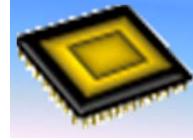


Chapter 1

Chapter 1

Introduction

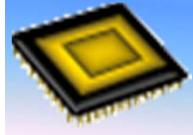


Chapter 1

2003~2006年各類IC產品銷售成長預估

IC 種類	2003 年		2004 年		2005 年		2006 年	
	銷售額	年增率	銷售額	年增率	銷售額	年增率	銷售額	年增率
微處理器	258	8.10%	286	11.10%	307	7.00%	319	4.00%
DRAM	157	2.90%	225	43.00%	164	26.80%	213	29.60%
Flash	97	25.00%	121	25.00%	135	11.00%	128	-5.00%
DSP	62	27.70%	75	20.80%	91	21.00%	96	6.00%
分離式元件	132	7.30%	149	12.10%	163	9.40%	168	3.30%
光電元件	83	22.00%	95	15.00%	106	11.70%	112	5.60%
類比晶片	258	7.90%	294	24.00%	334	13.50%	351	5.00%
邏輯元件	342	9.60%	386	12.60%	431	11.70%	452	5.00%
微控制器	103	9.90%	117	14.00%	121	3.20%	143	18.50%

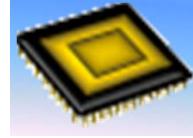
Source:WSTS 02/03



Chapter 1

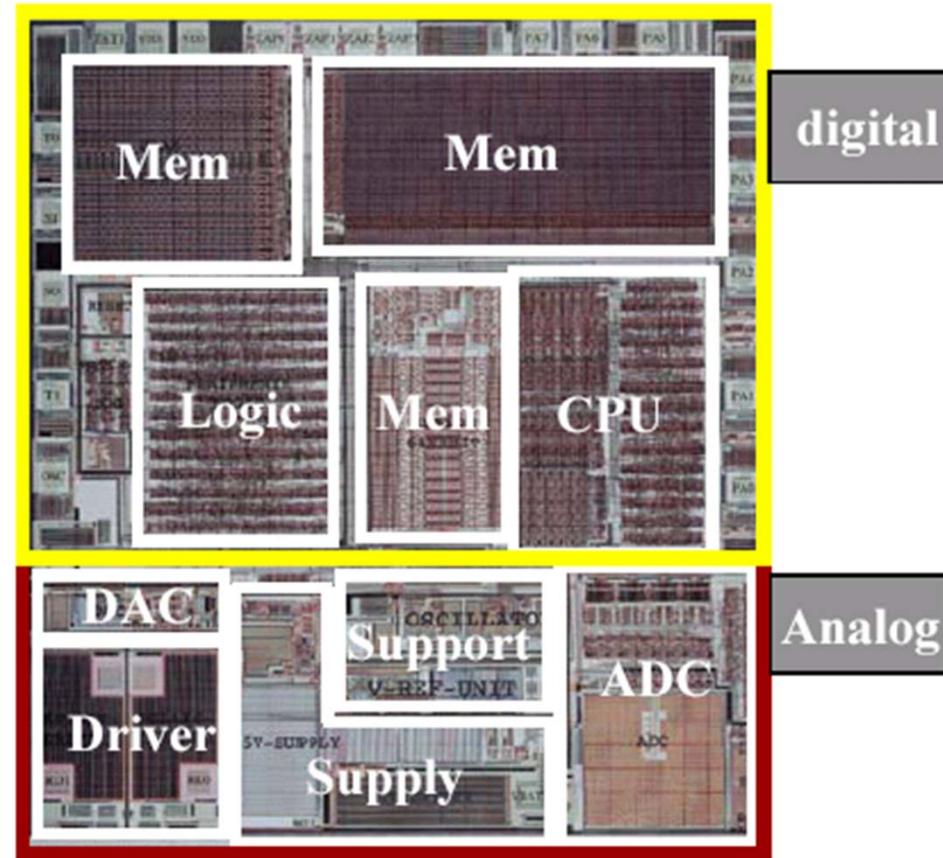
Why Integrated Circuits?

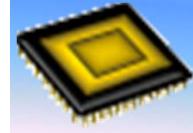
- Break this question into two questions
 - Why electronics
 - Why use ICs to build electronics
- Why use electronics
 - Electrons are easy to move / control
 - Easier to move/control electrons than real stuff
 - If you don't believe me look at a mechanical computer
 - <http://www.sciencemuseum.org.uk/on-line/treasure/mathematics.asp>
 - Move information, not things (phone, fax, WWW, etc.)
 - Takes much less energy and \$



Chapter 1

Mixed-Signal Chip

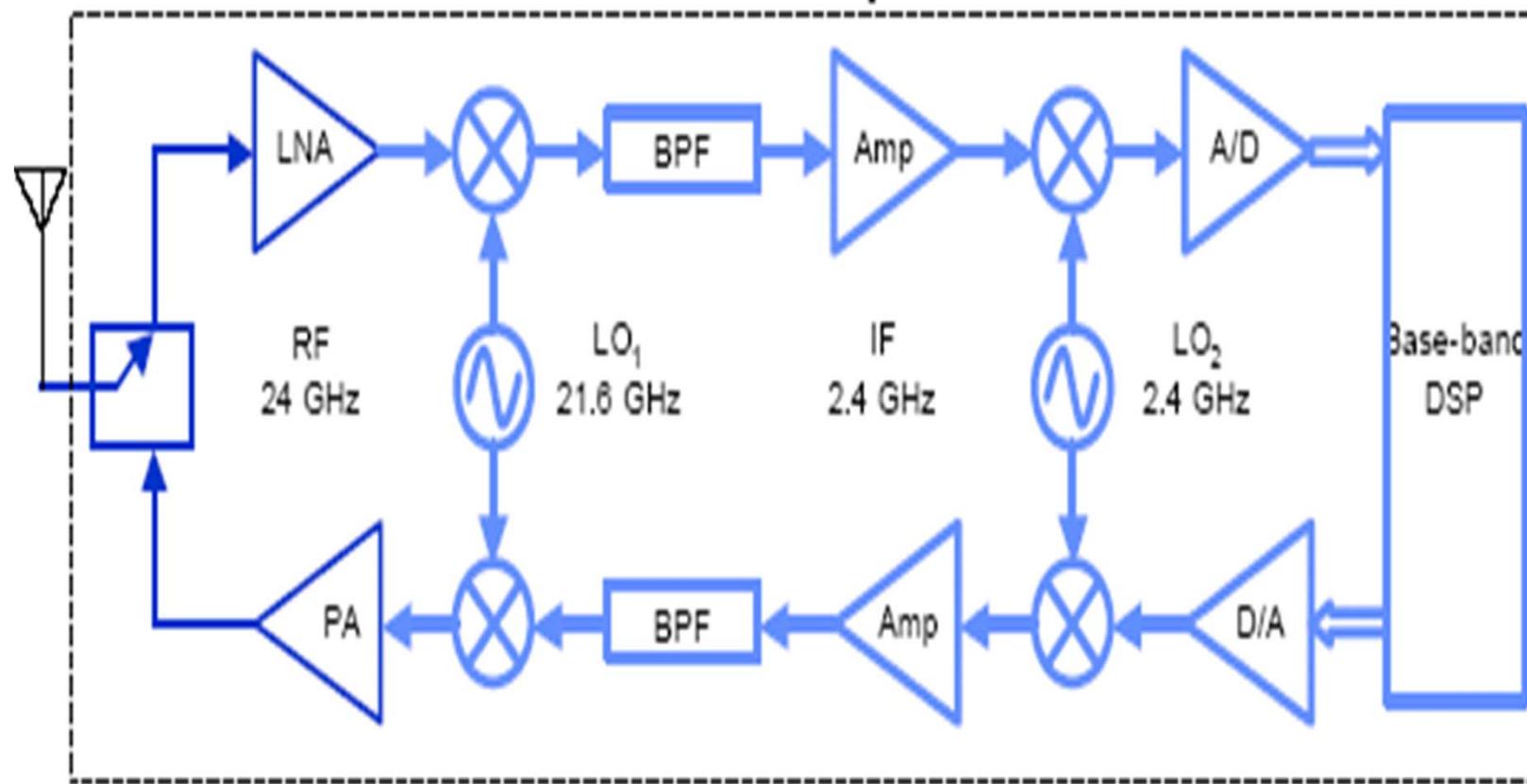


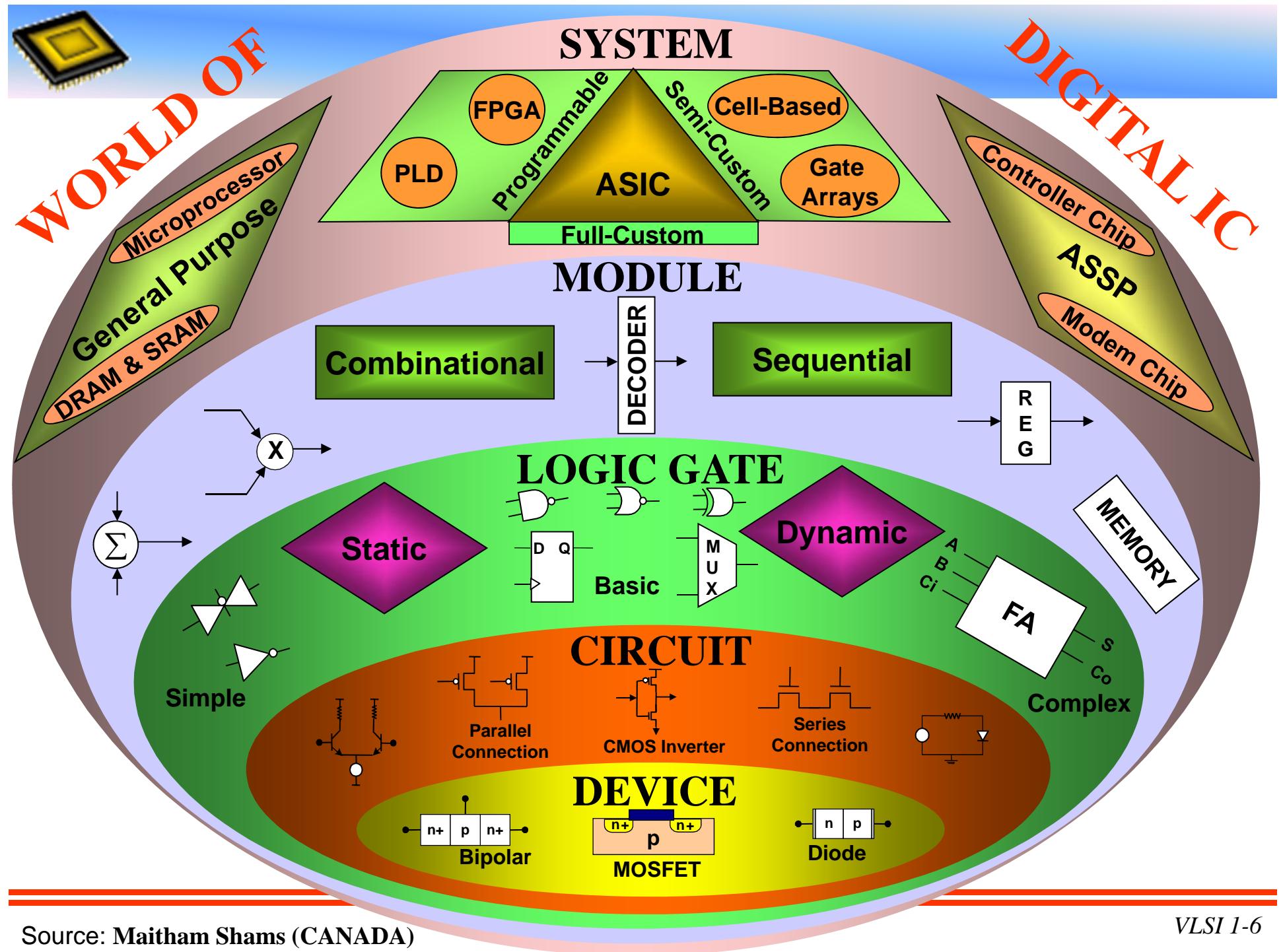


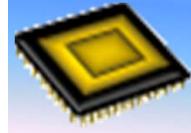
Chapter 1

Mixed-Signal Chip

➤ Transceiver /Receiver



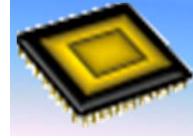




Chapter 1

Why is digital circuit so successful ?

