VLSI System Design (CS60067)

Dipanwita Roy Chowdhury
Professor
Dept. of Computer Sc. & Engg.
IIT Kharagpur

Course Information

Instructor: Dipanwita Roy Chowhury

drc@cse.iitkgp.ac.in

Course TA : Tapadyoti Banerjee

tapadyoti@iitkgp.ac.in

Books:

1. **Neil H. E. Weste and David Money Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, Pearson**
2. **Michael John Sevastian Smith, Application-Specific Integrated Circuits, Pearson**

Exams and Evaluation

- **Two tests, Mid sem and End sem.**
- **Two class tests / One class test + Quiz/ One class test + Assignments.**
- **Mid sem – 30%, End sem – 50%,**

  class tests / class test + Quiz / class test + Assignments – 20%

Course Goals

- **Learn to design and analyze state-of-the-art digital VLSI chips using CMOS technology**
- **Employ hierarchical design methods**
  - Understand design issues at the layout, transistor, logic and register-transfer levels
  - Use integrated circuit cells as building blocks
  - Use commercial design software in the lab
- **Understand the complete design flow**
  - Not cover solid-state physics, analog design
  - Superficial treatment of transistor functioning

Digital VLSI Design Process – the general principle

- MOS technology trends are largely driven by miniaturization and improved performance in integrated circuits.

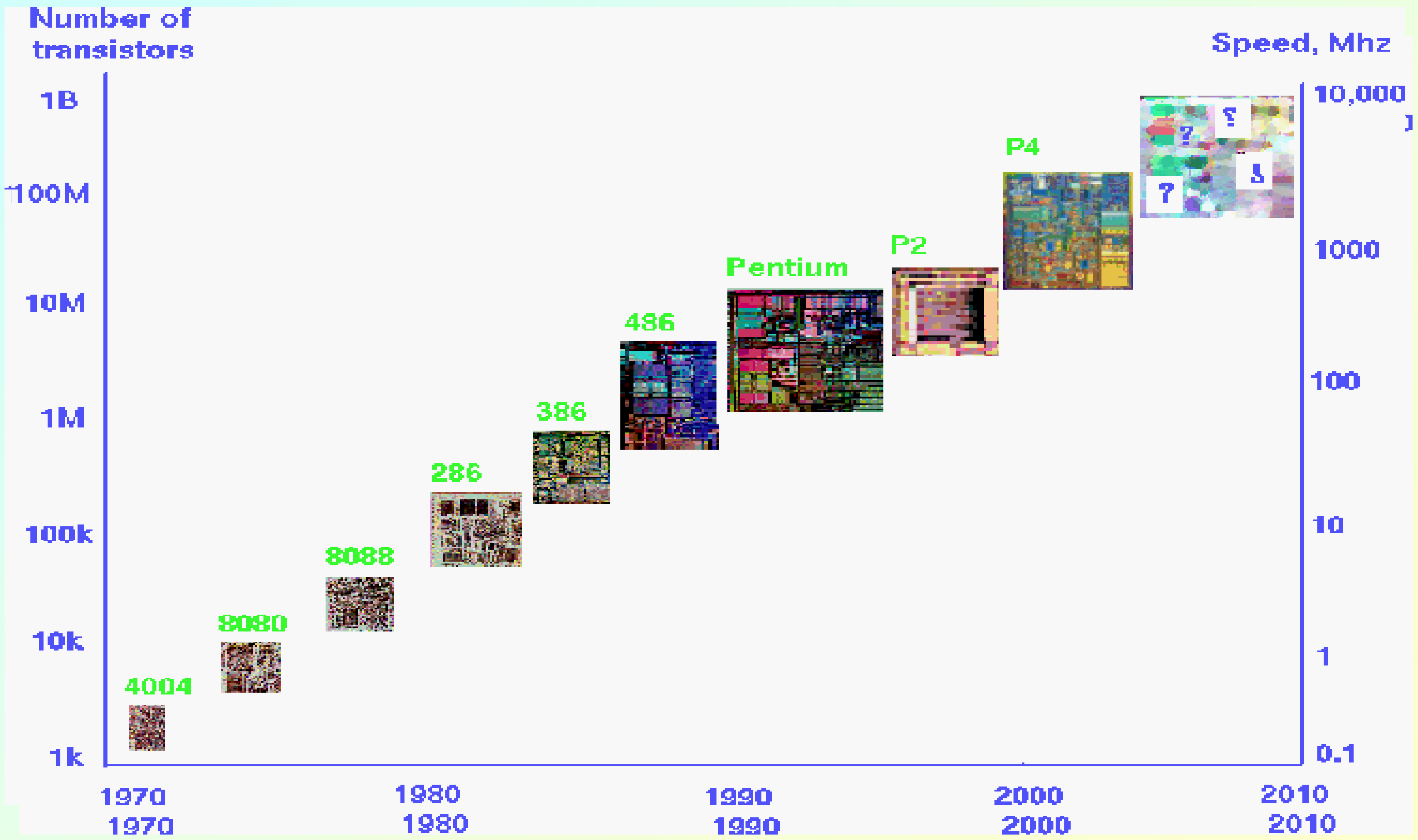
- Complementary Metal-Oxide-Semiconductor (CMOS) technology continues to dominate, offering a balance of low power consumption and high performance.

- Other key trends include the development of new materials, more complex device structures, and advanced fabrication techniques.

ion

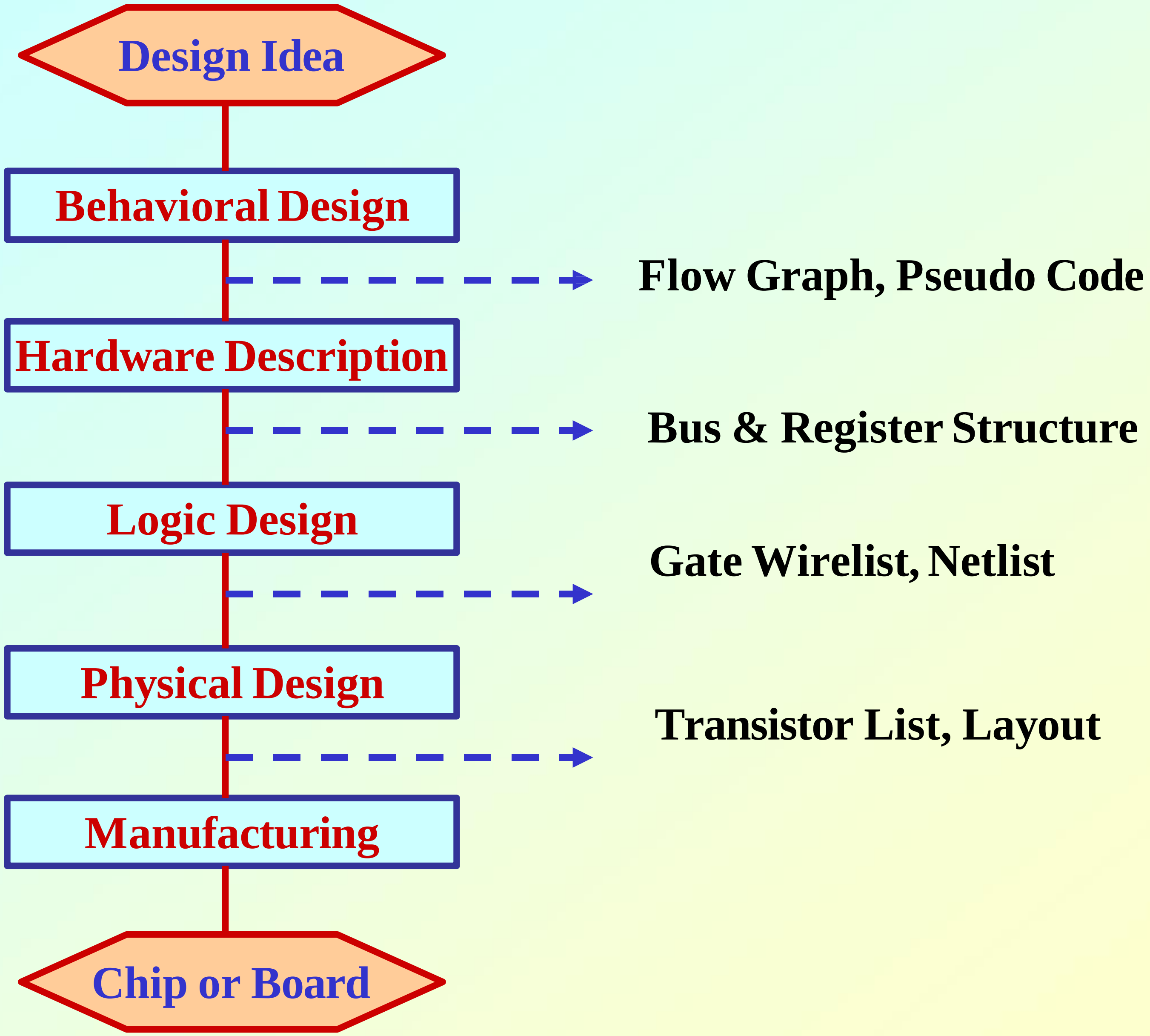
MOS Technology Trends

https://users.ece.utexas.edu~adnan>vlsi-07



Digital design process – the current trend

- CAD tools are introduced in the process due to its increased size and complexity.
- Designers want to standardize the design procedure, starting from the design idea to get the design implemented – the so called design flow.
- New CAD tools are based on Hardware description language(HDL) to improve the process.
- HDLs provide formats for representing the outputs of various design steps. An HDL based design automation(DA) tool transforms for its HDL input into an HDL which contains more hardware information.


Design Idea
Behavioral Design
Flow Graph, Pseudo Code
Hardware Description
Bus & Register Structure
Logic Design
Gate Wirelist, Netlist
Physical Design
Transistor List, Layout
Manufacturing
Chip or Board

Behavioral Representation

- Behavioral representation describes how a particular design should respond to a given set of inputs.

Behavior may be specified by

- boolean equations
- tables of input and output values
- algorithms written in standard high level computer language
- algorithms written in special hardware description language

Behavioral representation -- example

- A n-bit adder is constructed by cascading n 1-bit adders.

