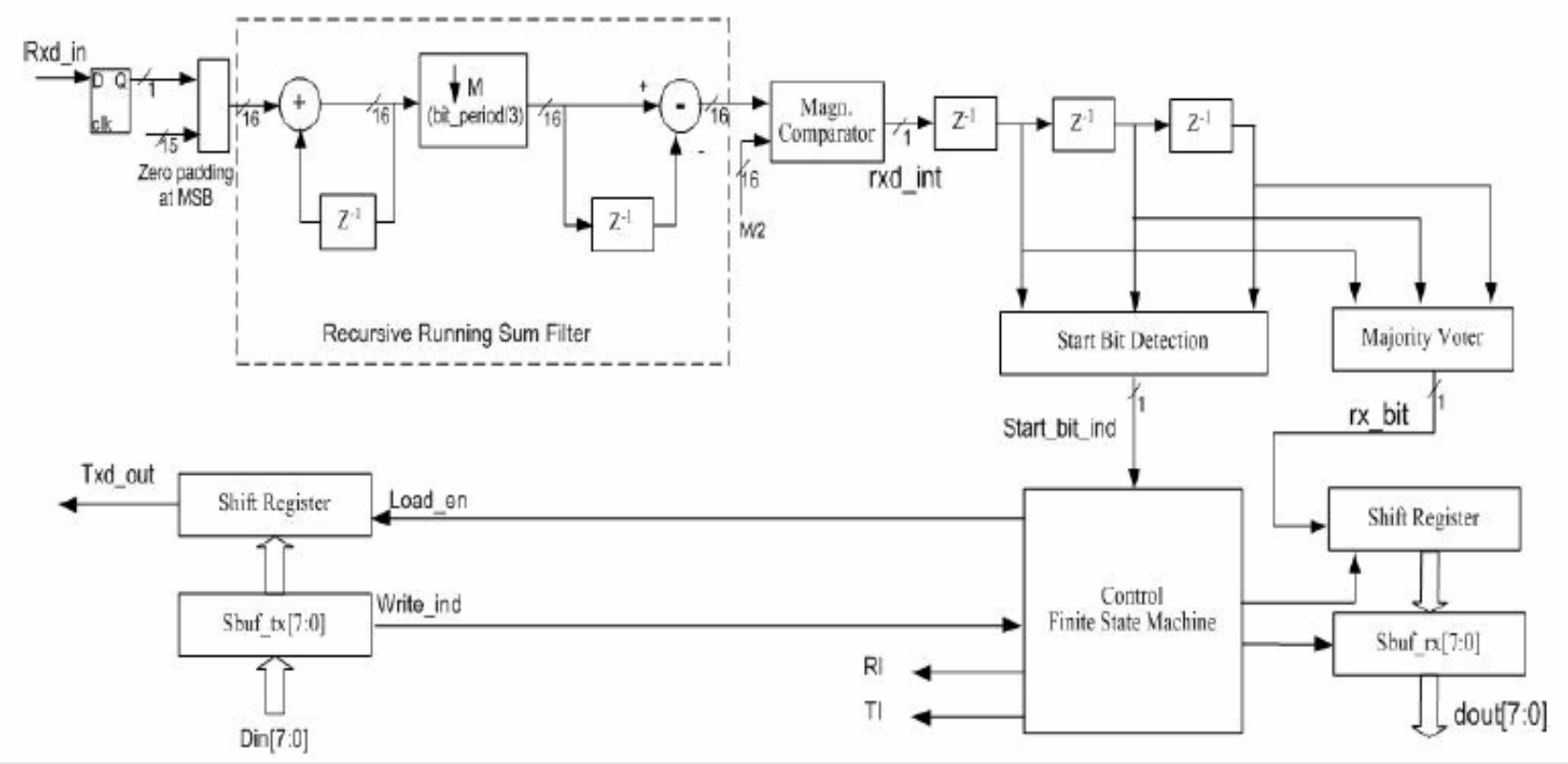
EEPPI - Architectute





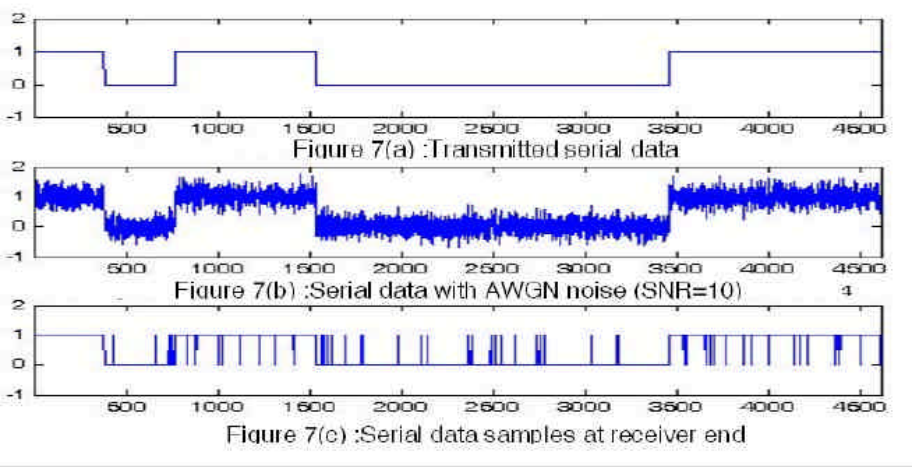
samples S(n-1),S(n)	Event
0,0	Level-'0' (low)
0,1	Rising event