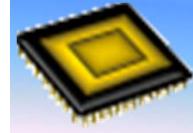
- *The success of digital technology is primarily based on simplicity of designing digital circuits and ease of their manufacture*
- Digital circuits are composed of
 - basic processing elements
 - basic memory elements
- Components are simply just gates and flip-flops.
 - input or output signals of each gate/flip-flop can assume only two values
 - Changes in the signal values are governed by laws of Boolean Algebra



Chapter 1

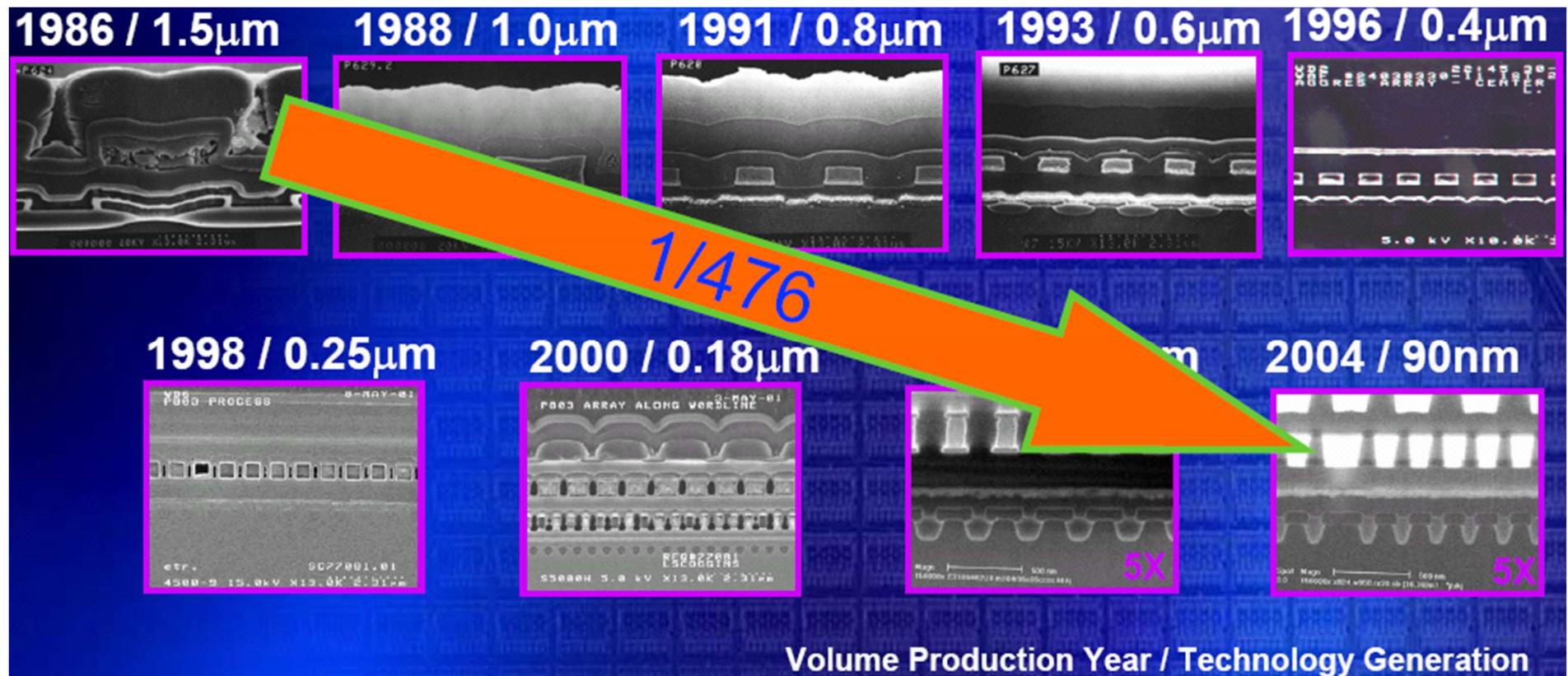
IC Design Challenges

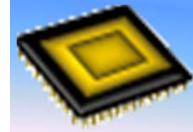
- Interconnection
- Power Dissipation
- Reliability
- Verification
- Mixed-Signal