A 1-bit adder has

- two operand inputs A and B
- a carry input C
- a carry output CO
- a sum output S

The boolean equations that implement this function

$S = A.\sim B.\sim C + \sim A.\sim B.C + \sim A.\sim C.B + A.B.C$

$CO = A.B + A.C + B.C$

An algorithmic level description of CO looks like

```
module  carry (co, a, b, c);
        output co;
        input a, b, c;
        assign
          co = (a&b) | (a&c) | (b&c);
 endmodule
```

- **To specify the timing behavior of the gate**

```
module  carry (co, a, b, c);
        output co;
        input a, b, c;
        wire  # 10  co = (a&b) | (a&c) | (b&c);
 endmodule
```

Boolean behavioral specification for CO

```
primitive  carry (co, a, b, c);
        output co;
        input a, b, c;
        table
        //  a  b  c     co
•             1  1  ?  :  1 ;
              1  ?  1  :  1 ;
              ?  1  1  :  1 ;
              0  0  ?  :  0 ;
              0  ?  0  :  0 ;
              ?  0  0  :  0 ;
      endtable
   endprimitive
```

Structural Representation

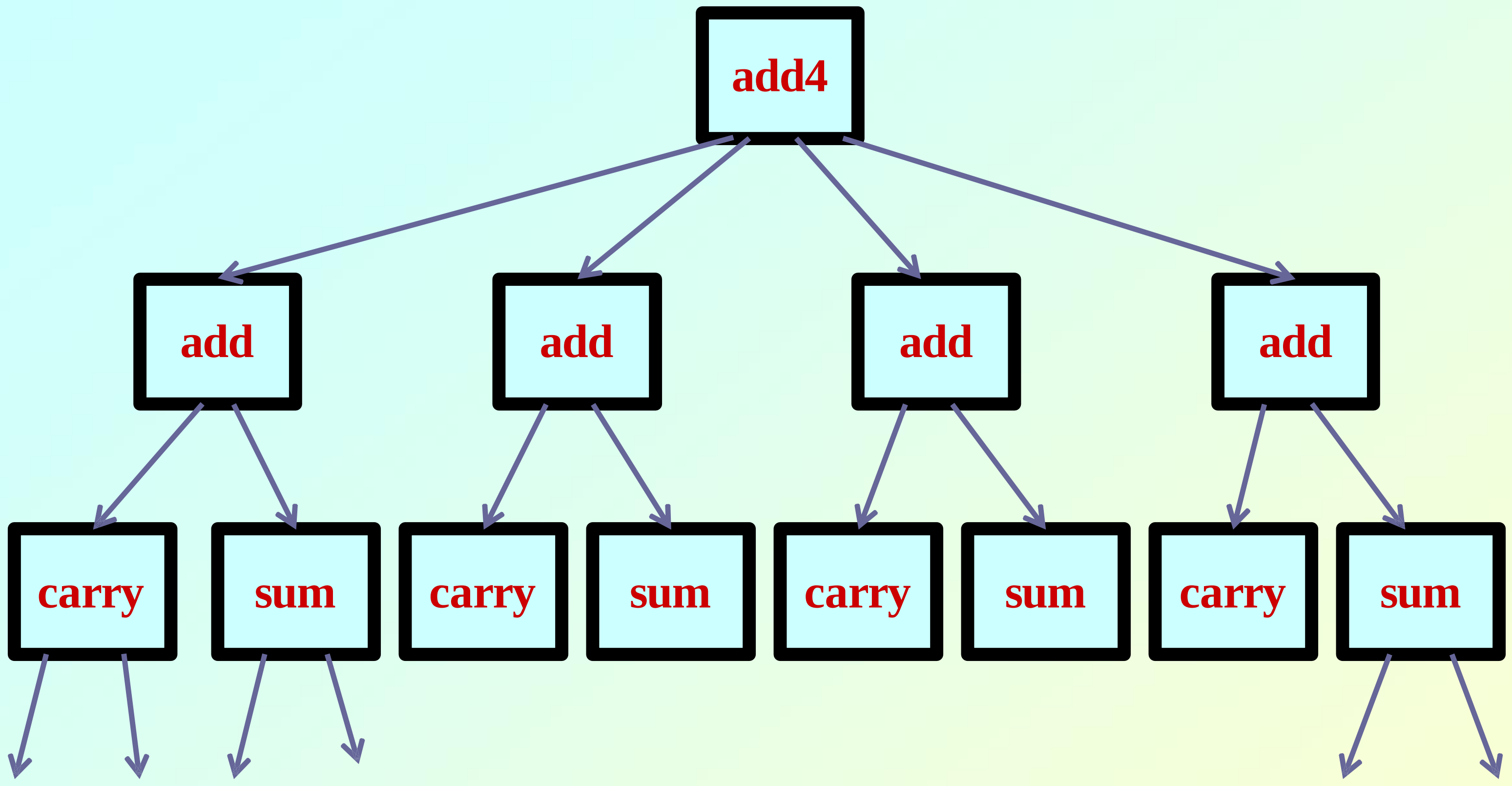
A structural representation specifies how components are interconnected to perform a certain function or achieve a designated behavior.

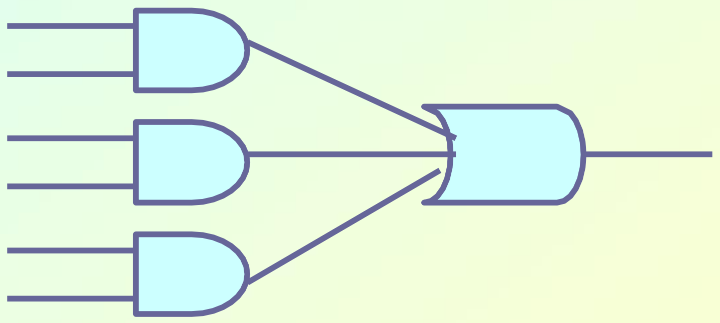
In general, the description is a list of modules and their interconnects – often called netlist.

At the structural level the levels of abstraction are

- the module level
- the gate level
- the switch level
- the circuit level

In each level more detail is revealed about the implementation.


add4
add
add
add
add
carry
sum
carry
sum
carry
sum
carry
sum



Structural representation – example

4-bit adder

```
module  add4 (s, c4, ci, a, b);
        input [3:0] a, b;
        input ci;
        output [3:0] s;
        output c4;
        wire [2:0]  co;
              add   a0  (co[0], s[0], a[0], b[0], ci);
              add   a1  (co[1], s[1], a[1], b[1], co[0]);
              add   a2  (co[2], s[2], a[2], b[2], co[1]);
              add   a3  (c4, s[3], a[3], b[3], co[2]);
endmodule
```

```
module  add  (co, s, a, b, c);
     input a, b, c;
     output  s, co;
           sum s1 (s, a, b, c);
            carry  c1 (co, a, b, c);
 endmodule



module carry (co, a, b, c);
       input a, b, c;
       output  co;
       wire   x, y, z;
             and  g1 (x, a, b);
             and  g2 (y, a, c);
             and  g3 (z, b, c);
             or     g4 (co, x, y, z);
  endmodule
```

Physical Representation

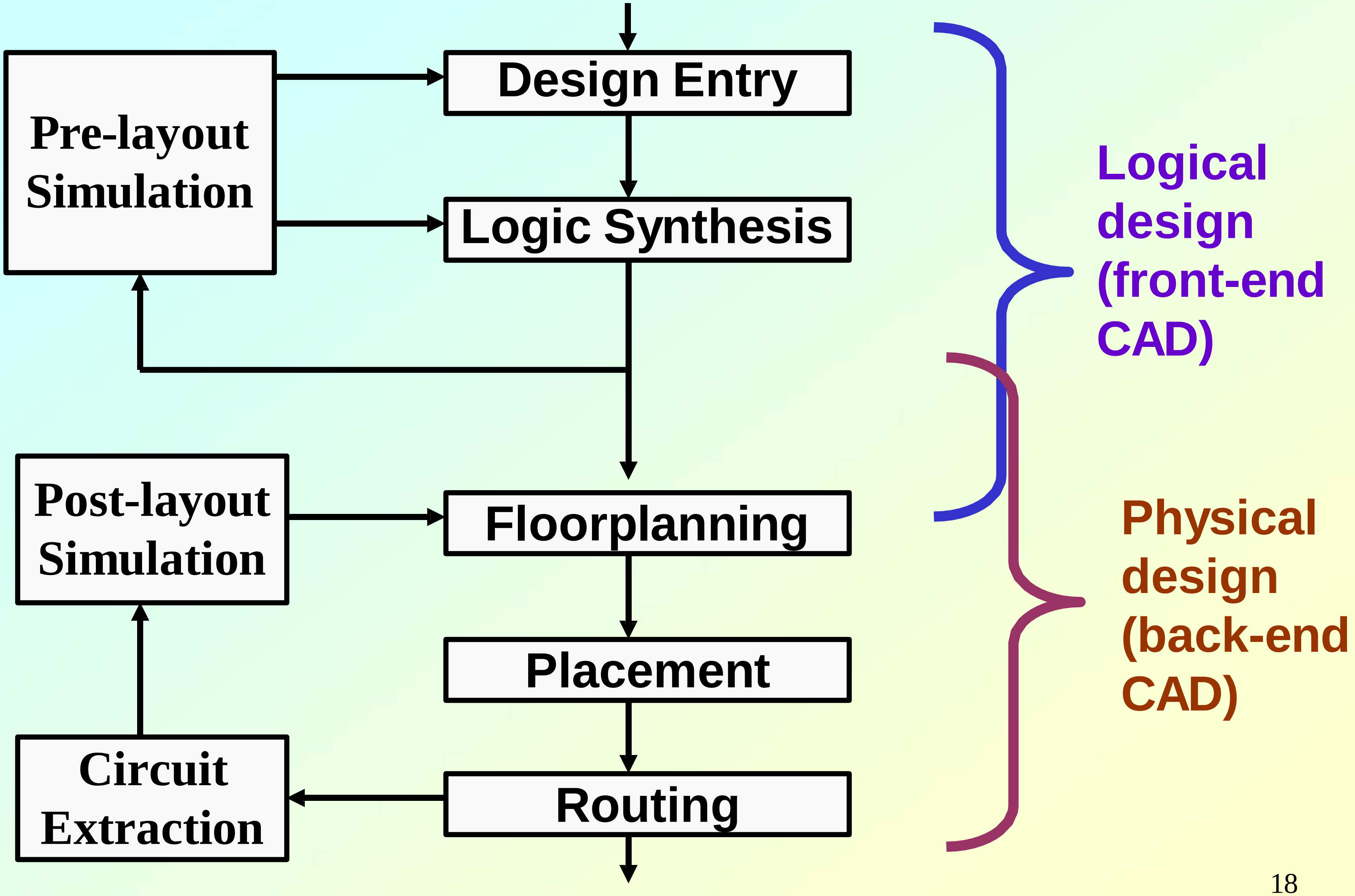
In an IC process , the lowest level of physical specification is the photo-mask information required by the various processing steps in the fabrication process.