1,0	Falling event
1,1	Level-'1' (high)

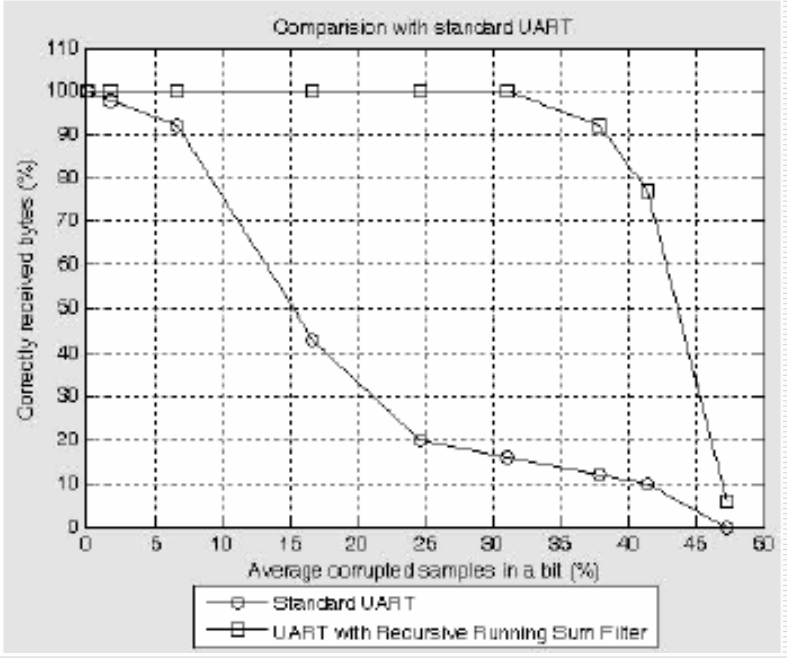
UART_RRS



- UART with Recursive Running Sum Filter to remove noise samples from incoming serial data