Chapter 1

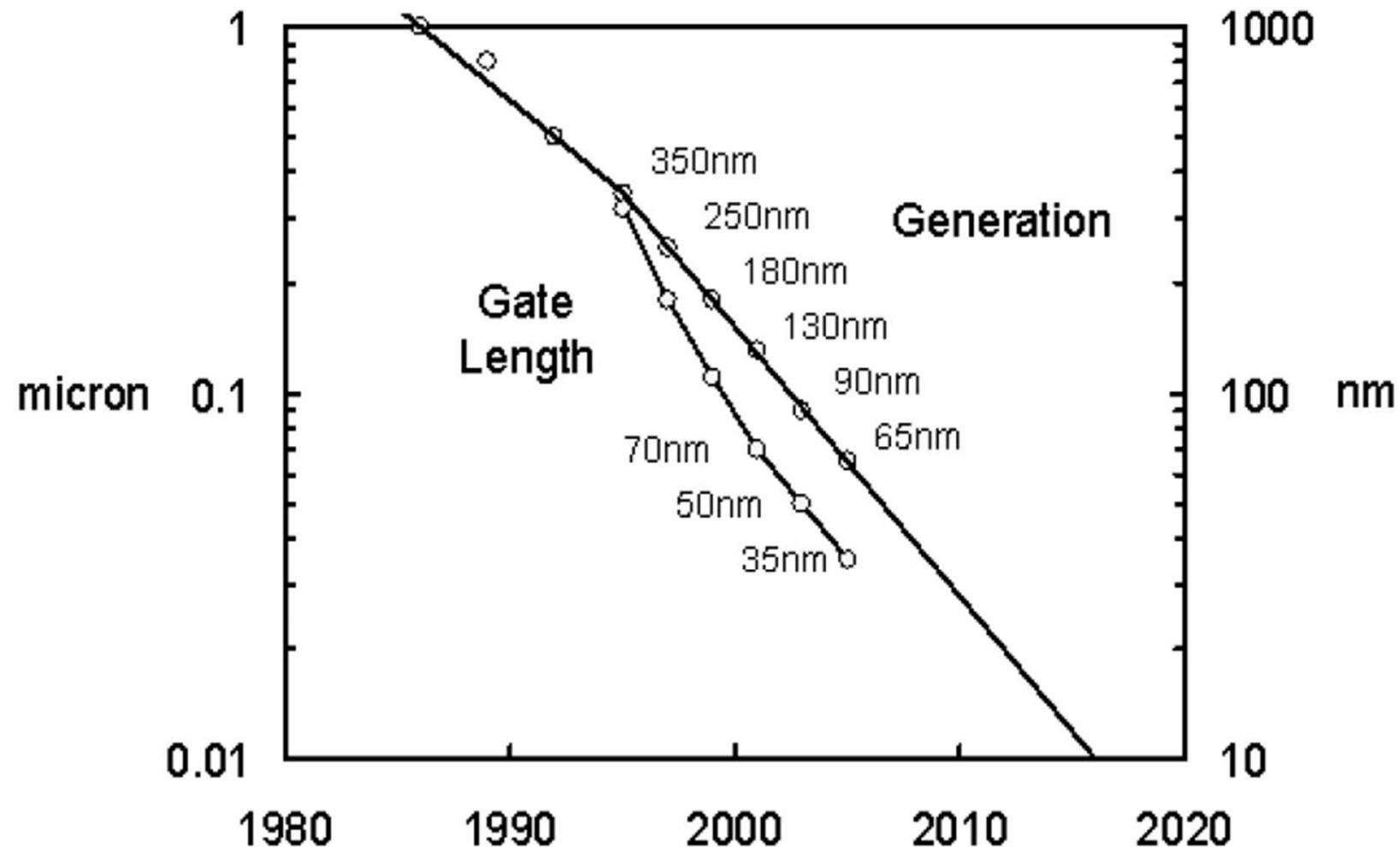
Intel Technology Scaling

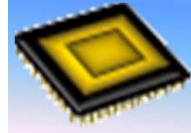




Chapter 1

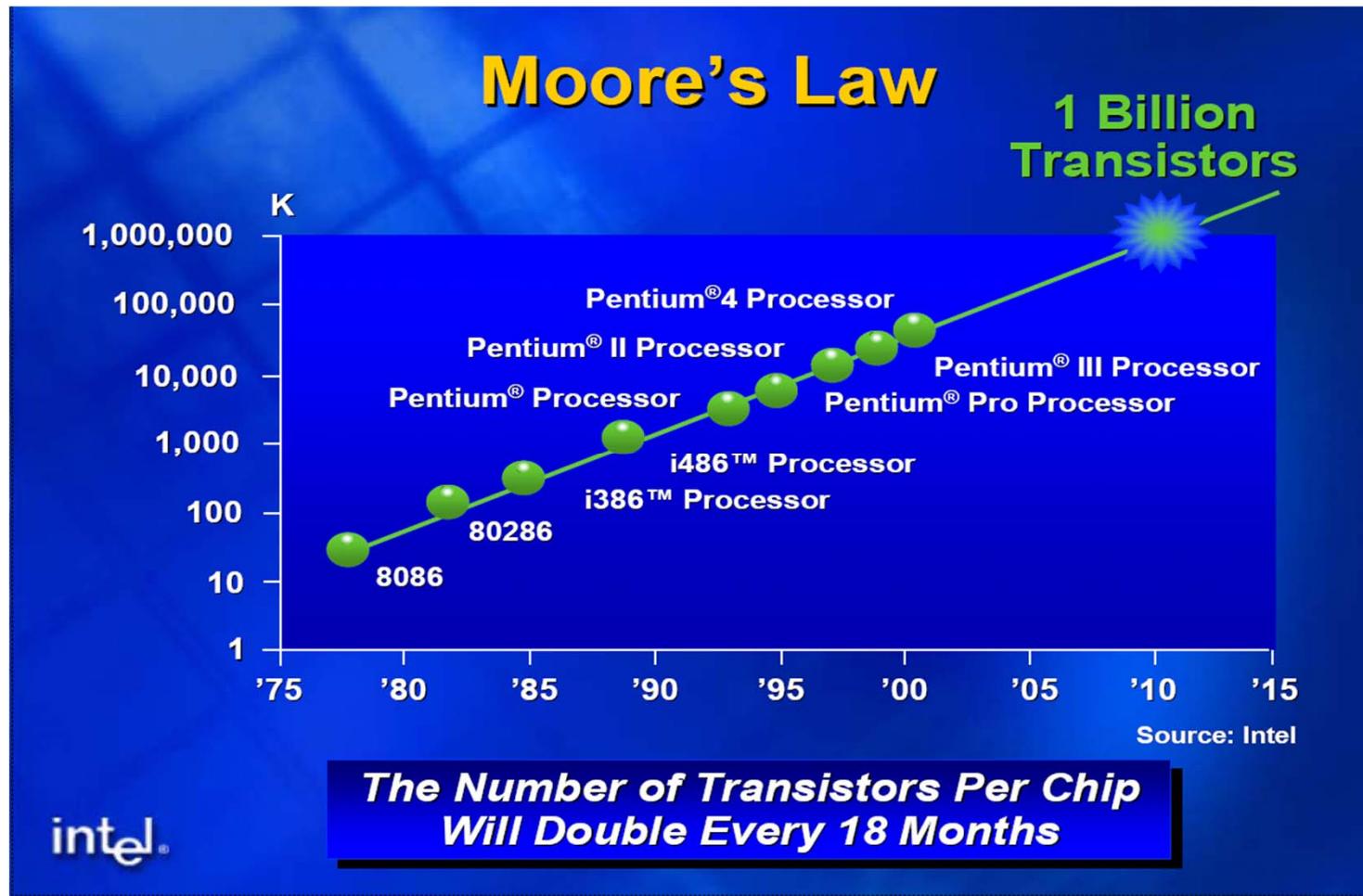
Logic Transistor Gate Length Trend

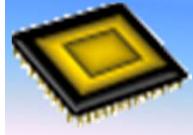




Chapter 1

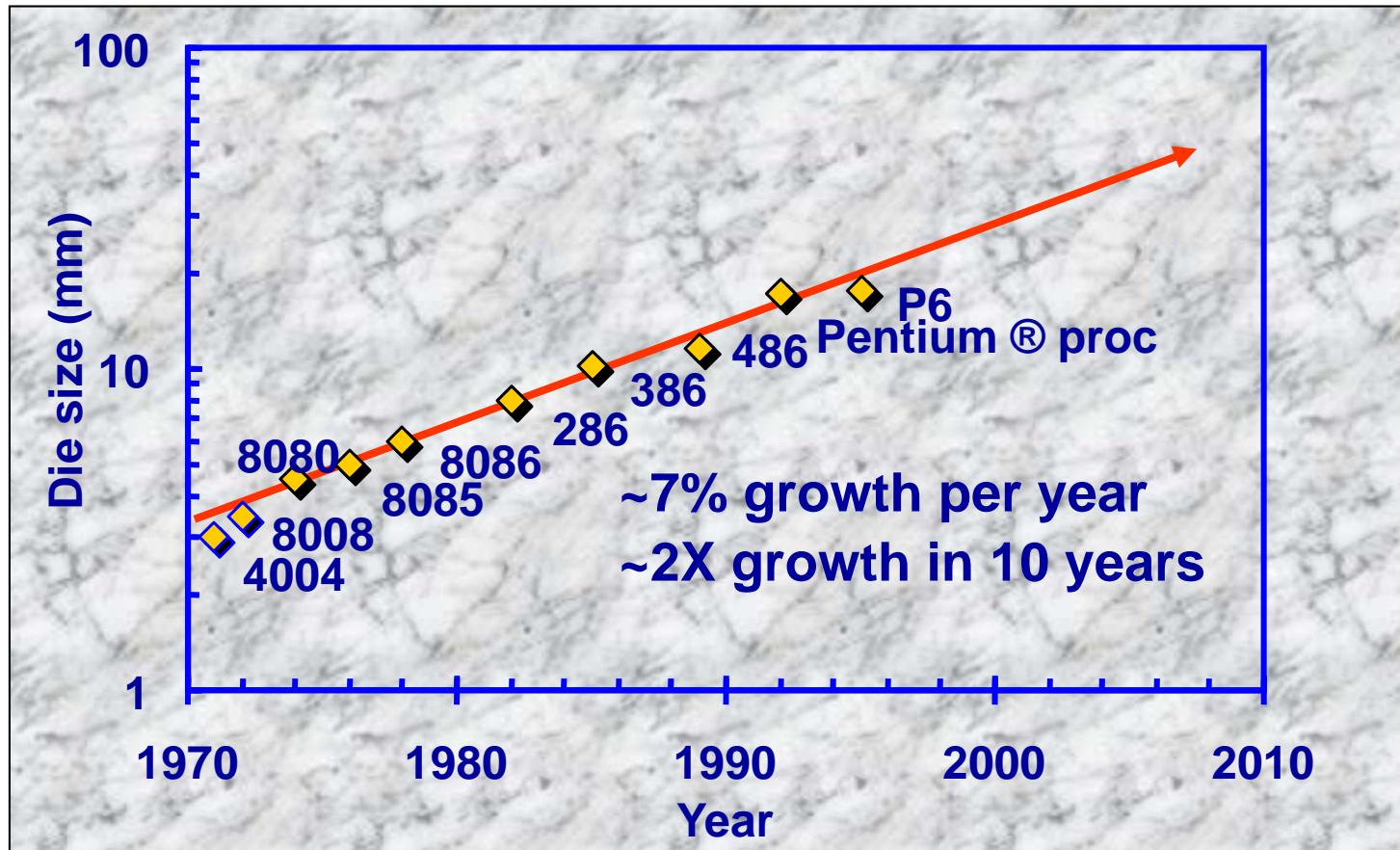
Intel's Roadmap

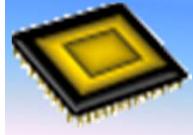




Chapter 1

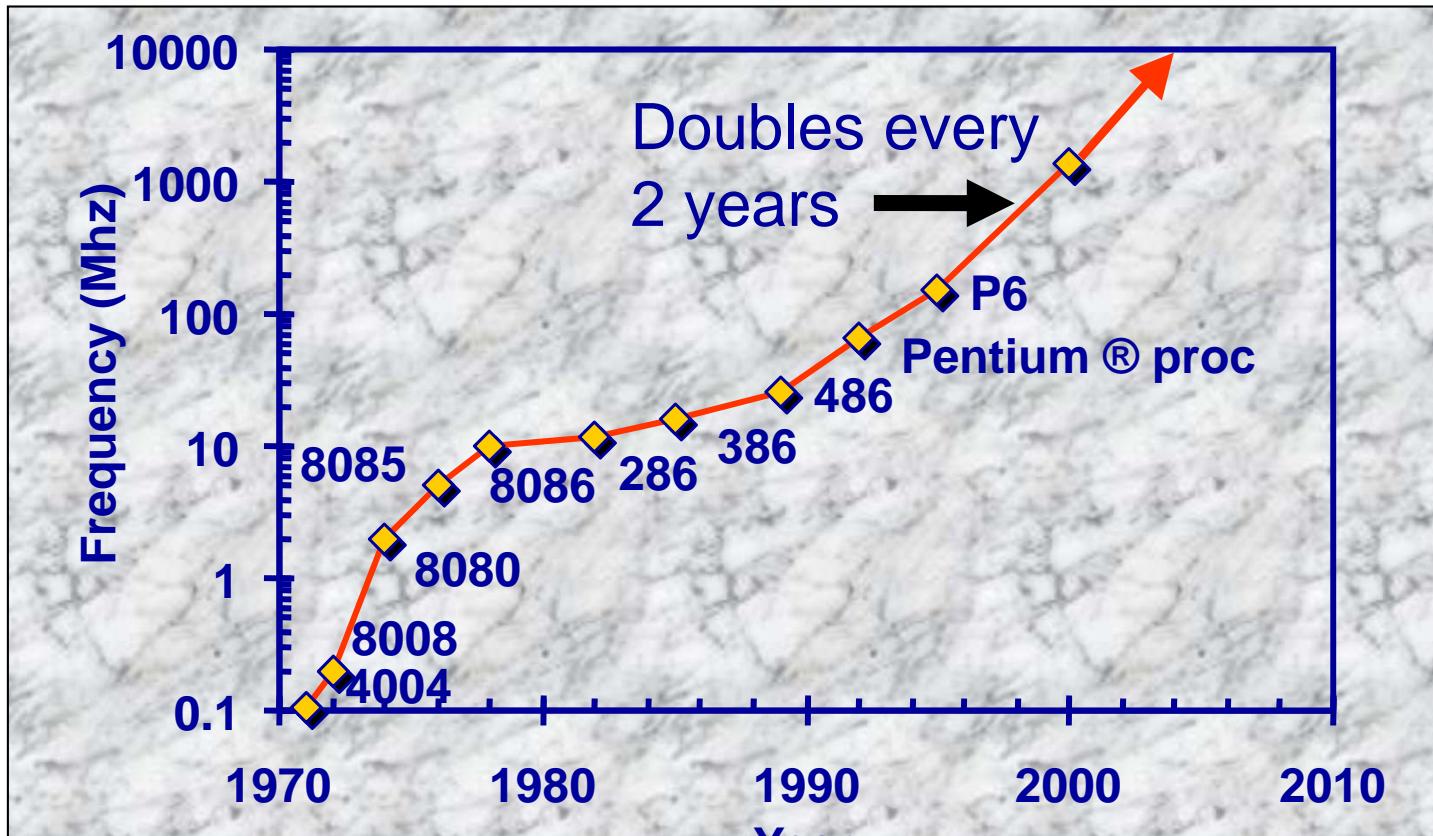
Die Size Growth



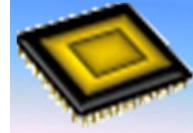


Chapter 1

Frequency



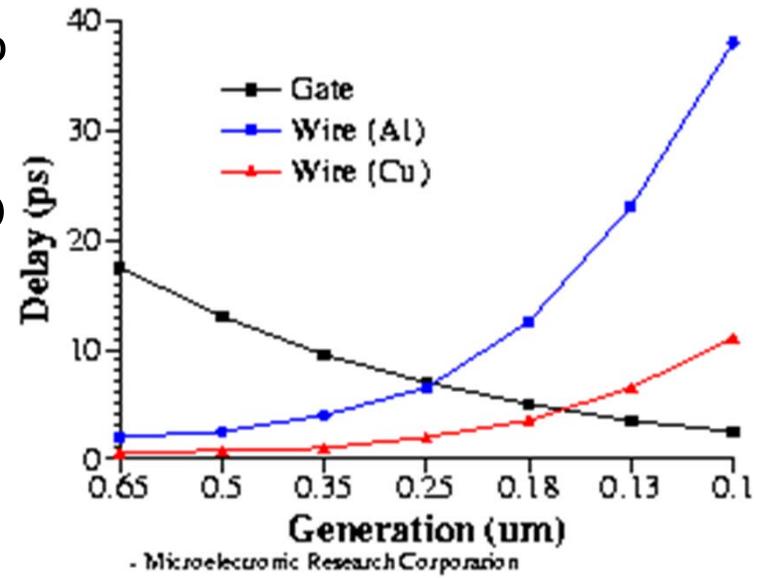
Lead Microprocessors frequency doubles every 2 years



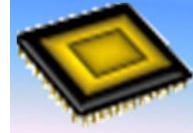
Chapter 1

Wire Delay Is Increasing

- Gate delay decreasing 25% per generation
- Wire delay increasing 100% per generation
- Communicate across a chip
 - 1 clock at 400 MHz in 0.35 μ m
 - 12.4 clocks at 1 GHz in 0.1 μ m