At the module level, the physical layout for the 4-bit adder may be defined by a rectangle or polygon that specifies the outer boundary of all the geometry for the adder, set of submodules and collection of ports.

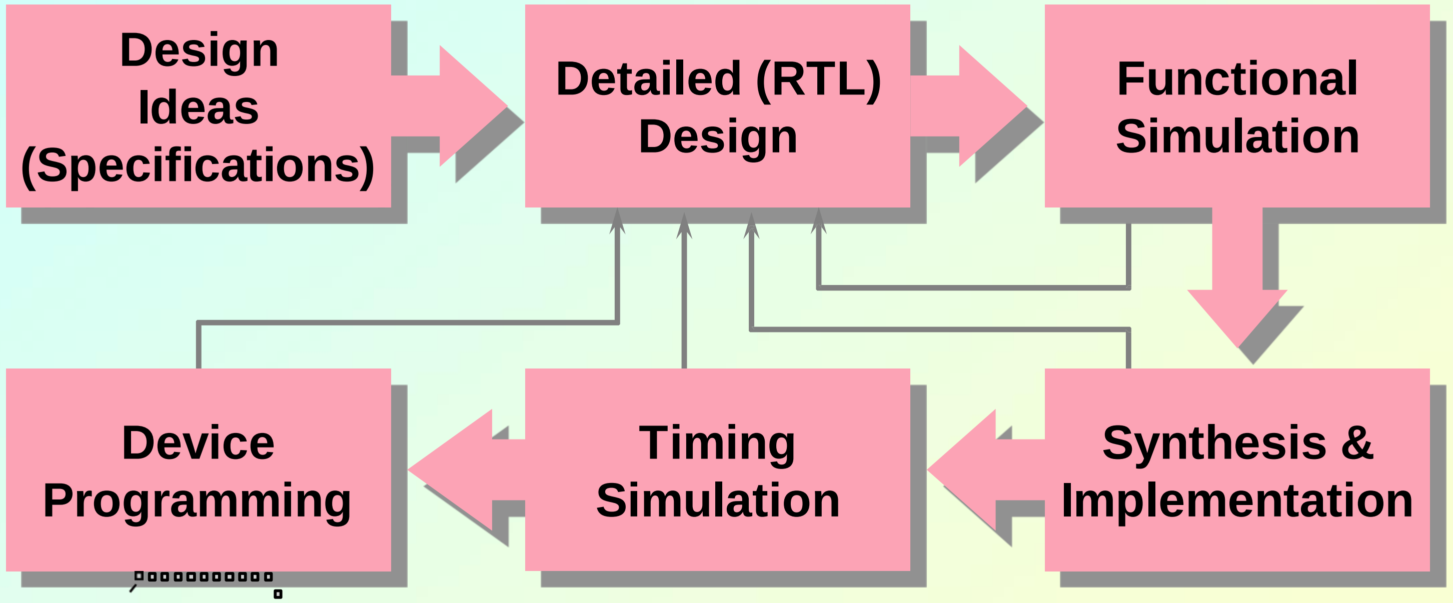
Each port corresponds to an I/O connection in the structural description.

The position, layer and width are also specified for each port.

ASIC Design Flow



FPGA Design Flow



Design Specification

- What are the main design considerations?
  - Design feasibility?
    - Performance
    - power consumption
    - cost
  - Design spec?
    - Written (Document)
    - Executable (C/C++, Behavioral VHDL, SystemVerilog)
  - Implementation platform
    - FPGA/ASIC ?
    - Which FPGA/CPLD vendor?
    - Which device family?
  - Development time?

RTL Specification

- Determine I/O signals
  - Standard interface, protocol, custom interface
- Partition design into functional blocks
  - Datapath, Control logic, Memory, etc.
- Determine block interfaces
- Specify each block separately at RTL
  - MUXs, counters, adders, flip-flops etc.

Detailed Design

- Choose the design entry method
  - Schematic
    - Intuitive & easy to debug
    - Not portable
  - HDL (Hardware Description Language), e.g. Verilog, VHDL, SystemC
    - Requires some experience, harder to debug
    - Descriptive & portable
    - Easy to modify
  - Mixed HDL & schematic
- Interpret the specifications
- Manage the design hierarchy
  - Design partitioning
    - Chip partitioning
    - Logic partitioning
  - Use vendor-supplied IP libraries to reduce design time
  - Create & manage user-created libraries

Functional Simulation

- Preparation for simulation
  - Generate simulation patterns
    - Waveform entry
    - HDL testbench
  - Generate simulation netlist
- Functional simulation
  - To verify the functionality of your design only
- Simulation results
  - Waveform display
  - Text output
  - Self-checking testbench
- Challenge
  - Sufficient & efficient test patterns

HDL Synthesis

- Synthesis = Translation + Optimization
  - Translate HDL design files into gate-level netlist
  - Optimize according to your design constraints
    - Area constraints
    - Timing constraints
    - Power constraints

- Main challenges
  - Learn synthesizable coding style
  - Use proper design partitioning for synthesis
  - Specify reasonable design constraints
  - Use HDL synthesis tools efficiently

Design Implementation

- Implementation flow
  - Netlist merging, flattening, data base building
  - Design rule checking
  - Logic optimization
  - Block mapping & placement
  - Net routing
  - Configuration bitstream generation (FPGA only)
  - Scan flip-flop insertion (ASIC only)
- Implementation results
  - Design error or warnings
  - Device utilization (FPGA)
  - Die size (ASIC)
  - Timing reports
- Challenge
  - How to reach high performance & high utilization implementation?

Timing Simulation (optional)

- Post-layout simulation
- Includes component and wire delays, clock skew, setup and hold times
- Same input vectors with functional simulation
- Not necessary if Static Timing Analysis shows no problems

Device Programming (FPGA only)

- Choose the appropriate configuration scheme
  - SRAM-based FPGA/CPLD devices
    - Downloading the bitstream via a download cable
    - Programming onto a non-volatile memory device & attaching it on the circuit board
  - OTP, EPROM, EEPROM or Flash-based FPGA/CPLD devices
    - Using hardware programmer
- Finish the board design
- Program the device
- Challenge
  - Board design
  - System considerations

Testing (ASIC only)

- Find defects in chips that return from the foundry due to the manufacturing process
- Identify differences between specification/verification/prototyping and manufactured ASIC

Common pitfalls/important points

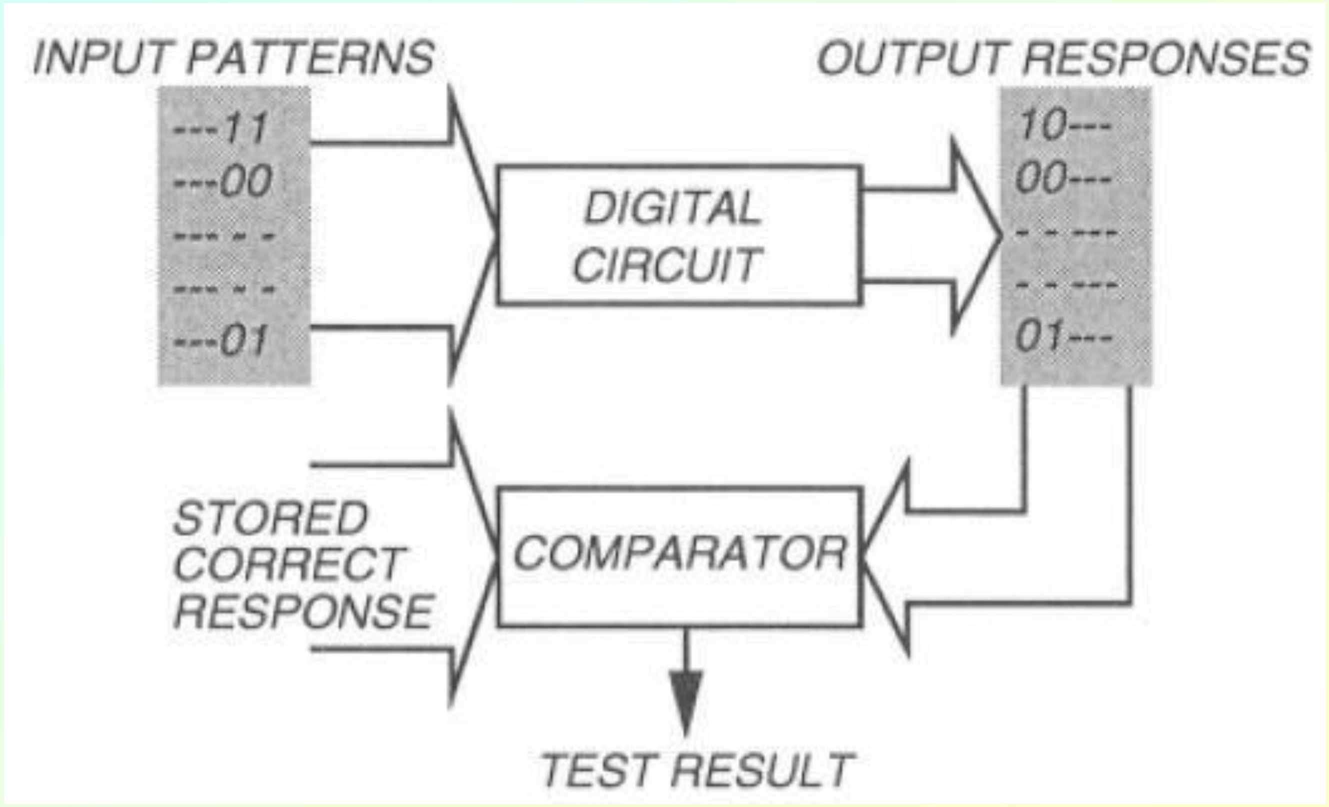
- Not detailed enough specifications
  - Do not start design entry until all details are clear
  - A poor design cannot be saved by good code and synthesis/implementation constraints
  - Sometimes though, it is better to let the tool do the optimization (for example state machines)
- Always remember that you cannot prove a complex design has no bugs
- For complex designs, verification (simulation) and redesign is 80% of total design time!