UART_RRS Test Results



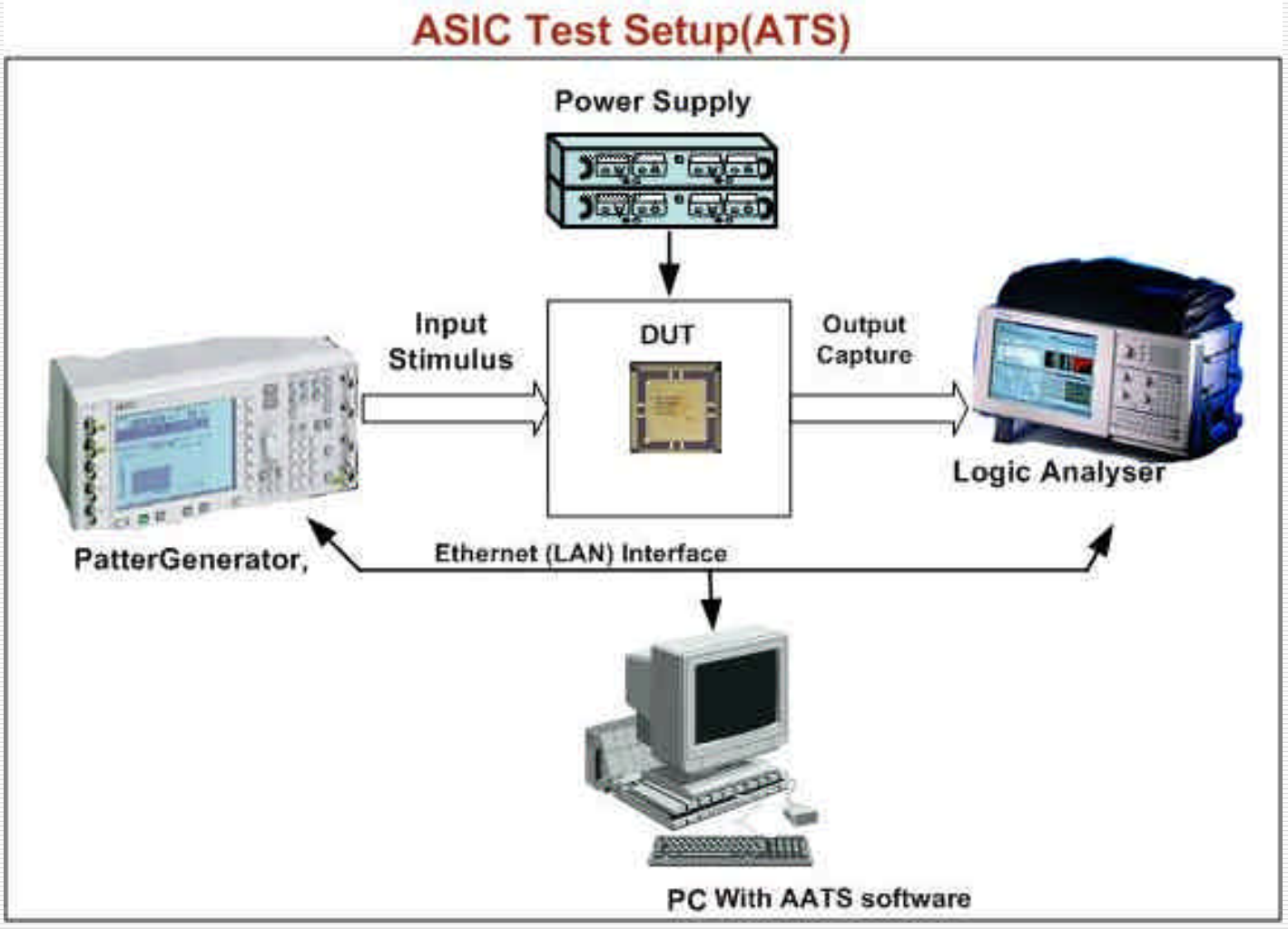


- UART_RRS has better performance than standard UART at higher Noise levels
  - UART_RRS can decode data correctly up to **37 %** corrupted sample
  - Standard UART can decode data up to **6 %** only

Challenges Faced during OBC design

- Almost all OBC modules are programmable/ configurable because exact funtionality was not freezed during front-end design.
- Multi Clock Domain
  - OBC contains 7 clock domains so 7 Scan Chains were inserted.
- Data crossing clock domains
  - Asynchronous DPRAM, with Left and Right Ports accessing data at different clocks
  - Series of FFs to avoid Metastable condition
- Rad Hard cells
  - All Flip flops used in OBC are radhard (ASIC library contained Soft, Rad Tol. & Rad hard)
- Monitor Program
  - On chip 1K ROM contains assembly software , which was regorously tested as lateron software modification was not possible

Proto ASIC Testing



Thank You

To Probe Further :

http://Geocities.com/hnpatel81/asic.htm