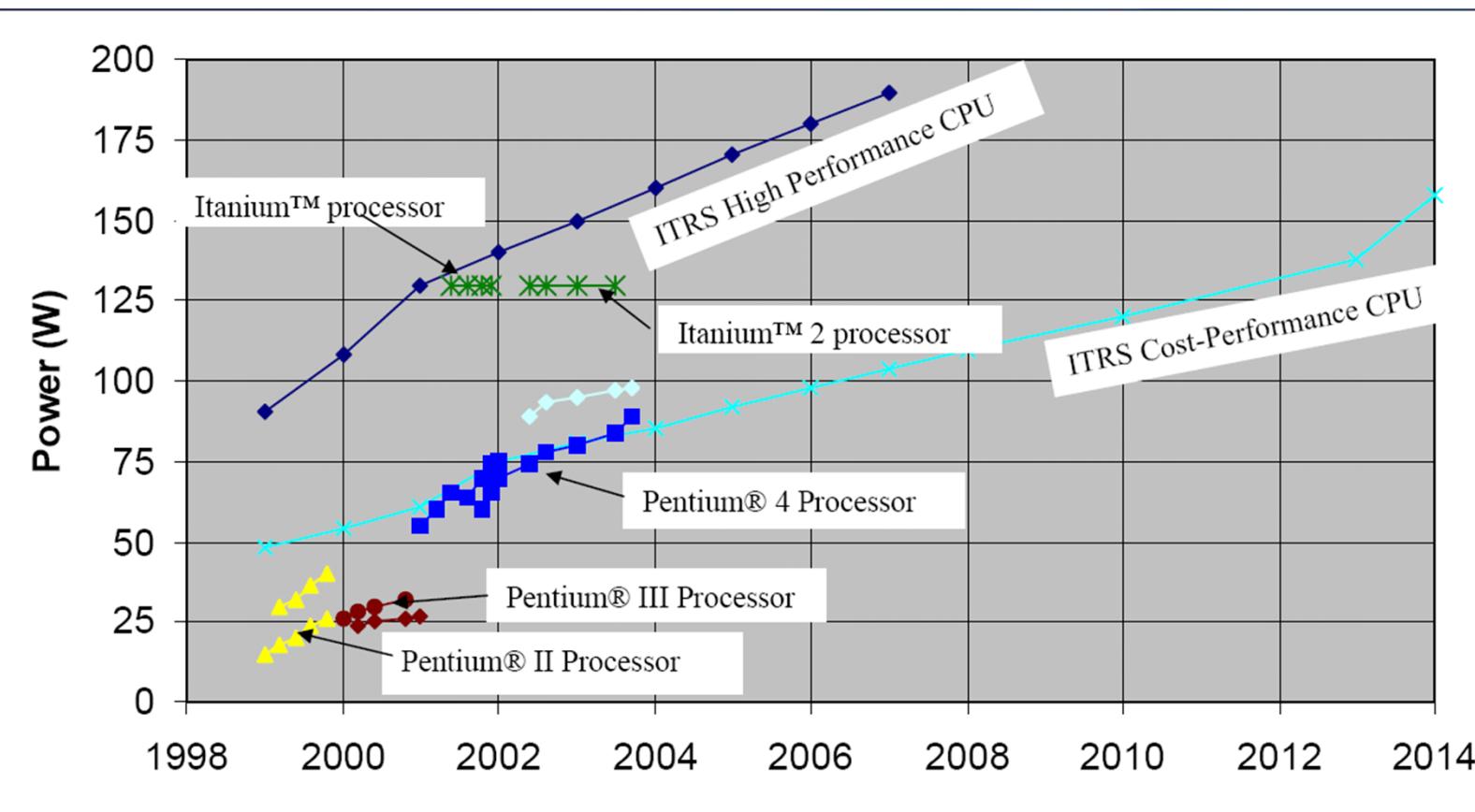


Source: Intel Corporation

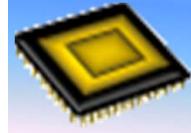


Chapter 1

Increasing Power Requirements

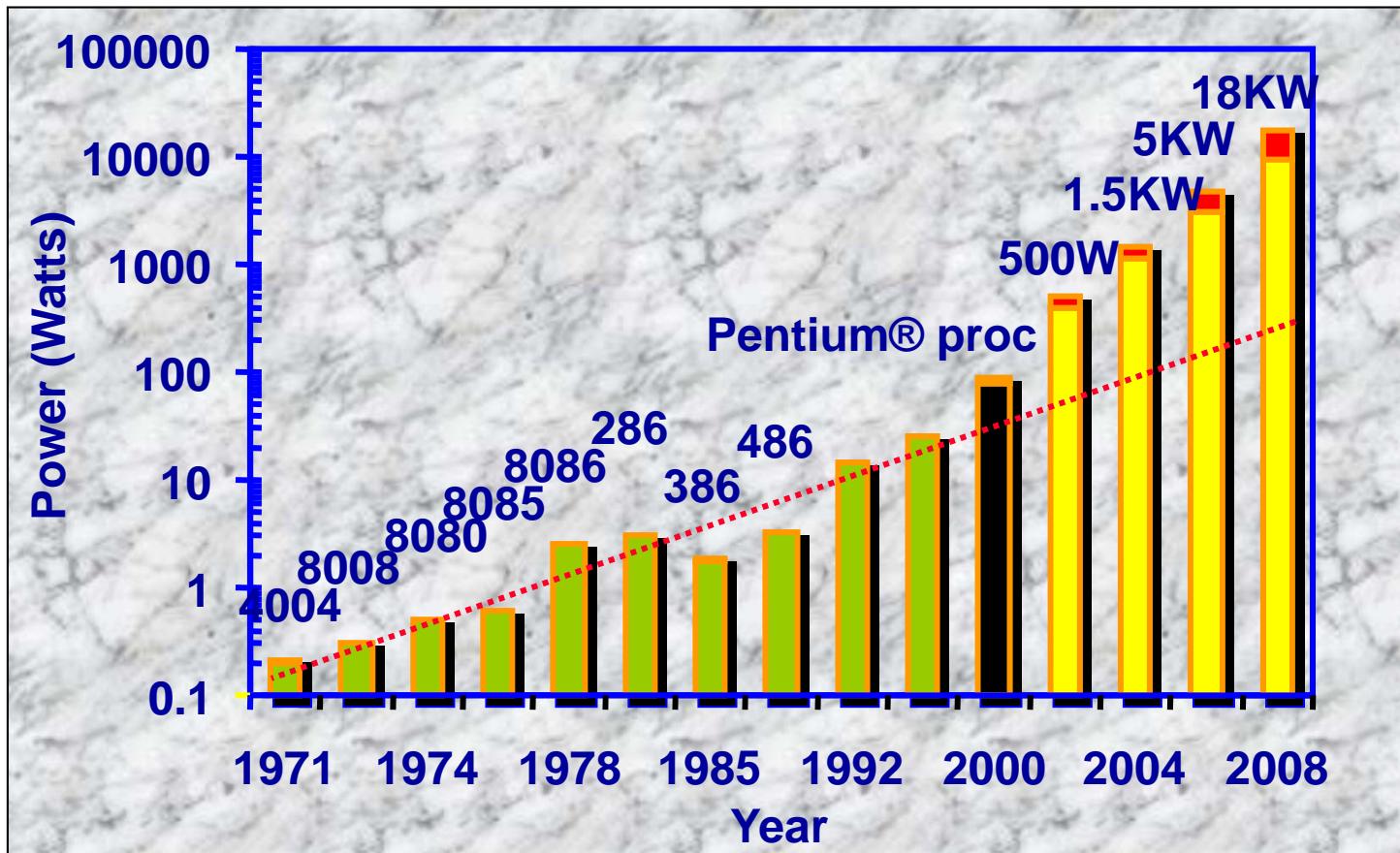


➤ 100W+ processors in 2006

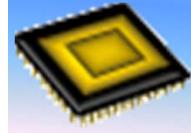


Chapter 1

Power will be a major problem

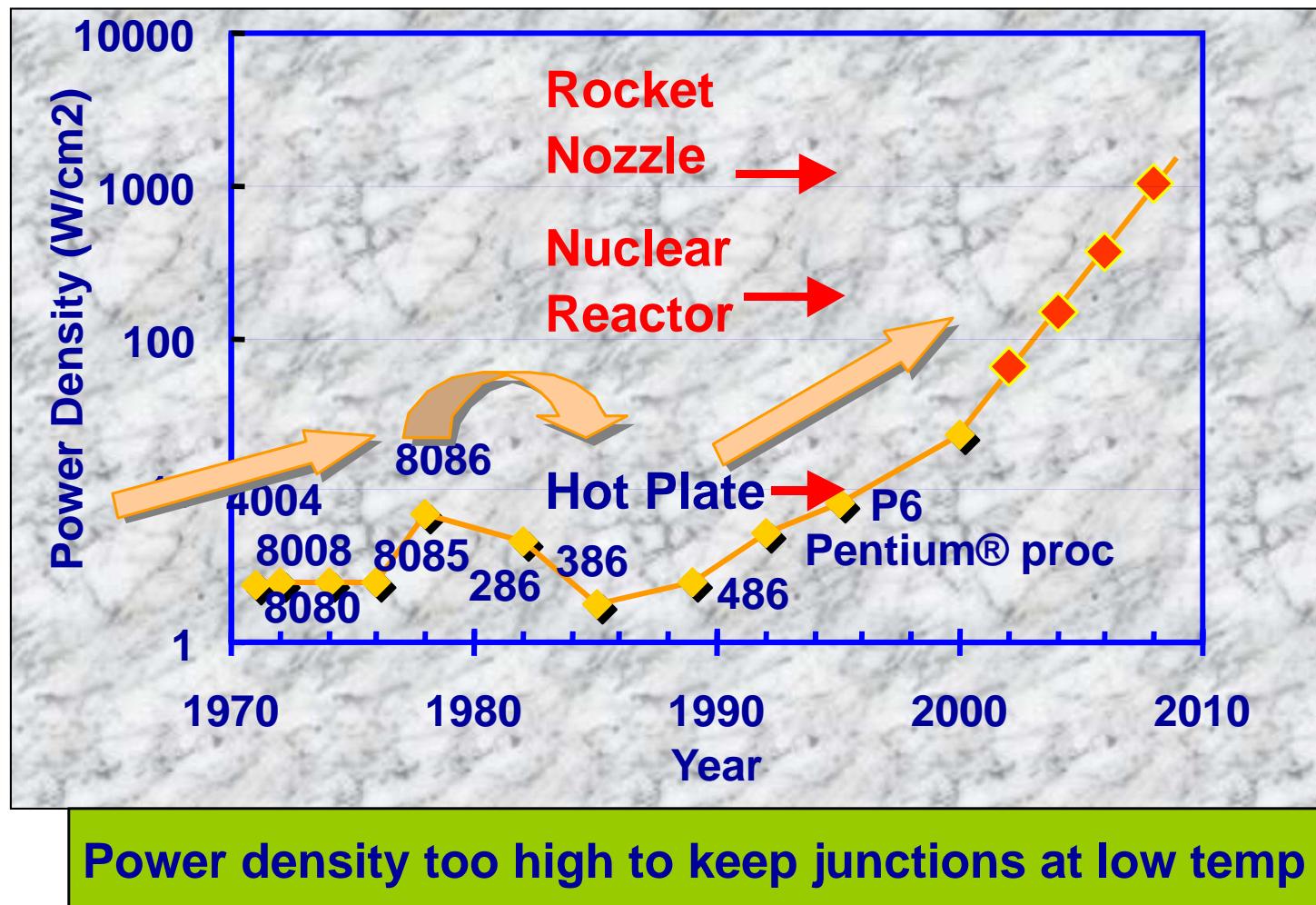


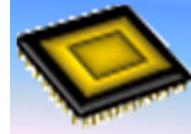
Power delivery and dissipation will be prohibitive



Chapter 1

Power density

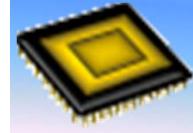




Chapter 1

Power Gap

Year of Production	2003	2006	2009	2012	2015	2018
Process Technology (nm)	101	90	65	45	32	22
Supply Voltage (V)	1.2	1	0.8	0.6	0.5	0.4
Clock Frequency (MHz)	300	450	600	900	1200	1500
Application (Maximum Required Performance)	➤ Still Image Processing		Real Time Video Codec (MPEG4/CIF)		Real Time Interpretation	
	➤ Web Brower ➤ Electronic Mailer ➤ Scheduler		➤ TV Telephone (1:1) ➤ Voice Recognition (Input) ➤ Authentication (Crypto Engine)		➤ TV Telephone (3:1) ➤ Voice Recognition (Operation)	
Processing Performance (GOPS)	0.3	2	14	77	461	2458
Required Average Power (W)	0.1	0.1	0.1	0.1	0.1	0.1
Required Standby Power (mW)	2	2	2	2	2	2
Battery Capacity (Wh/kg)	120	200	200	400	400	400



Chapter 1

Not Only Microprocessors

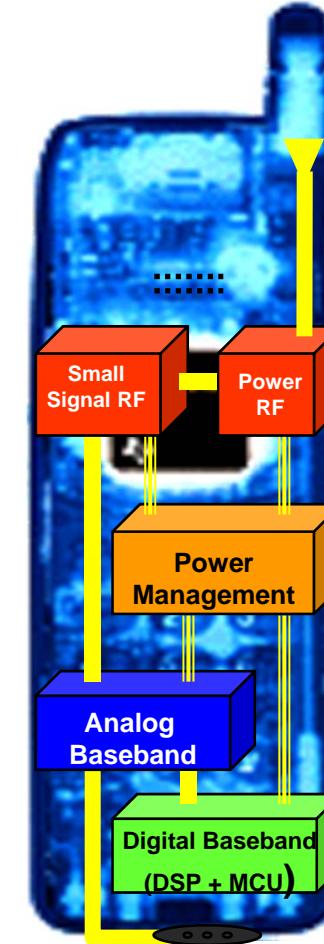
Cell
Phone

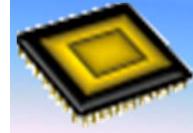


Digital Cellular Market
(Phones Shipped)

	1996	1997	1998	1999	2000
Units	48M	86M	162M	260M	435M

(data from Texas Instruments)





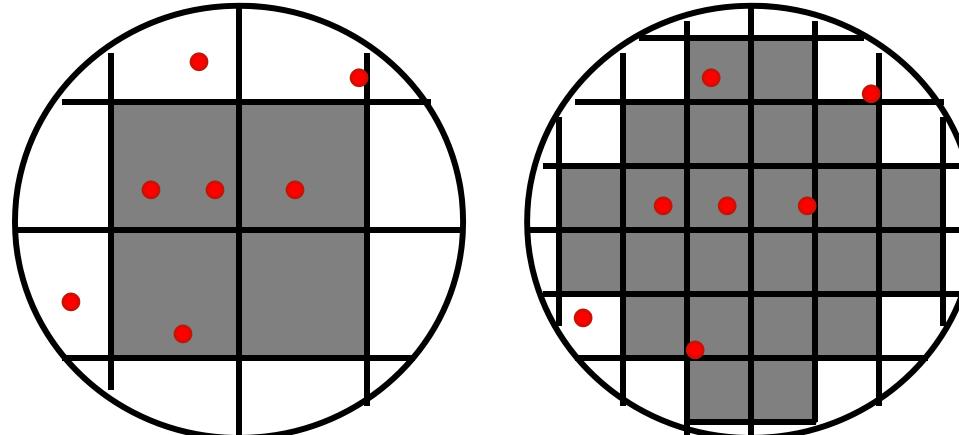
Chapter 1

Cost of Die

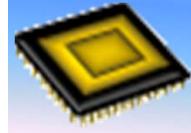
$$\text{variable cost} = \frac{\text{cost of die} + \text{cost of die test} + \text{cost of packaging}}{\text{final test yield}}$$

$$\text{cost of die} = \frac{\text{cost of wafer}}{\text{dies per wafer} \times \text{die yield}}$$

$$\text{dies per wafer} = \frac{\pi \times (\text{wafer diameter}/2)^2}{\text{die area}} = \frac{\pi \times \text{wafer diameter}}{\sqrt{2} \times \text{die area}}$$



$$\text{die yield} = (1 + (\text{defects per unit area} \times \text{die area})/\alpha)^{-\alpha}$$



Chapter 1

Cost of Integrated Circuits

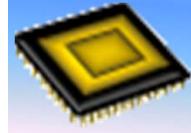
➤ Yield Example

- wafer size of 12 inches, die size of 2.5 cm^2 , 1 defects/ cm^2 ,
 $\alpha = 3$ (measure of manufacturing process complexity)
- 252 dies/wafer (remember, wafers round & dies square)
- die yield of 16%
- $252 \times 16\% =$ only 40 dies/wafer die yield !