Testing Basics

Defect: A difference between intended design and actual h/w

Error: A wrong output produced through a defect

Fault: A defect in a higher abstraction level



Controllability and observability

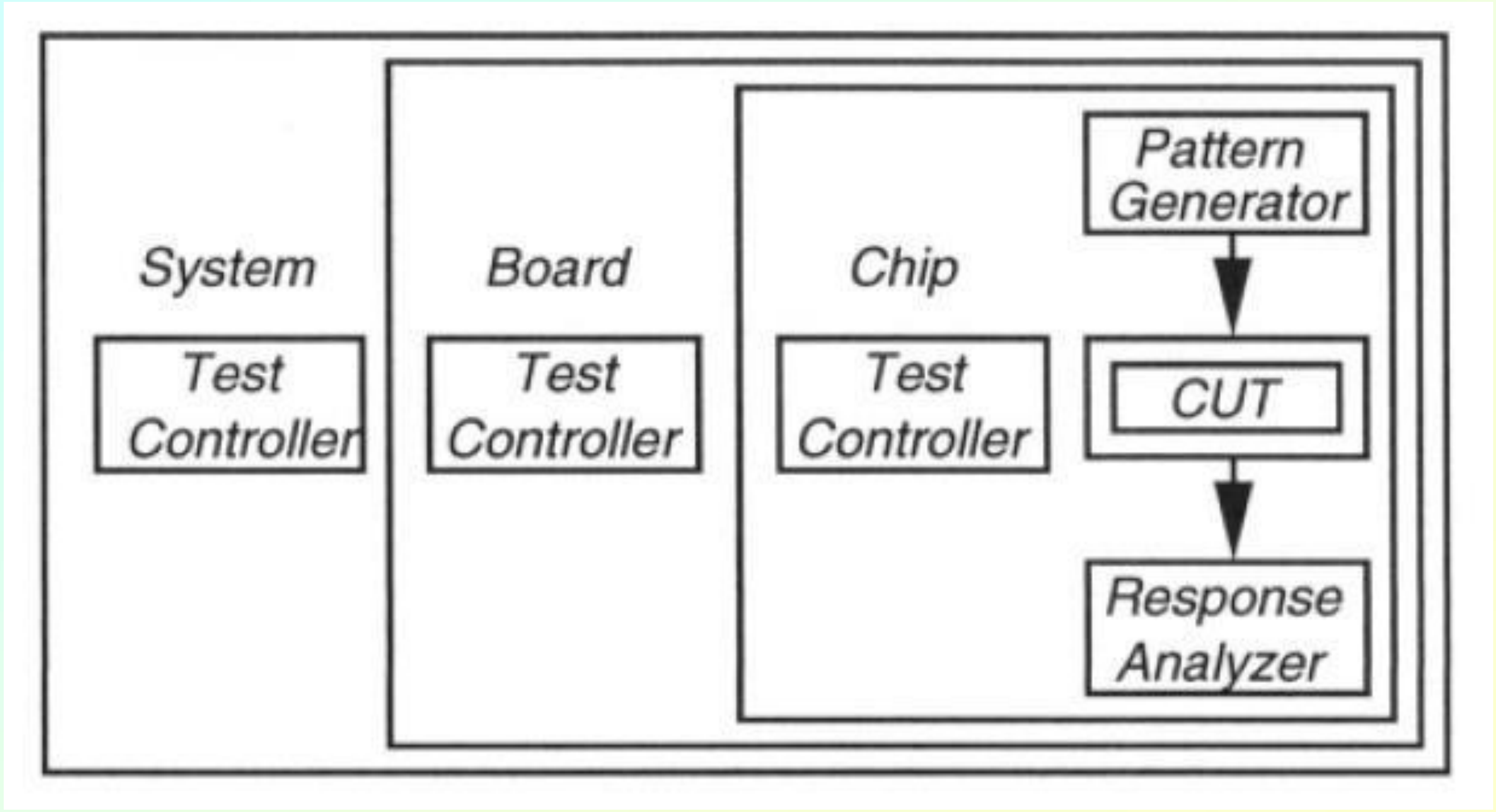
- Controllability: The difficulty of setting a specific signal to 0 or 1
- Observability: The difficulty of reading a specific signal
- Electron beam testing is too expensive
- Must set signal through primary inputs and observe through primary outputs

Design For Testability (DFT)

Boundary scan

- In boundary scan, all flip-flops enter a test mode where they are controllable and observable
- After functional verification, normal flip-flops are replaced by scan flip-flops
- Only D flip-flops must be used
- Clocks must not be generated internally

Built-In Self-Test (BIST)



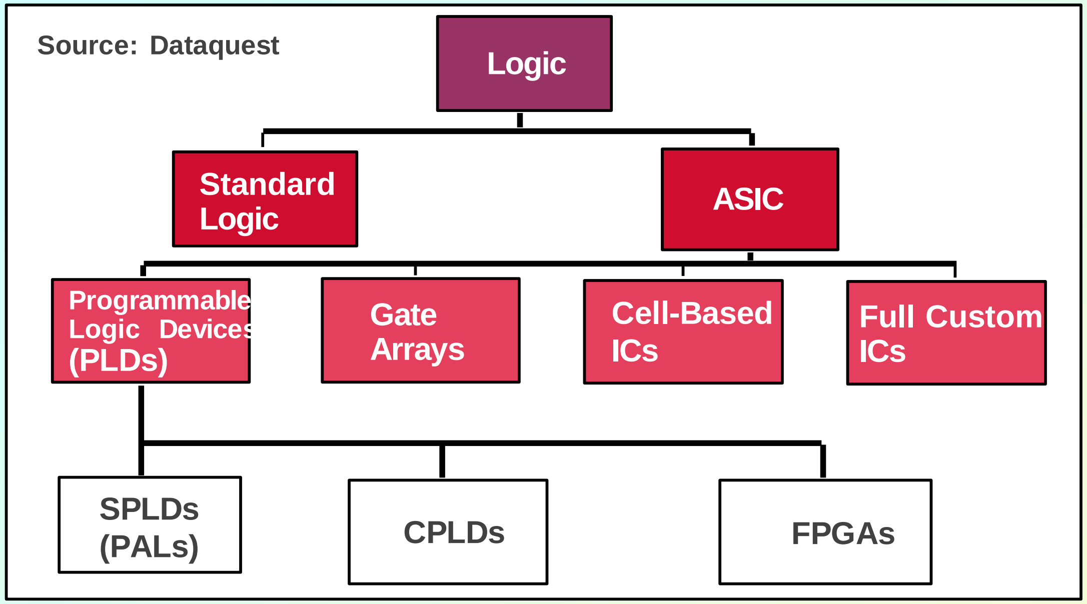
Design Methodology

VLSI Design Styles

- Programmable Logic Devices
- Standard Cell Based Design
- Full Custom Design

Programmable Logic Devices

Programmable Logic Device Families



Acronyms

SPLD	= Simple Prog. Logic Device
PAL	= Prog. Array of Logic
CPLD	= Complex PLD
FPGA	= Field Prog. Gate Array

Common Resources

Configurable Logic Blocks (CLB)

- Memory Look-Up Table
- AND-OR planes
- Simple gates

Input / Output Blocks (IOB)

- Bidirectional, latches, inverters, pullup/pulldowns

Interconnect or Routing

- Local, internal feedback, and global

What is FPGA?

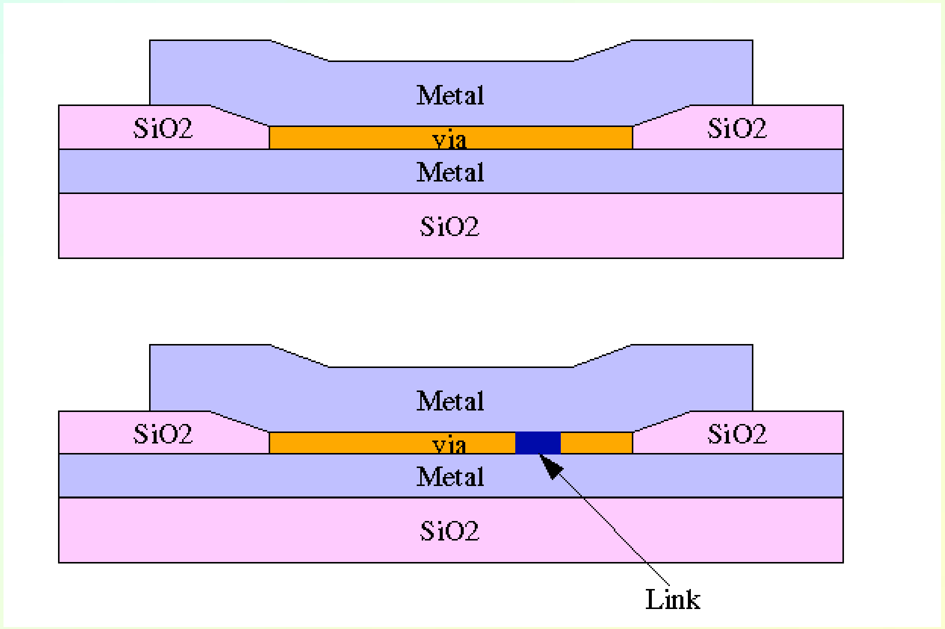
- Field Programmable Gate Arrays.
- Array of logic cells connected via routing channels.
- Special I/O cells.
- Logic cells are mainly lookup tables (LUT) with associated registers.
- Interconnection on SRAM basis or antifuse elements.