➤ Die cost is strong function of die area

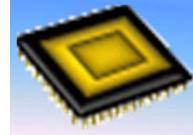
- proportional to the third or fourth power of the die area

$$\text{cost per IC} = \text{variable cost per IC} + \frac{\text{fixed cost}}{\text{volume}}$$



Chapter 1

- 1-1 The Integrated Circuit (IC) Era
- 1-2 VLSI Design Strategies
- 1-3 CMOS Design Options
- 1-4 MOS Transistor
- 1-5 MOS Transistor Switches
- 1-6 CMOS Logic
- 1-7 CMOS Binary Logic Circuits
- 1-8 Circuit and System Representations
- 1-9 CMOS Attributes

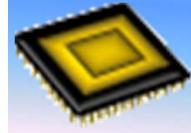


Chapter 1

1.1 The Integrated Circuit (IC) Era

• Microelectronics Evaluation

ERA (number of logic blocks per chip)	Date	Complexity	Product
Signal transistor	1959	Less tan 1	_____
Unit logic (one gate)	1960	1	diode
Multi - function	1962	2 - 4	Logic gates, FF
Complex function	1964	5 - 20	counter, adder multipliers
MSI	1967	2 - 200	8-bit up, RAM
LSI	1972	200 - 2000	16 & 32 bits up peripherals
VLSI	1978	2000 - 20000	Special and Real time processor
ULSI	1989	20000 - ?	?



Chapter 1

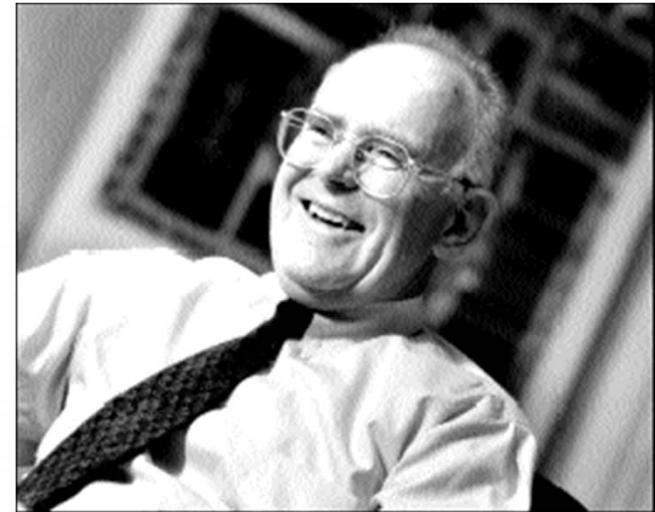
1.1 The Integrated Circuit (IC) Era

- **Moore's Law**

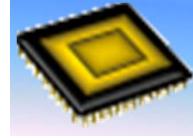


2X transistors/Chip Every 1.5 years Called “_____”

Microprocessors have become smaller, denser, and more powerful.



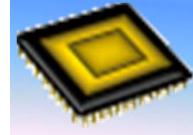
Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.



Chapter 1

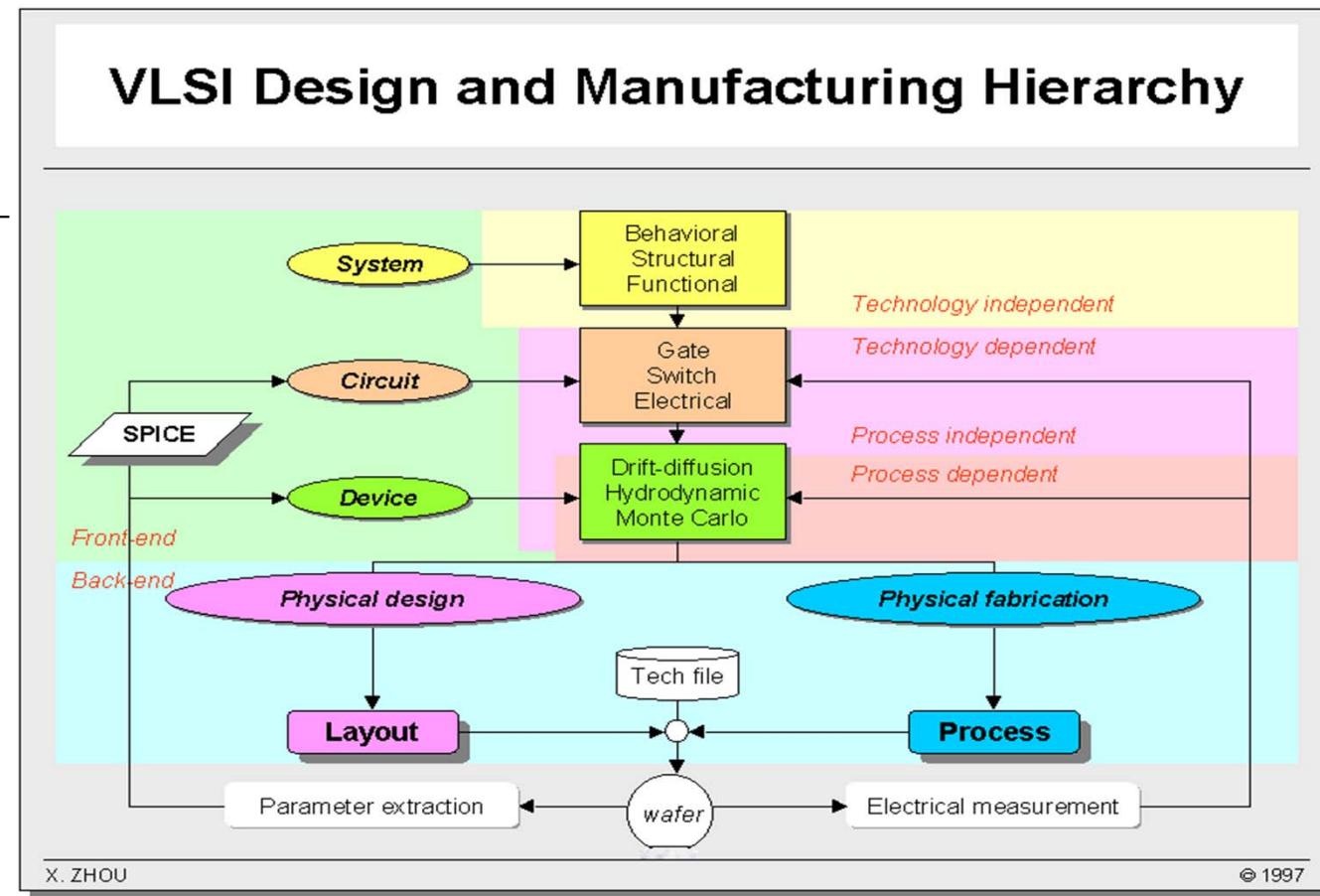
1.2 VLSI Design Strategies

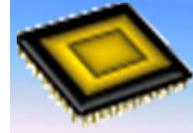
- Design parameters may be summarized in terms of
 - _____
 - _____
 - _____
 - _____
- Design is a trade-off to achieve adequate results for all of the above parameters
- Structure Design Strategies
 - _____
 - _____
 - _____
 - _____



Chapter 1

1.2.1 Hierarchy



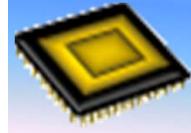


Chapter 1

1.2.1 Hierarchy

- The successive decomposition of the problem into sub problems as well as of the synthesis of small elements into larger functional elements shows the hierarchical nature of hardware design process

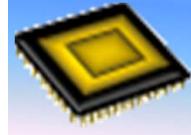
- The actual design process is combination of both top-down decomposition and bottom-up implementation with a large amount of feedback between the different level



Chapter 1

Top-down vs. Bottom-up design

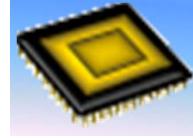
- **Top-down design** adds functional detail.
 - Create lower levels of abstraction from upper levels.
- **Bottom-up design** creates abstractions from low-level behavior.
- Good design needs **both** top-down and bottom-up efforts.



Chapter 1

1.2.2 Regularity

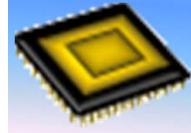
- The design attempts to divide the hierarchy into a set of similar building block, however, it may be made of regular structures to simplify the design process.
 - For instance, the interface between modules (power, ground, clock, busses) might be common but the internal details of modules may differ according to function
- Regularity can exist at all level of the design hierarchy



Chapter 1

1.2.3 Modularity

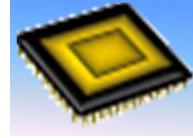
- For a good VLSI architecture, it should be implemented by only a few different types of simple cell
 - ※ easily design and check



Chapter 1

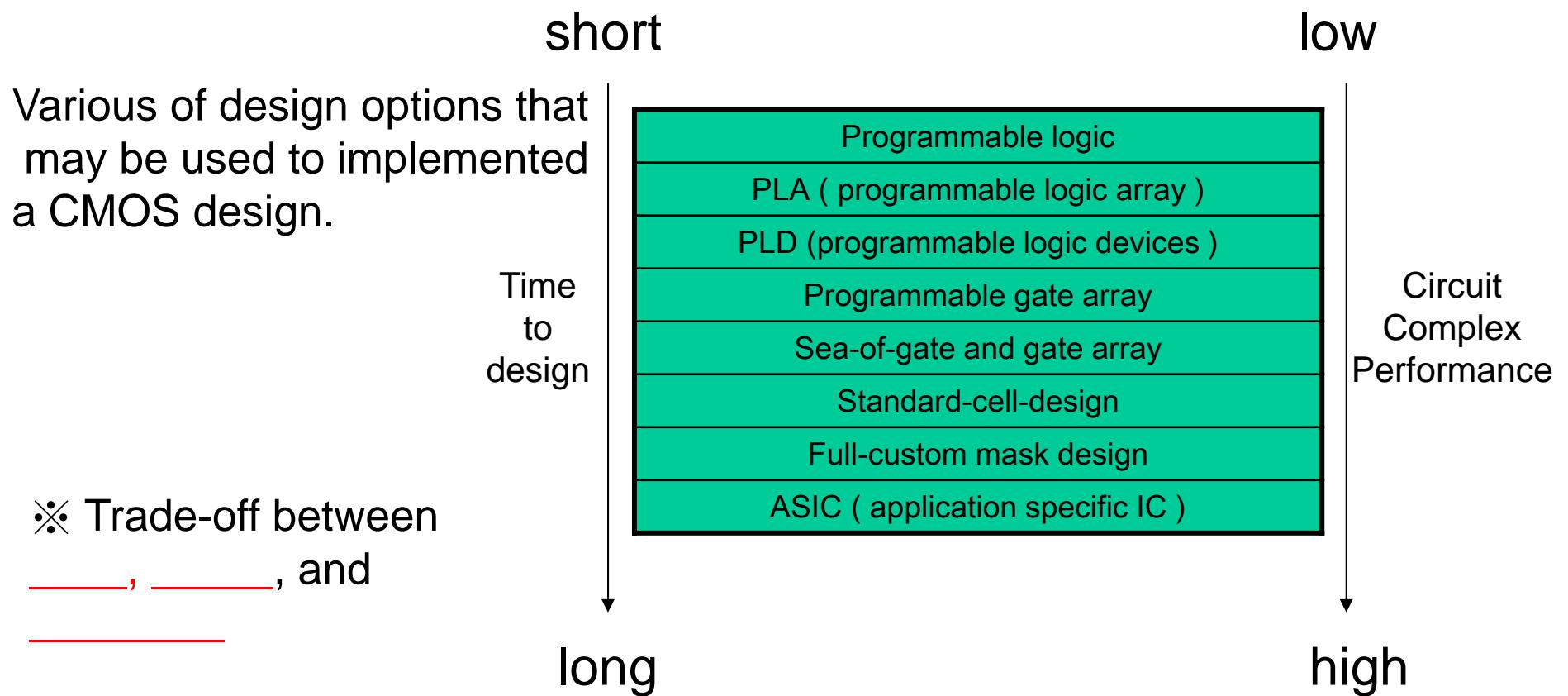
1.2.4 Locality

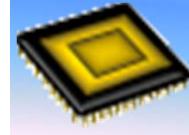
- For a good hardware design, a well-characterized interfaces for modules is that the modules can be connected by a network with local interconnection
- Locality has come to mean “ time locality ”, and hence synchronous –timing between local cells
- In the software world this is parallel by the reduction of global variables to minimum (hopefully to zero)



Chapter 1

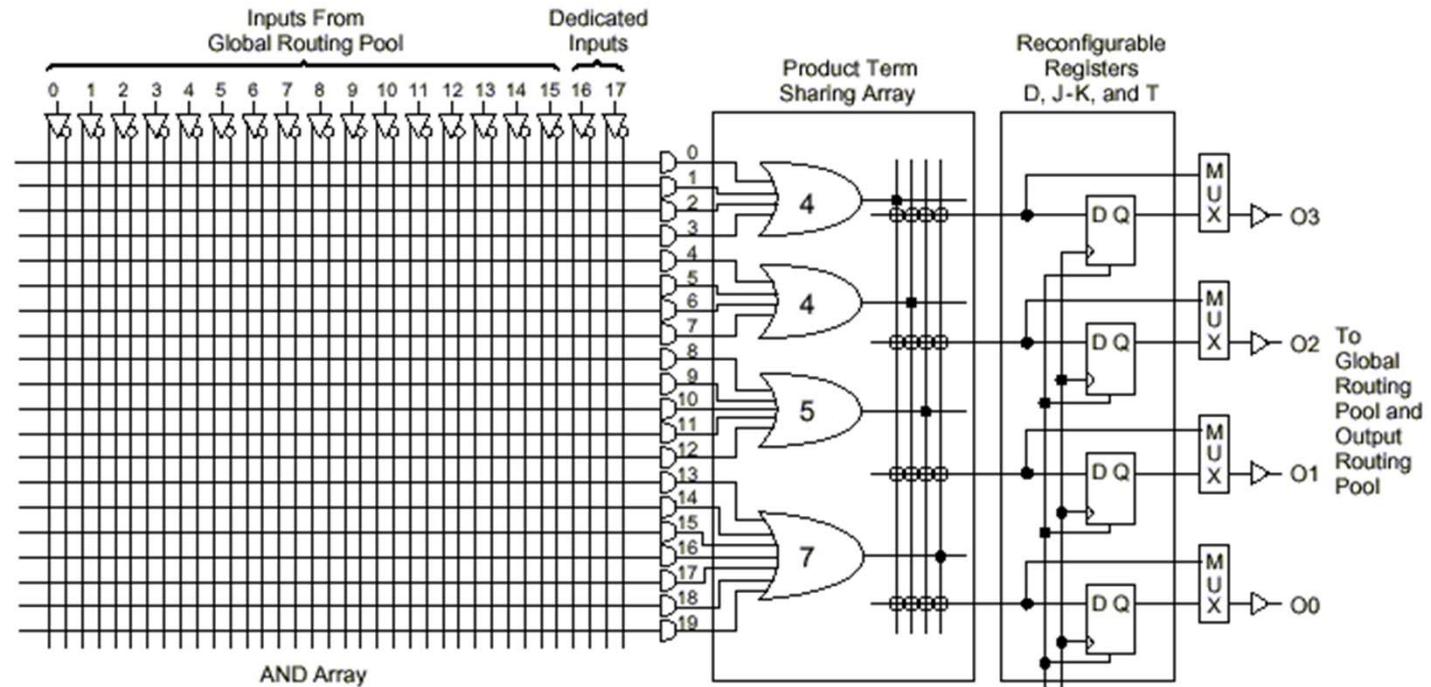
1.3 CMOS Chip Design Options





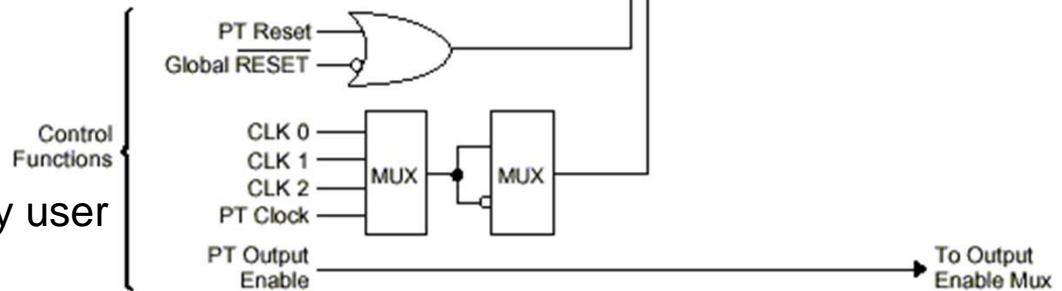
Chapter 1

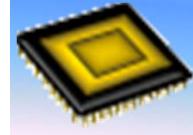
1.3.1 Programmable Logic



This part is the ISPLSI1016E-100LJ
by Lattice Semiconductor.

- ※ Easier to program the product terms by user
- ※ Limited digital function

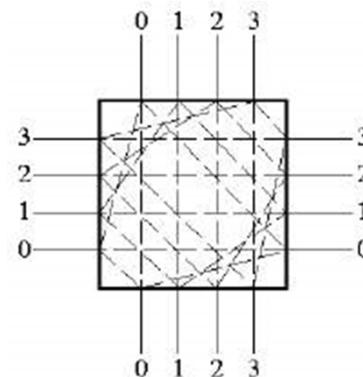




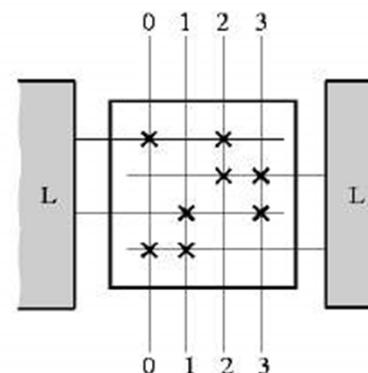
Chapter 1

1.3.2 Programmable Gate Array

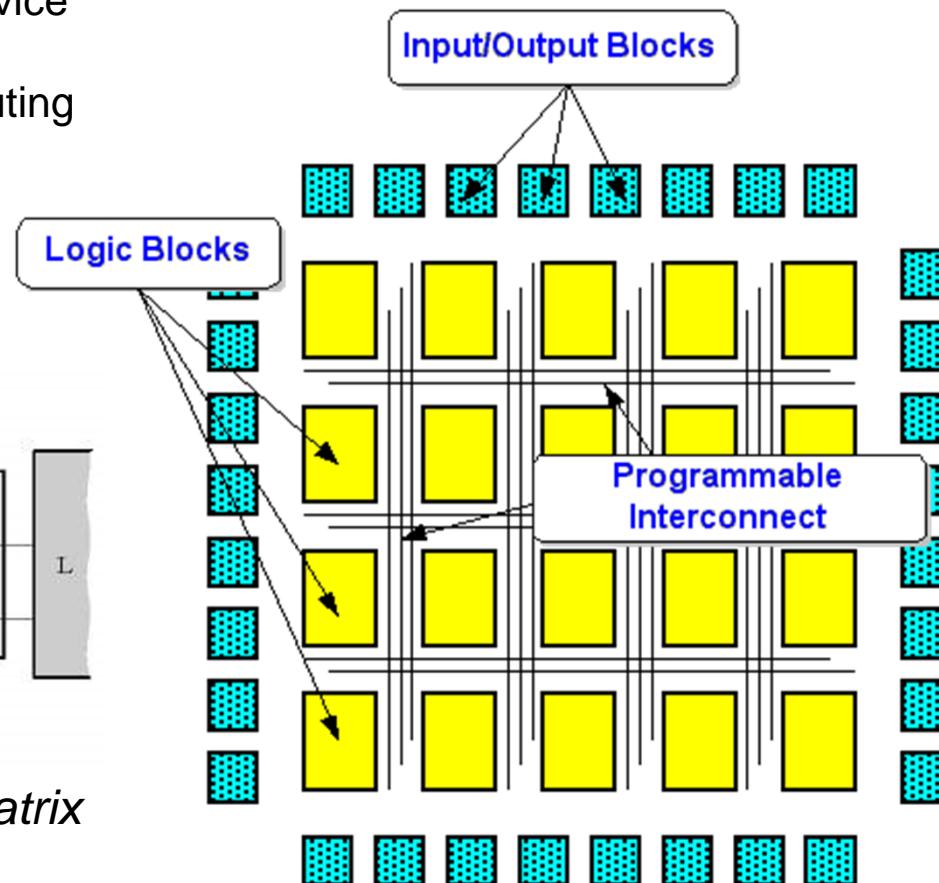
- ※ A further class programmable device
- ※ More flexibility
- ※ Logic gate are surrounded by routing channel and switching matrix
- ※ Flexible I/O buffers and I/O pads connections

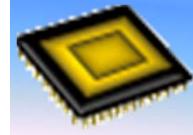


Switching Matrix



Channel Matrix

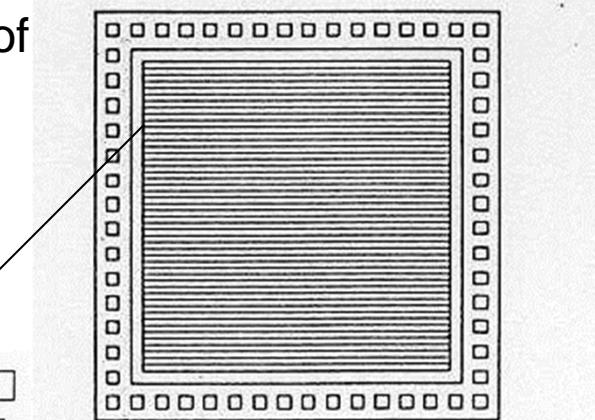
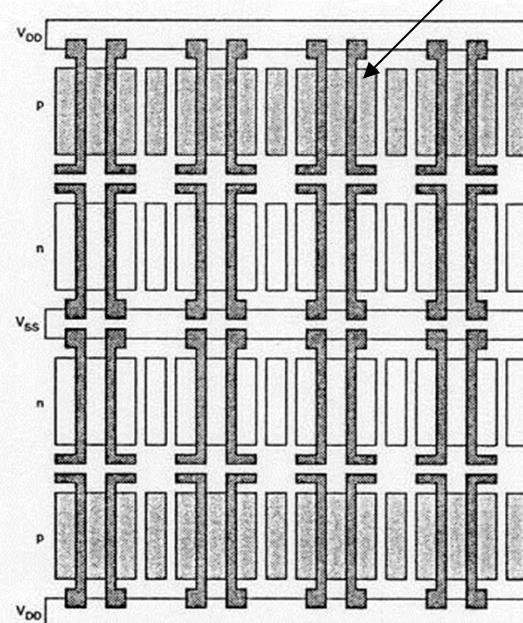




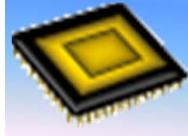
Chapter 1

1.3.3 Sea-of Gate and Gate Array Design

- ※ The core of the chip contains a continuous array of n- and p-transistors
- ※ The fixed array of transistors is customized by personalizing the metal interconnection layers
(Metal Mask is required)
- ※ 60% ~ 80% Gate utilization
- ※ CELL Structure

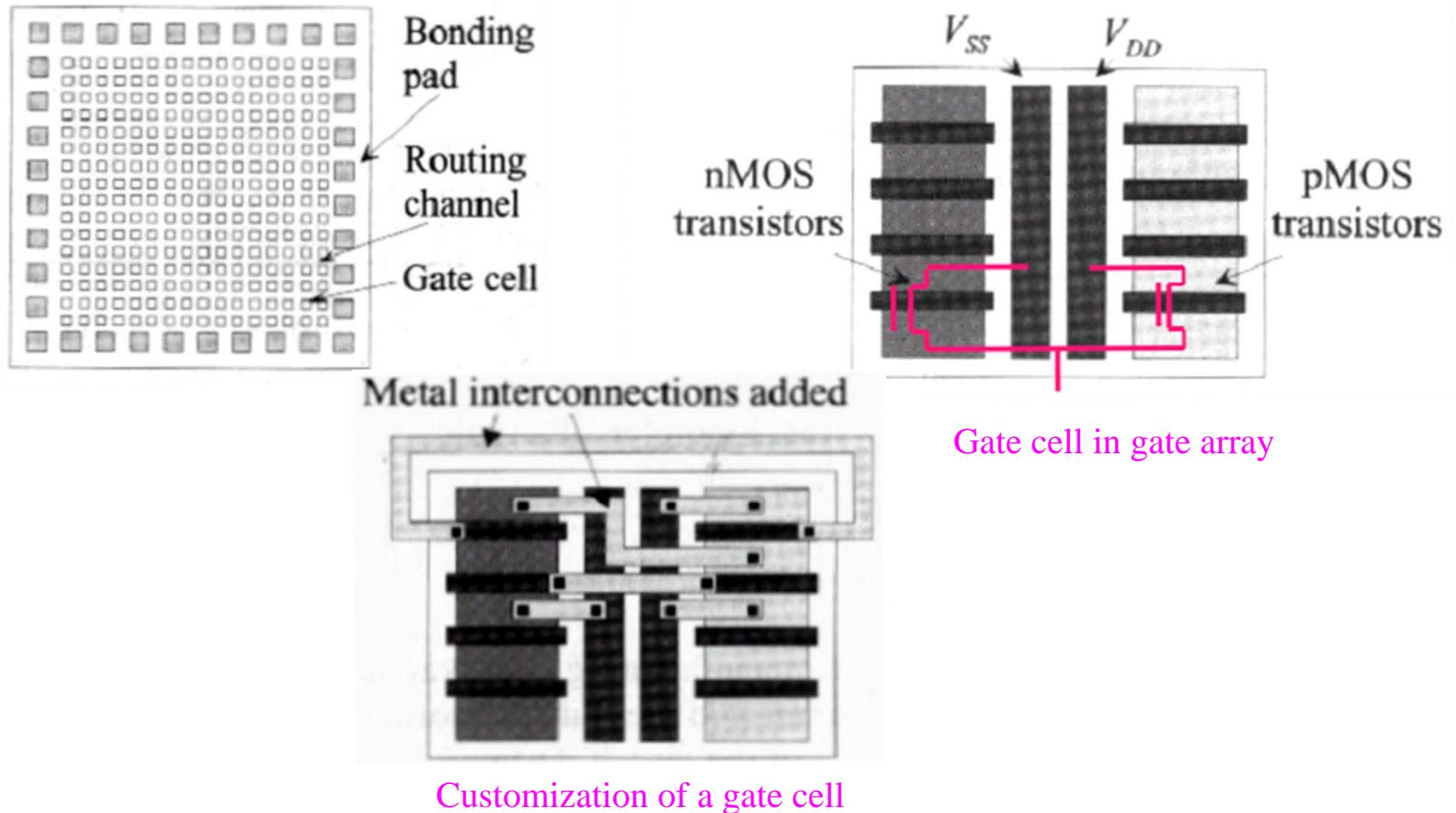


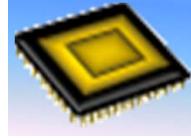
flexibility in channel definition (position & width)
over-the-cell routing
higher packing density
RAM-compatible
supports variable-height cells & macrocells
now universally used