Basic Characteristics

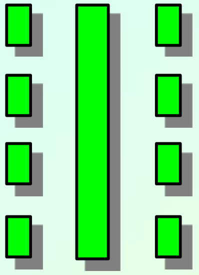
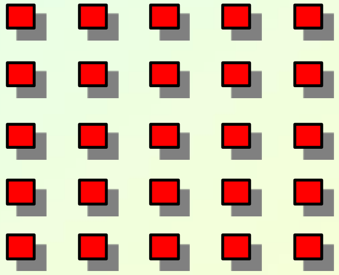
- Performance
- Density and capacity
- Ease of use
- In-system programmability and in-circuit reprogrammability

Technology and Architecture Tradeoffs

- **Antifuse elements**
- **High density**
- **Non volatile**
- **Not reprogrammable**

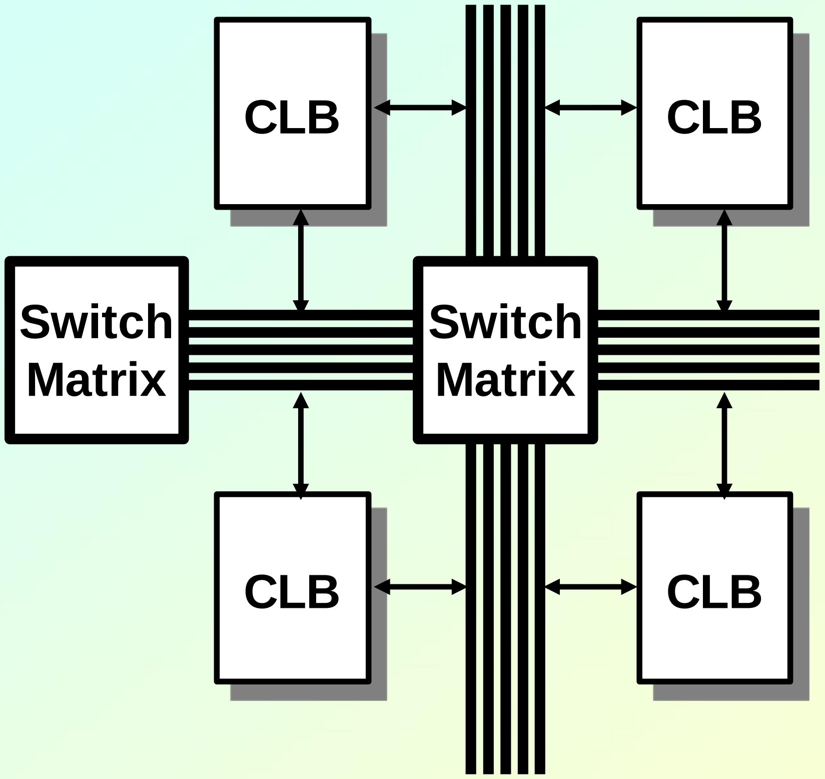


CPLDs and FPGAs

	CPLD Complex Programmable Logic Device	FPGA Field-Programmable Gate Array
		
Architecture	PAL/22V10-like More Combinational	Gate array-like More Registers + RAM
Density	Low-to-medium 0.5-10K logic gates	Medium-to-high 1K to 500K system gates
Performance	Predictable timing Up to 200 MHz today	Application dependent Up to 135MHz today

Xilinx FPGA Routing

- **1) Fast Direct Interconnect - CLB to CLB**
- **2) General Purpose Interconnect - Uses switch matrix**



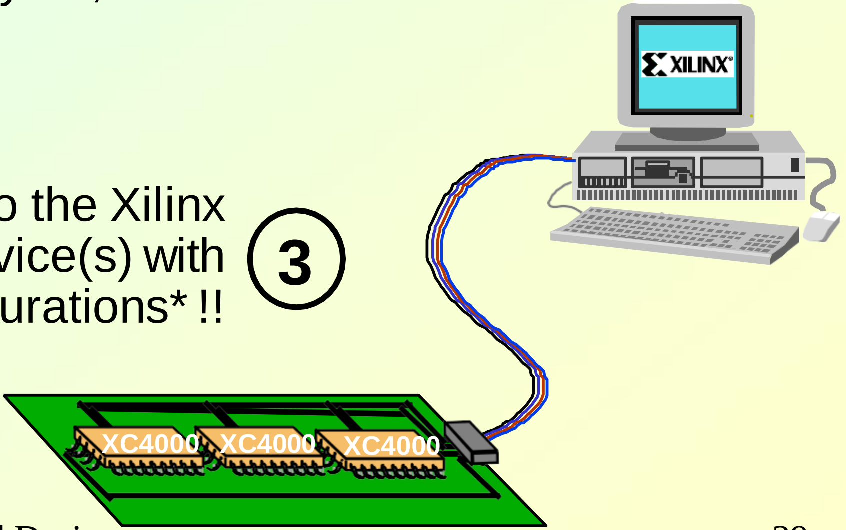
Design Flow

1 **Design Entry** in schematic, ABEL, VHDL, and/or Verilog. Vendors include Synopsys, Aldec (Xilinx Foundation), Mentor, Cadence, Viewlogic, and 35 others.

M1 Technology

2 **Implementation** includes Placement & Routing and bitstream generation using Xilinx's M1 Technology. Also, analyze timing, view layout, and more.

Download directly to the Xilinx hardware device(s) with unlimited reconfigurations* !! 3



Altera FPGA Family

- Altera Flex10K/10KE
  - LEs (Logic elements) have 4-input LUTS (look-up tables) +1 FF.
  - Fast Carry Chain between LE’s, cascade chain for logic operations.
  - Large blocks of SRAM available as well.
- Altera Max7000/Max7000A
  - EEPROM based, very fast (Tpd = 7.5 ns).
  - Basically a PLD architecture with programmable interconnects.

Standard Cell Based Design

Introduction

- One of the most prevalent custom design styles.
  - Also called semi-custom design style.
  - Requires development of a full custom mask set.
- Basic idea:
  - All of the commonly used logic cells are developed, characterized, and stored in a standard cell library.
  - A typical library may contain a few hundred cells including inverters, NAND gates, NOR gates, complex AOI, OAI gates, D-latches, and flip-flops.

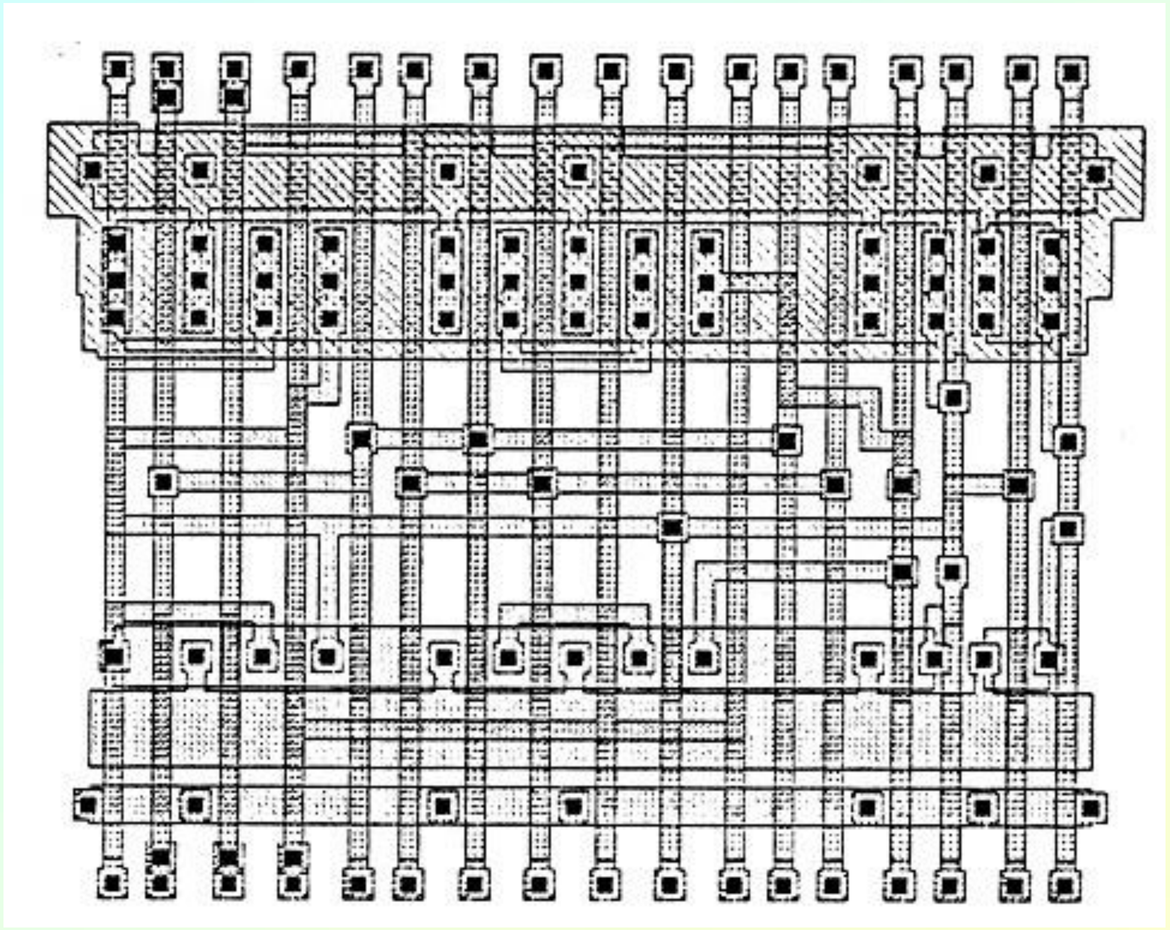
Contd.

- Each gate type can have multiple implementations to provide adequate driving capability for different fanouts.
  - For instance, the inverter can have
    - standard size transistors,
    - double size transistors, and
    - quadruple size transistors
  - The chip designer can choose the proper size to achieve high circuit speed and layout density.

Characteristic of the Cells

- Each cell is designed with a fixed height.
  - To enable automated placement of the cells, and
  - Routing of inter-cell connections.
  - A number of cells can be abutted side-by-side to form rows.
- The power and ground rails typically run parallel to the upper and lower boundaries of the cell.
  - Neighboring cells share a common power and ground bus.
  - nMOS transistors are located closer to the ground rail while the pMOS transistors are placed closer to the power rail.
- The input and output pins are located on the upper and lower boundaries of the cell.