Chapter 1

1.3.3 Sea-of Gate and Gate Array Design

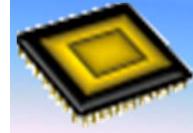




Chapter 1

1.3.4 Standard-Cell design

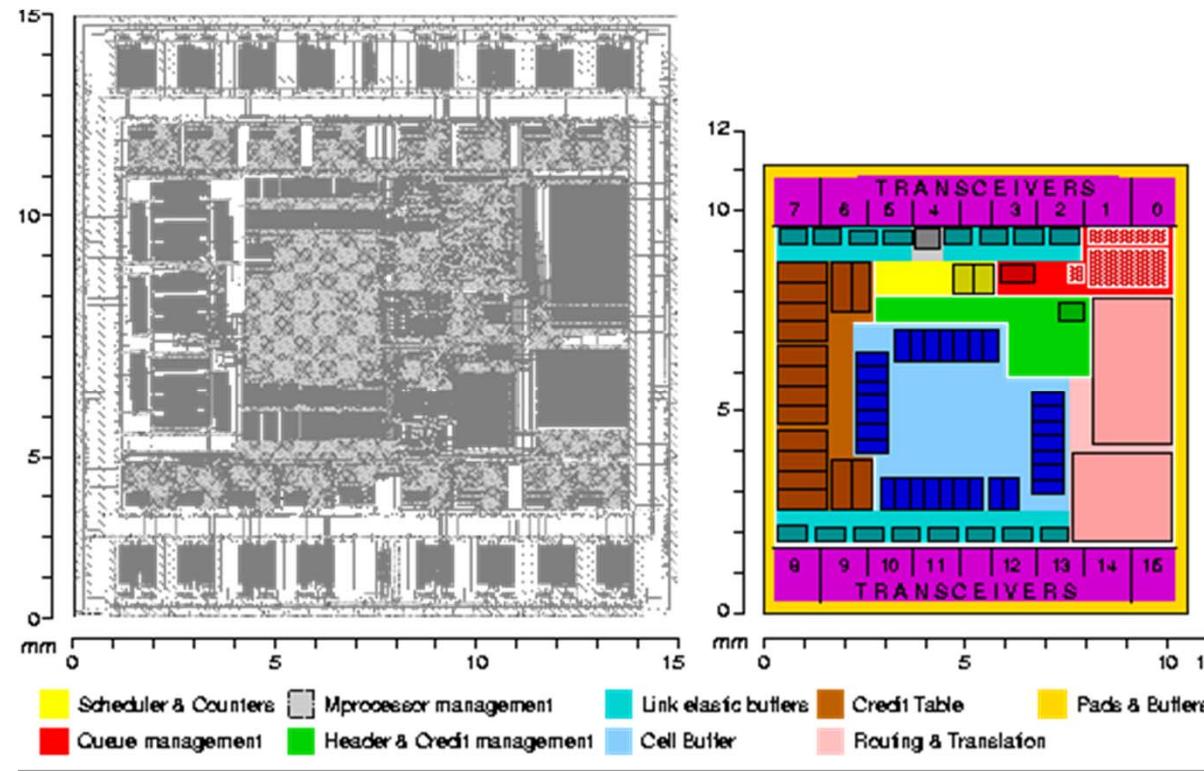
- Similar to design with **standard logic** building blocks: e.g. TTL, CMOS, RAMs, ROMs, PLAs, UP, ...
- Difference is building blocks are integrated on silicon, not on a PC board.
- A standard-cell IC is a circuit that is made up of customer selected functions (**standard cell**) with which are custom interconnected on silicon to implement desired circuit function

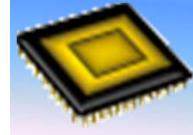


Chapter 1

1.3.4 Standard-Cell design

- Complete IC fabrication procedures (masks)are required to fabricate the standard-cell IC

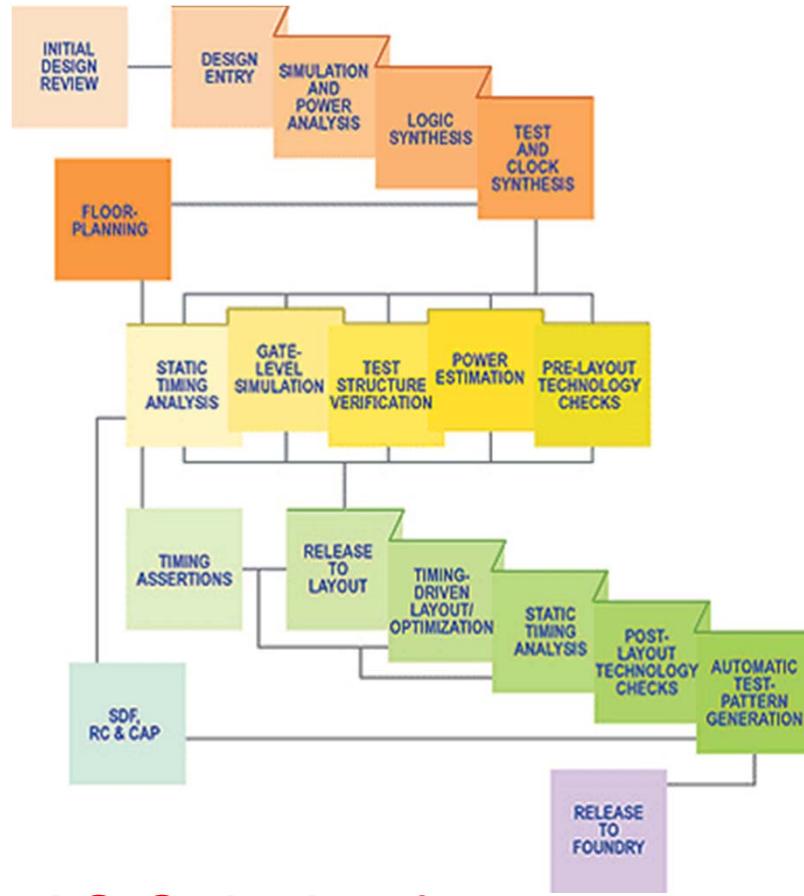




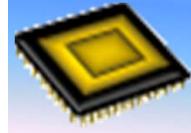
Chapter 1

1.3.5 Full-Custom mask design

- ASIC
 - Function, layout, and **every transistor is optimized**
- Contains each step of hierarchy design procedure
- **High flexibility, high performance, long time to design**



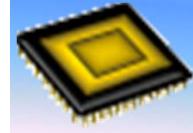
ASIC design flow



Chapter 1

1.4 MOS Transistor

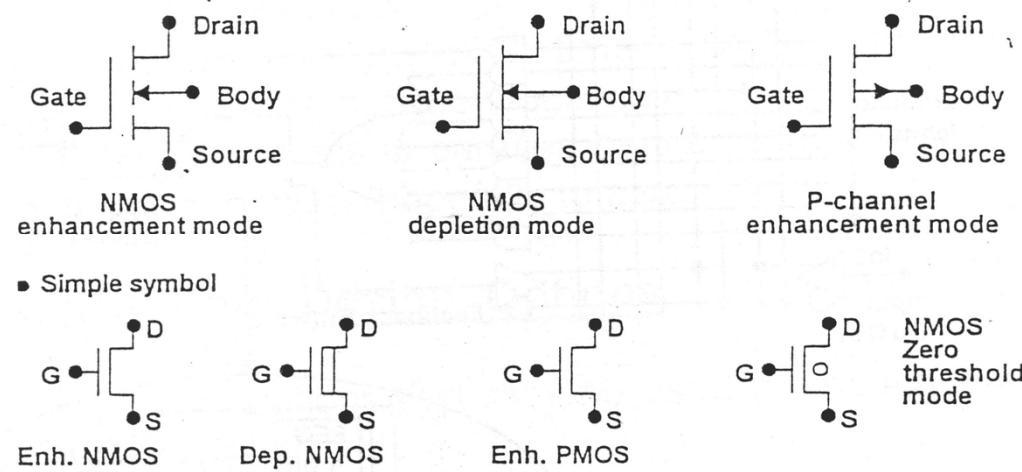
- Metal-Oxide-Semiconductor (MOS)
Field-Effect-Transistor (MOSFET)
 - N-channel MOSFET : _____
 - P-channel MOSFET : _____