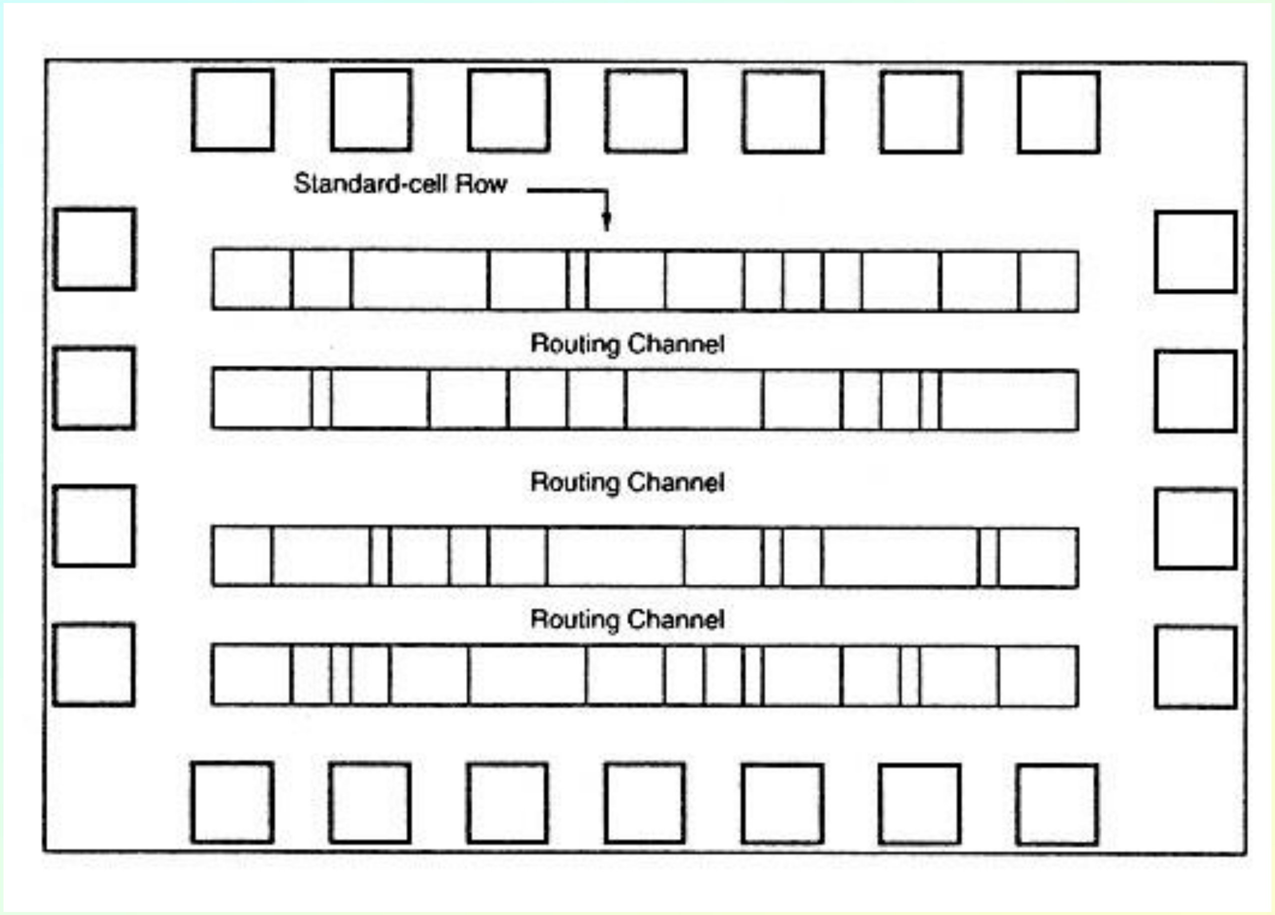
Layout of a Typical Standard Cell



Floorplan for Standard Cell Design

- **Inside the I/O frame which is reserved for I/O cells, the chip area contains rows or columns of standard cells.**
  - **Between cell rows are channels for dedicated inter-cell routing.**
  - **Over-the-cell routing is also possible.**
- **The physical design and layout of logic cells ensure that**
  - **When placed into rows, their heights match.**
  - **Neighboring cells can be abutted side-by-side, which provides natural connections for power and ground lines in each row.**

Contd.


Standard-cell Row
Routing Channel
Routing Channel
Routing Channel

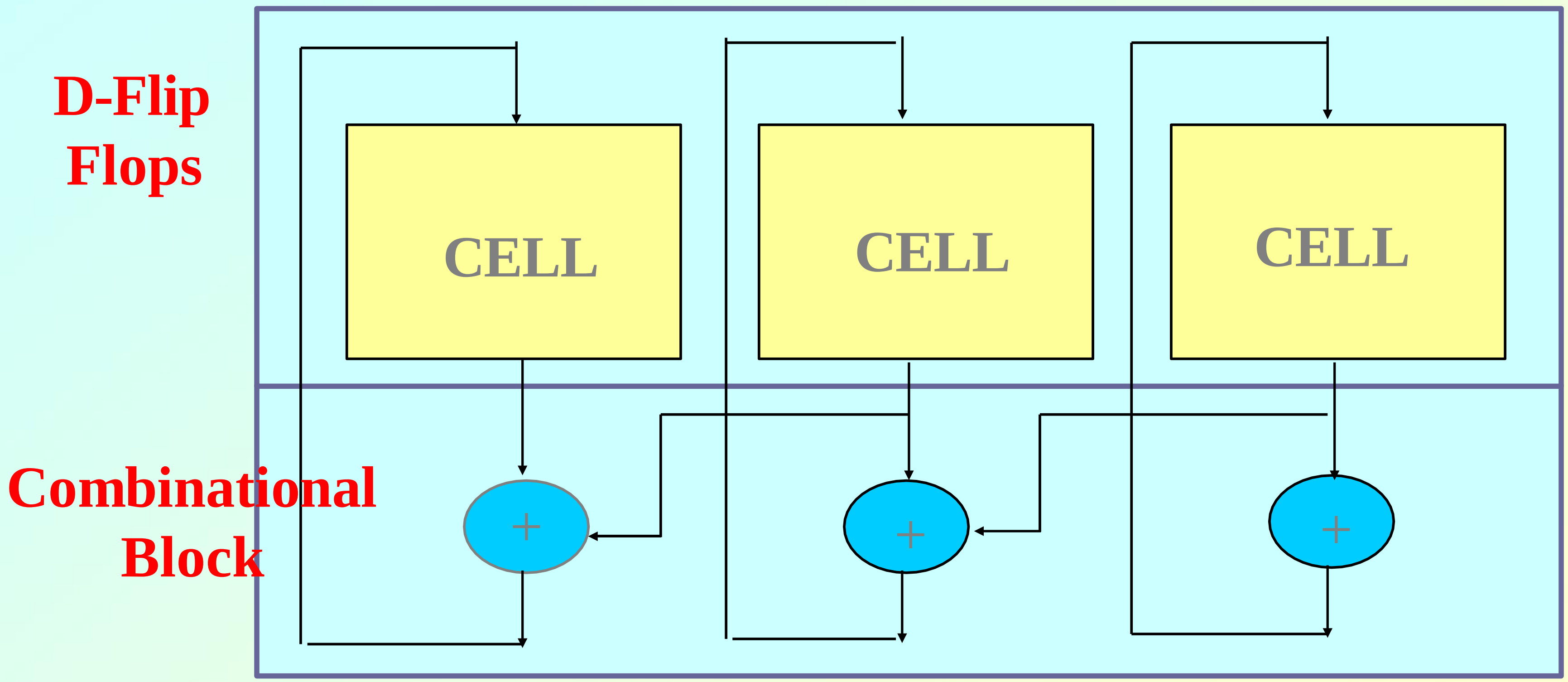
Contd.

- After chip logic design is done using standard cells in the library:
  - The most challenging task is to place individual cells into rows.
  - Interconnect them in a way that meets stringent design goals in
    - circuit speed,
    - chip area, and
    - power consumption.
  - Many advanced CAD tools for place-and-route have been developed and used to achieve the above goals.

A Real Life Design Example (The 1st Digital Chip designed and fabricated in Adv. VLSI Lab, IIT KGP)

Standard Cell Based

Design of a Programmable Cellular Automata (CA) Sequence Generator

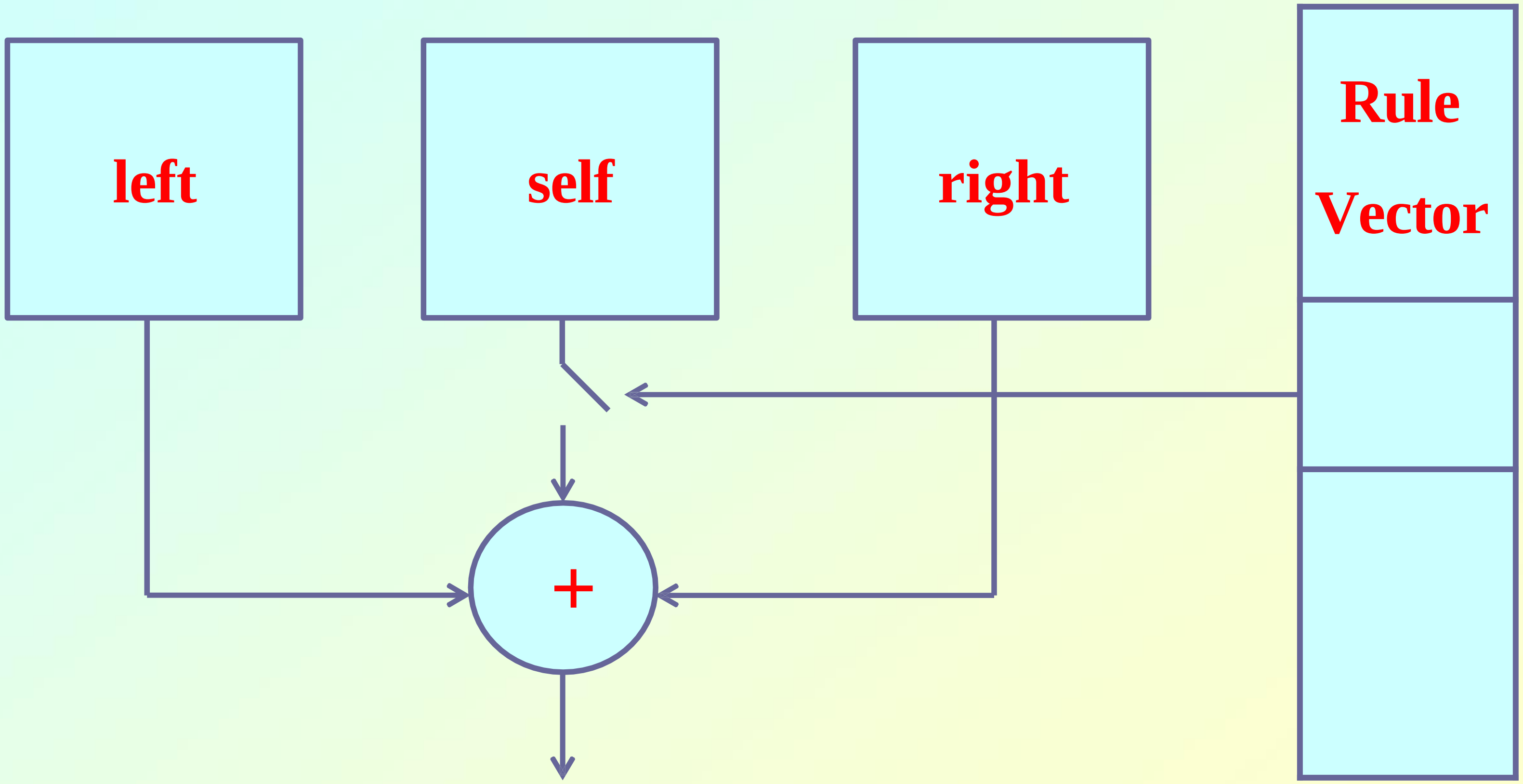


A Cellular Automaton

Behavioral Description

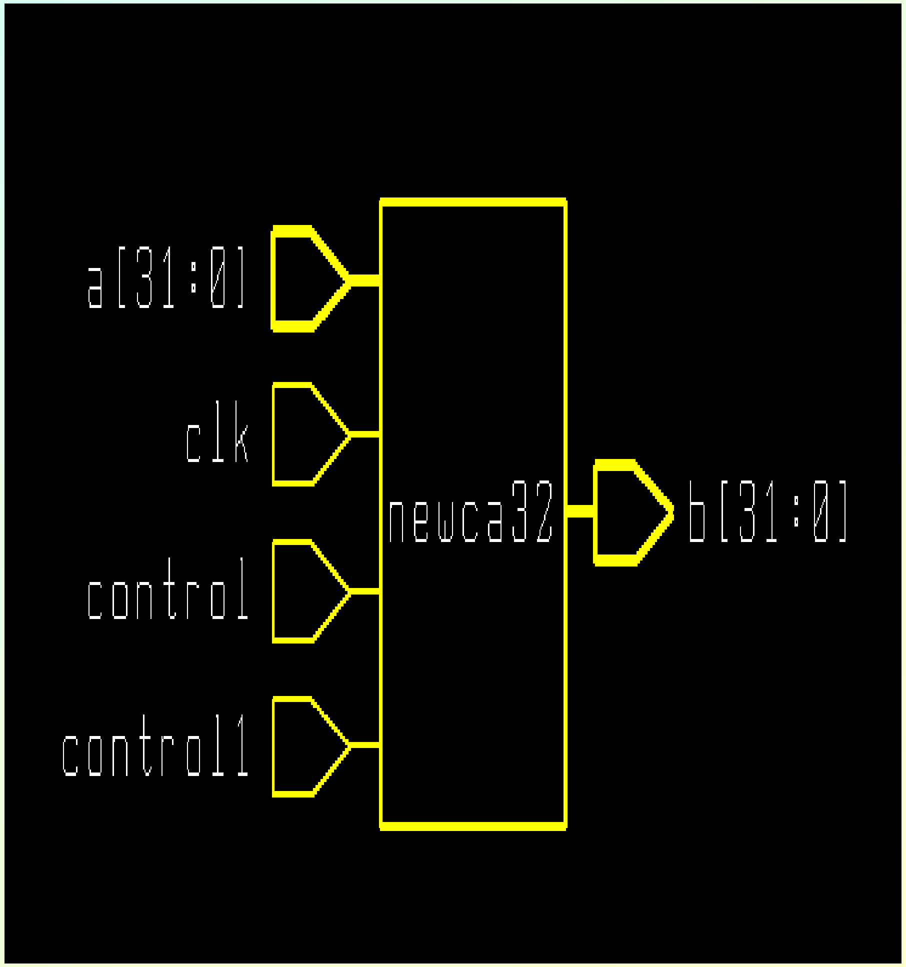
- **Data Width : 32 bits**
- **Programmable Cellular Automata for rule 90 and 150**
- **Definition : Rule is the binary representation of the truth table of the boolean function represented by the CA.**
- **3-neighbourhood CA**
- **Rule 90 : left^ right**
- **Rule 150 : left ^ self ^ right**

Programmable CA



Pin Diagram

- 32 bit input (Port ‘a’ : The rule vector)
- 32 bit output (Port ‘b’ : The state output of CA)
- 2 bit control : control and control1
- Clock port : clk



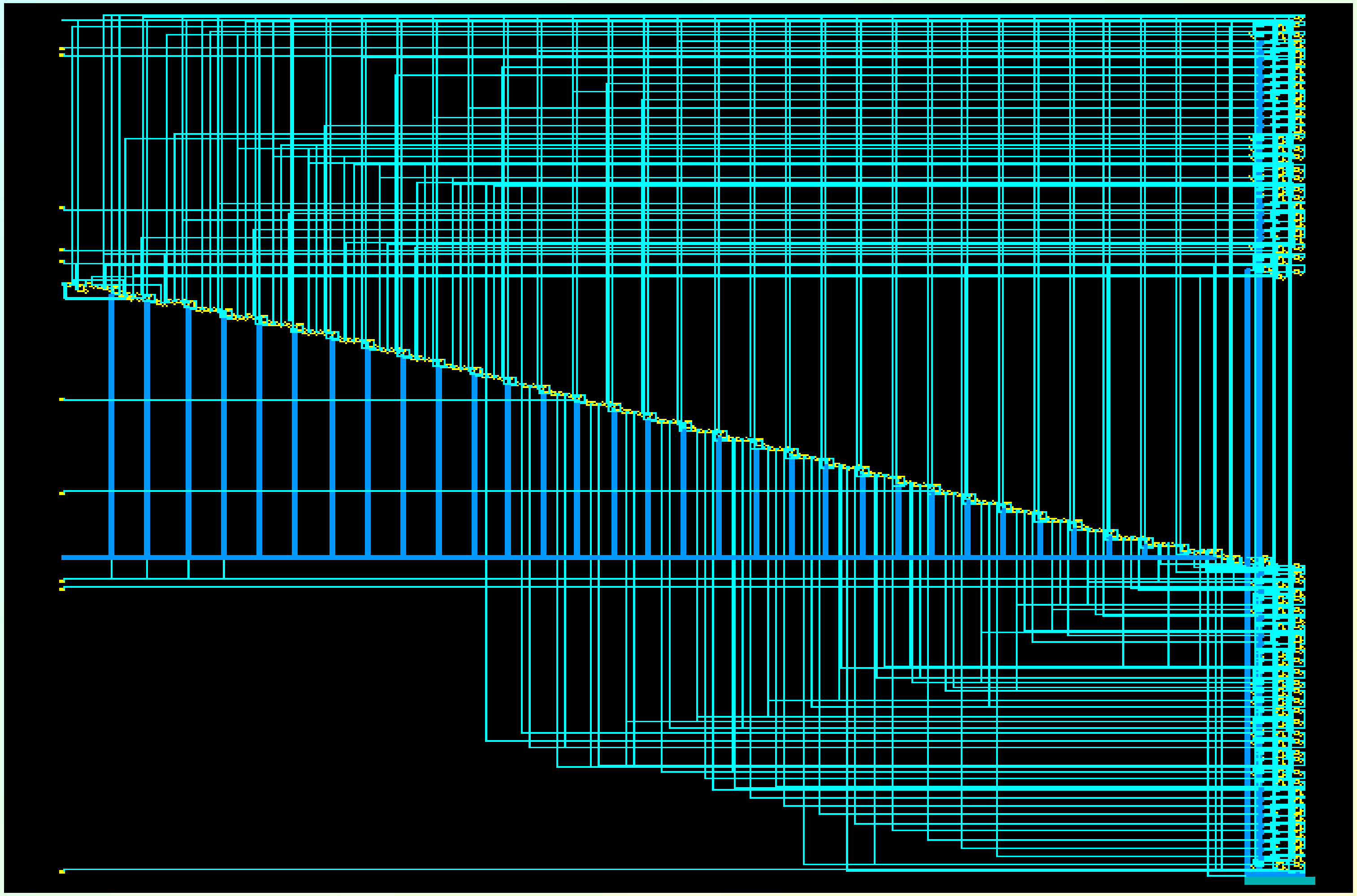
Verilog Code (HDL Entry)

```
module newca32(a,control,control1,b, clk);
     input [31:0] a;
    wire [31:0] a;
    input control,clk,control1;
    output [31:0] b;
    reg [31:0] b;
    reg [31:0] c,d,e;
    integer i;
          always @(posedge clk)
              if(control && control1) c=a;
              always @(a)
                          d=a;
```

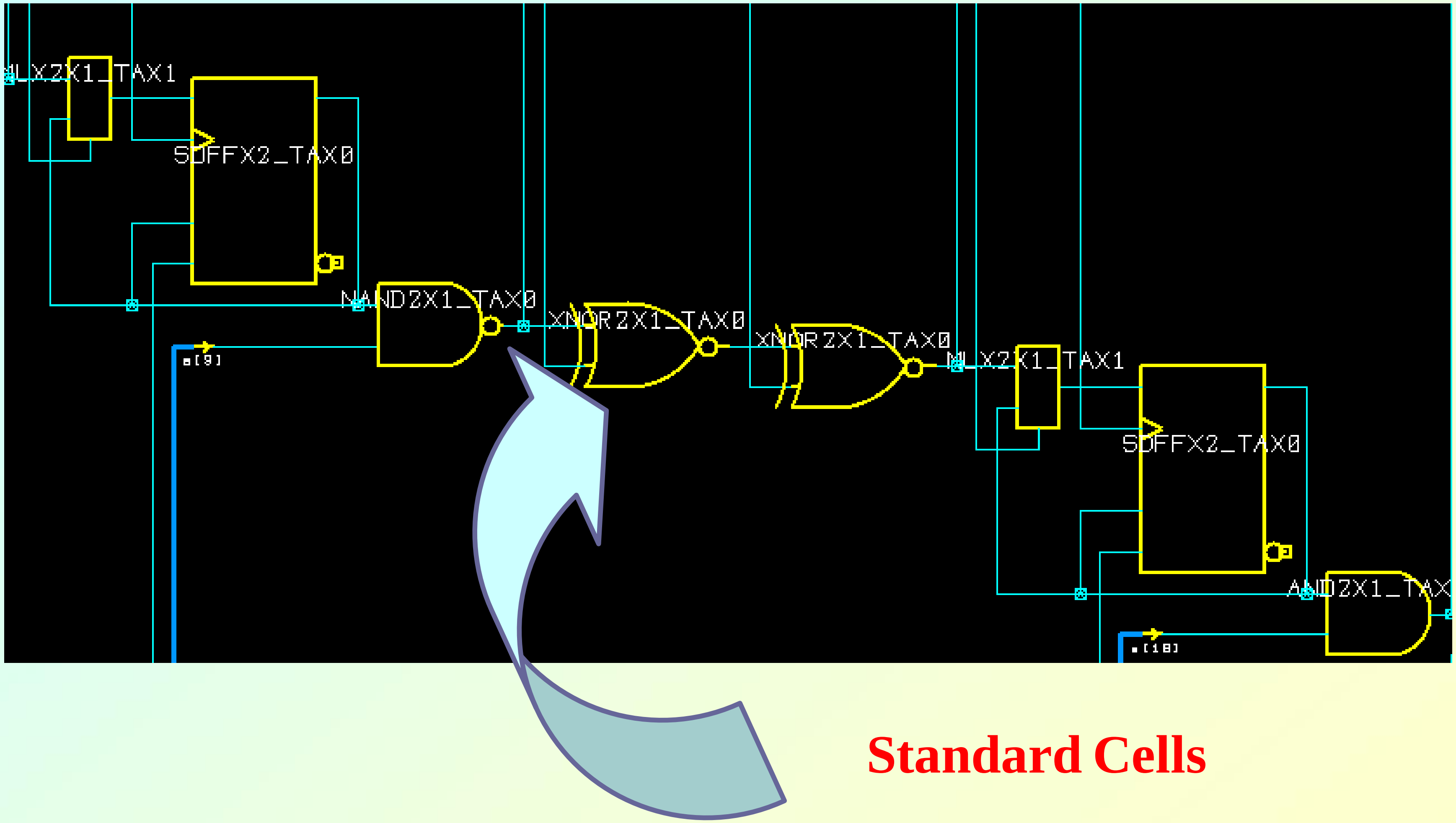
Verilog (Contd.)

```
     always @(d)
              e[0]= (d[0]&c[0])^d[1];
    always @(d)
            for(i=1;i<31;i=i+1)
               e[i]=d[i-1]^(d[i]&c[i])^d[i+1];
     always @(d)
               e[31]=d[30]^(d[31]&c[31]);
     always @(posedge clk )
        begin
           if(~control && ~control1)
           begin
              d=e;
              b=e;
         end
         end
endmodule
```

Synthesized Netlist (Using Synopsys Design Compiler)



Zoomed In Picture of the Netlist



Standard Cells

Full Custom Design

Introduction

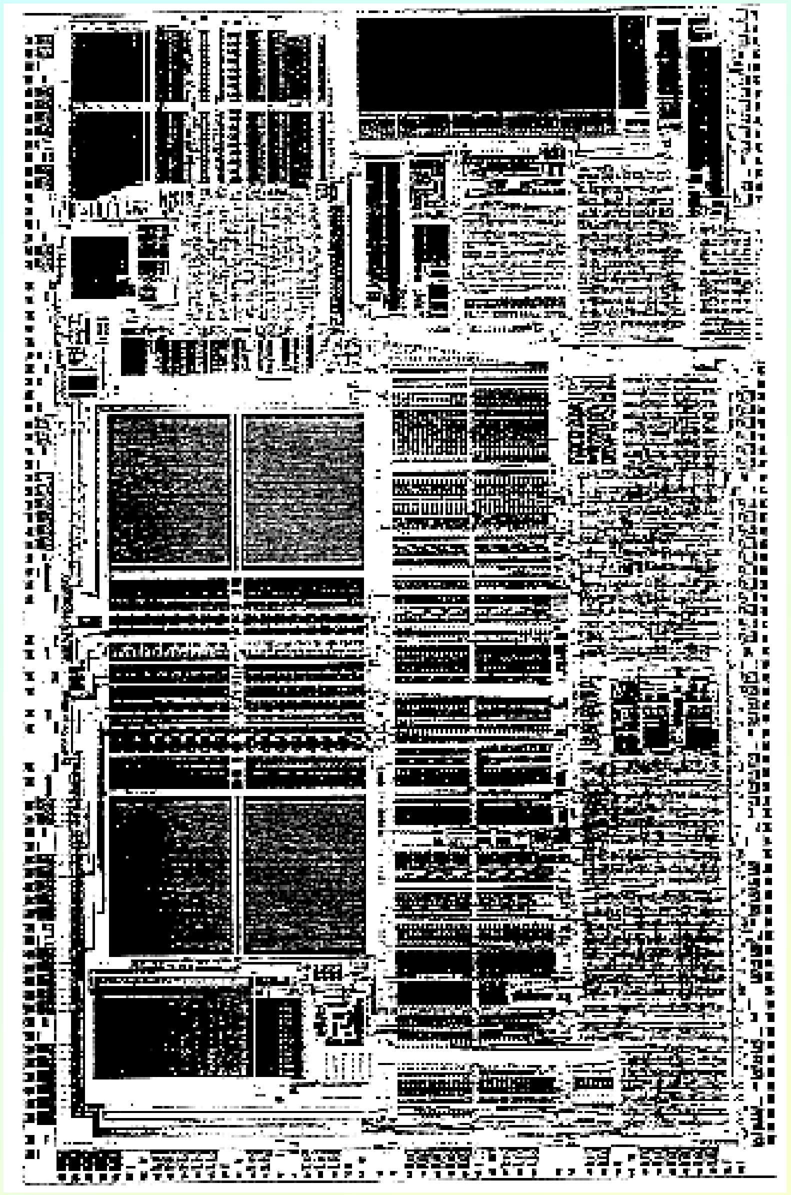
- The standard-cells based design is often called semi custom design.
  - The cells are pre-designed for general use and the same cells are utilized in many different chip designs.
- In the full custom design, the entire mask design is done anew without use of any library.
  - The development cost of such a design style is prohibitively high.
  - The concept of design reuse is becoming popular in order to reduce design cycle time and cost.

Contd.

- The most rigorous full custom design can be the design of a memory cell.
  - Static or dynamic.
  - Since the same layout design is replicated, there would not be any alternative to high density memory chip design.
- For logic chip design, a good compromise can be achieved by using a combination of different design styles on the same chip.
  - Standard cells, data-path cells and PLAs.

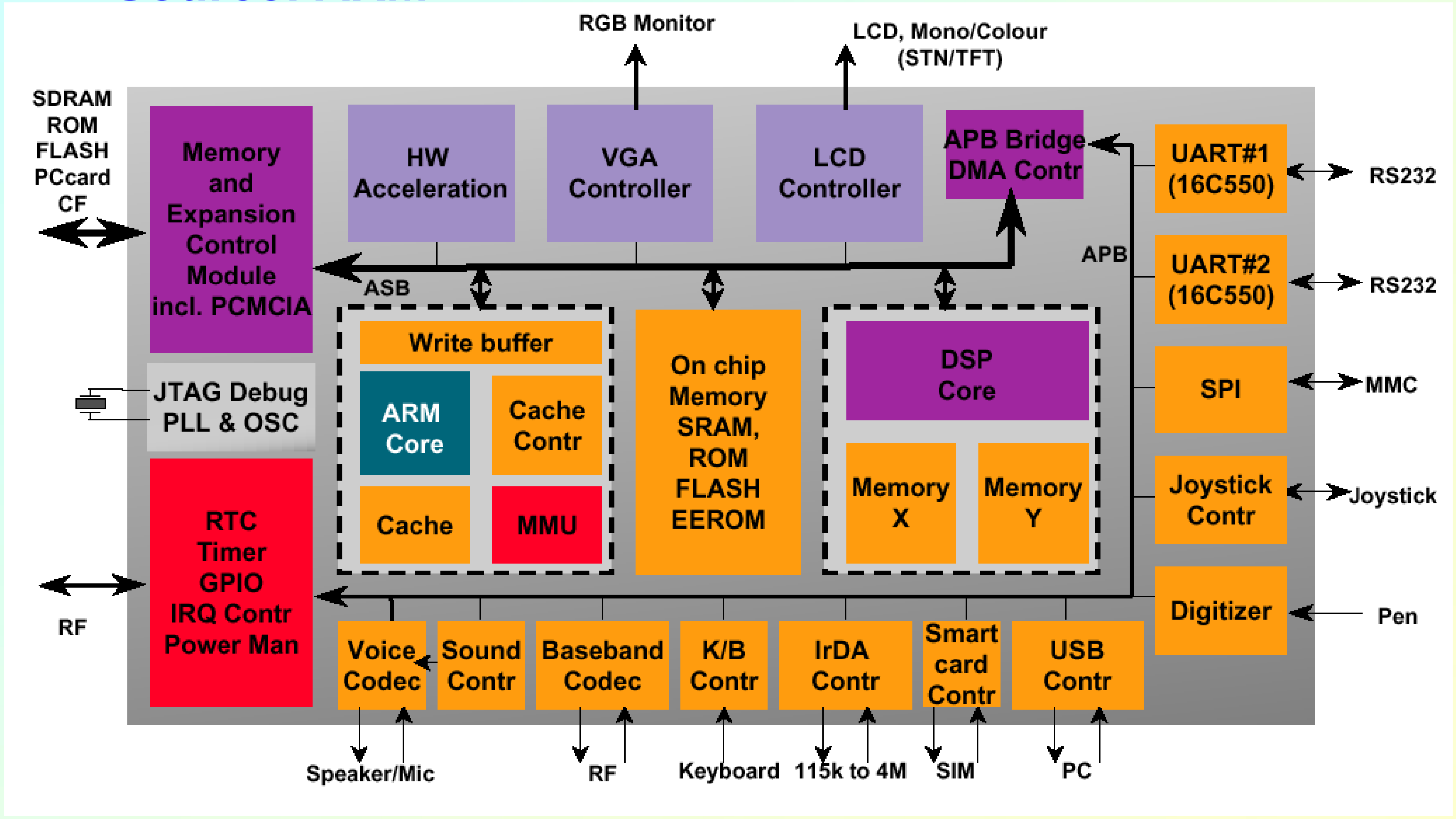
Contd.

- In real full-custom layout in which the geometry, orientation and placement of every transistor is done individually by the designer,
  - Design productivity is usually very low.
    - Typically 10 to 20 transistors per day, per designer.
- In digital CMOS VLSI, full-custom design is rarely used due to the high labor cost.
  - Exceptions to this include the design of high-volume products such as memory chips, high-performance microprocessors and FPGA masters.
- Next slide shows the full layout of the Intel 486 µP chip.
  - Good example of a hybrid full-custom design.



System on a Chip

Source: ARM



Comparison Among Various Design Styles

	Design Style			
	FPGA	Gate array	Standard cell	Full custom
Cell size	Fixed	Fixed	Fixed height	Variable
Cell type	Programmable	Fixed	Variable	Variable
Cell placement	Fixed	Fixed	In row	Variable
Interconnections	Programmable	Variable	Variable	Variable
Design time	Very fast	Fast	Medium	Slow