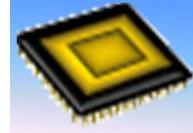
Chapter 1

1.4.1 Fundamentals of MOS transistor Structure

1. Symbol for MOS



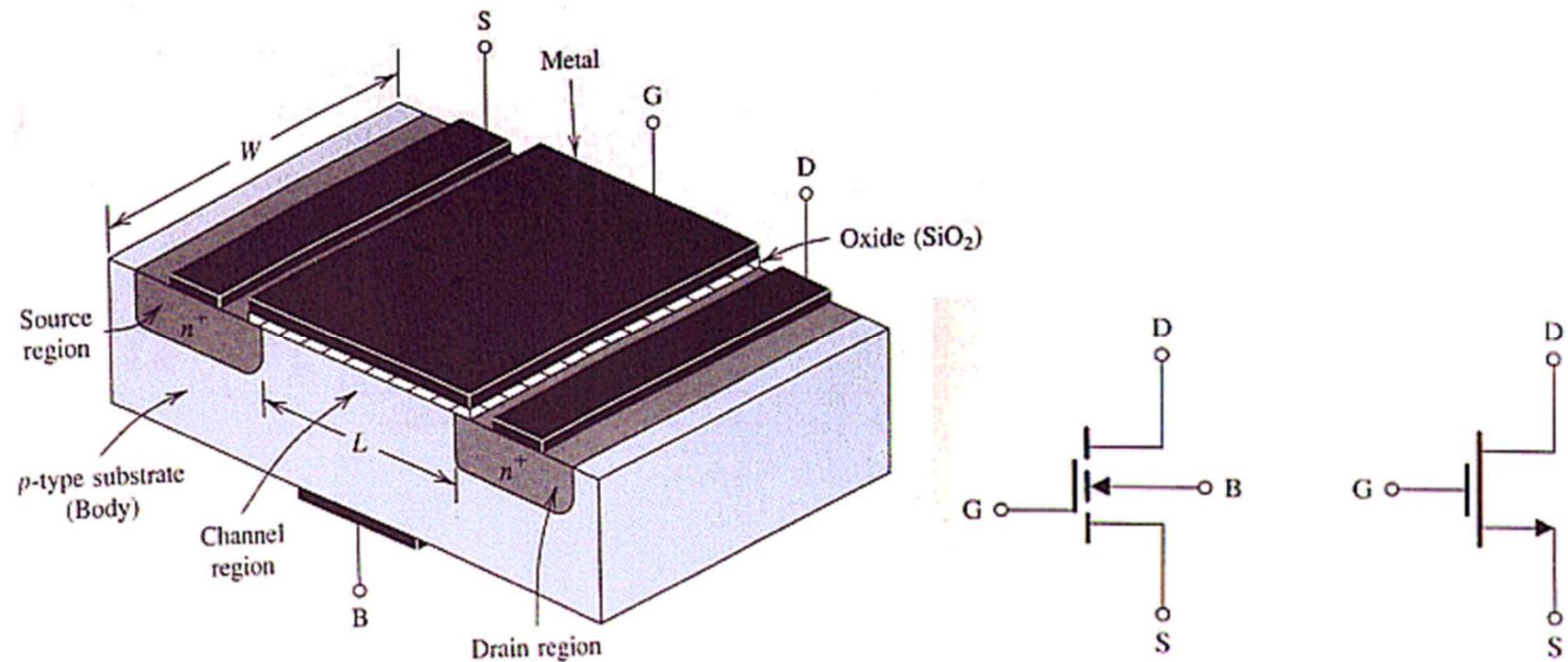
⌘ IEEE standard

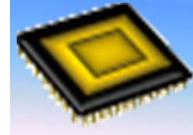


Chapter 1

1.4.1 Fundamentals of MOS transistor Structure

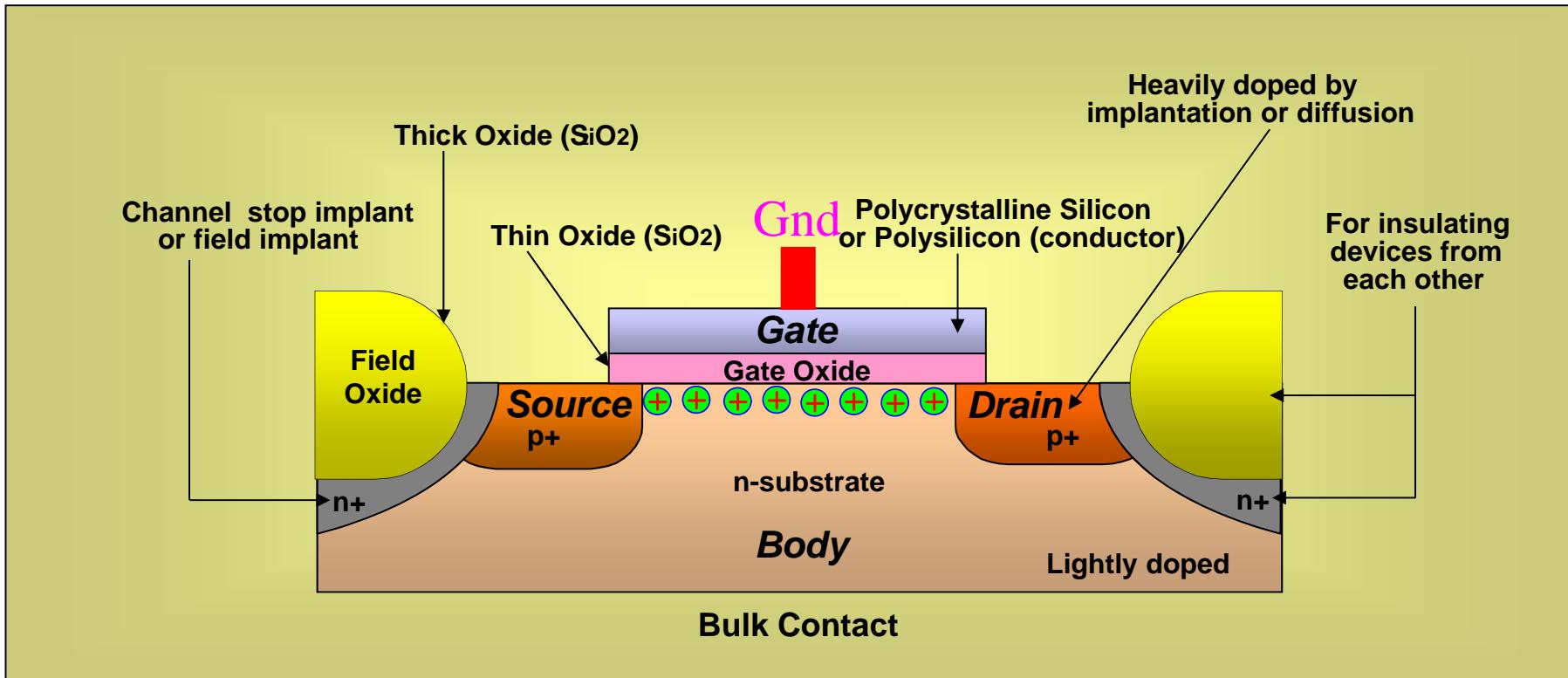
2. MOS Transistor Physical Structure





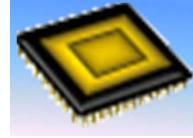
Chapter 1

PMOS Transistor Physical Structure



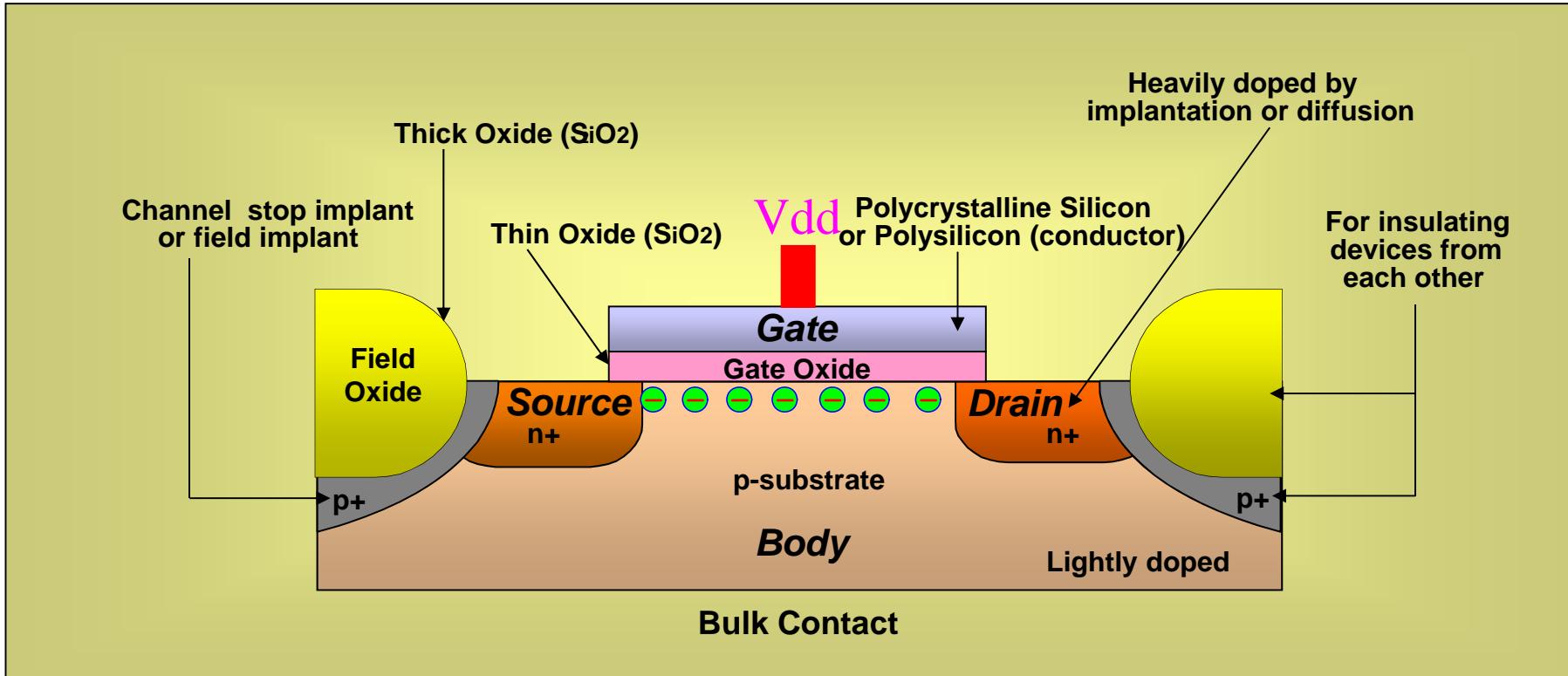
Unipolar and symmetric device with four terminals of Gate, Source, Drain, and Body

PMOS: p-type Source and Drain, n-type Body connected to power supply (V_{DD})



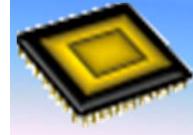
Chapter 1

NMOS Transistor Physical Structure



Unipolar and symmetric device with four terminals of Gate, Source, Drain, and Body

NMOS: n-type Source and Drain, p-type Body connected to ground (GND)

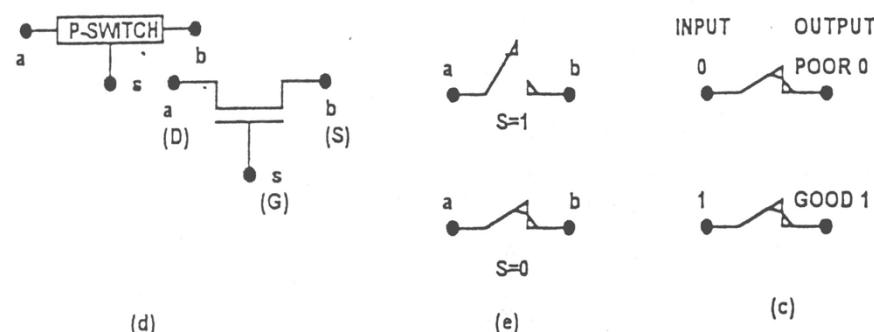
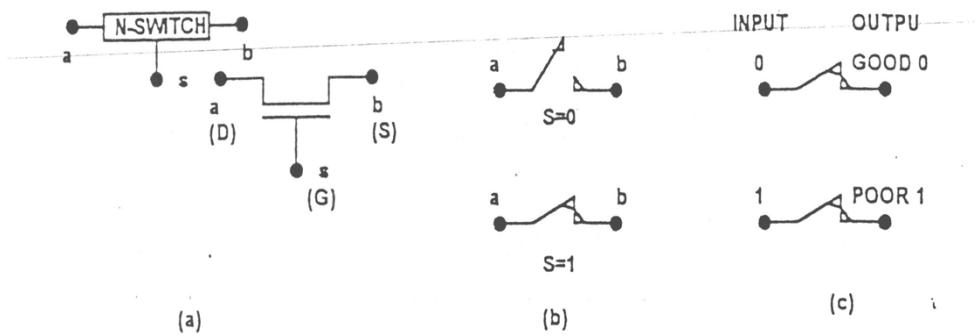


Chapter 1

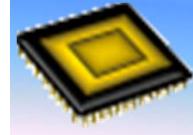
1.5 MOS transistor switches

➤ The **gate** controls the passage of **current** between the **source** and the **drain** → it allows the MOS transistor to be viewed as **simple on/off switches**

➤ Assume logic value “**1**” is **high voltage** (called power or VDD) ; logic value “**0**” is **zero volts** (GND)



MOS transistors viewed as switches

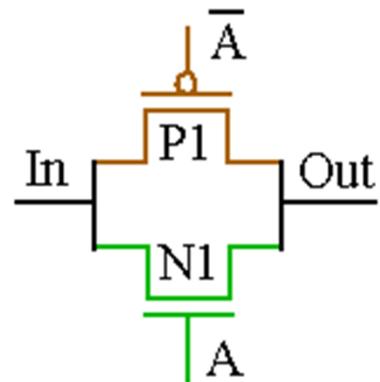


Chapter 1

1.5 MOS transistor switches

➤ Complementary switch

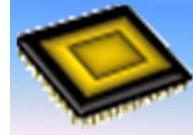
- Complementary switch is also called a _____ or _____



One pMOS and one nMOS in parallel.

Note that neither transistor is connected to V_{DD} or GND.

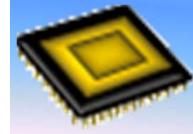
A and \bar{A} control the transmission of a signal on *In* to *Out*.



Chapter 1

➤ Basic Logic Library

Name	Graphic	Functional Expression	Cost Transistors
Invert		$C = \bar{A}$	2
Driver		$C = A$	4
AND		$C = AB$	6
OR		$C = A + B$	6
NAND		$C = \overline{AB}$	4
NOR		$C = \overline{A + B}$	4
XOR		$C = A \oplus B$ $C = \overline{A}B + A\overline{B}$	14
XNOR		$C = \frac{A \circ B}{A \oplus B}$ or $C = \overline{A}B + A\overline{B}$ $C = AB + A\overline{B}$	12

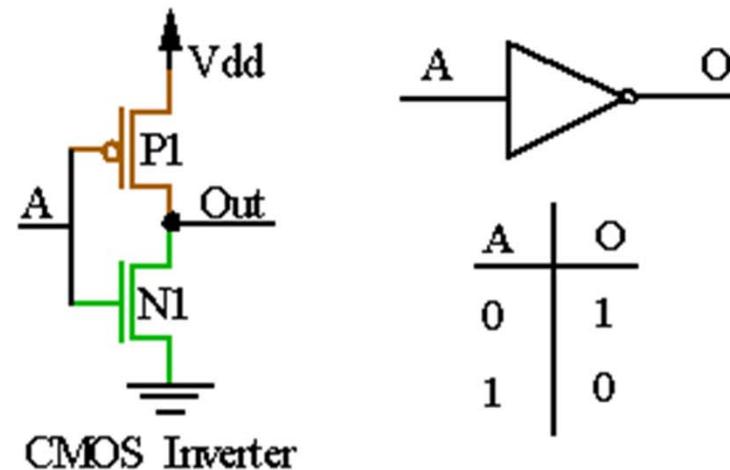


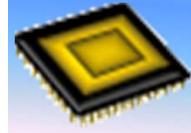
Chapter 1

1.6 CMOS Logic

➤ 1.6.1 CMOS Inverter

- One of the MOS (PMOS or NMOS) transistor is turned-on, the other is turn-off

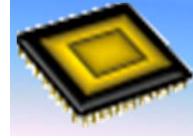




Chapter 1

1.6.1 CMOS Inverter (con't)

- The output node has strong “1” (VDD) or strong “0” (VSS) logic value when input value known
- An unknow input may make an unknow output → crowbarred level → PMOS and NMOS are simultaneously turn on → causes static power to be dissipated



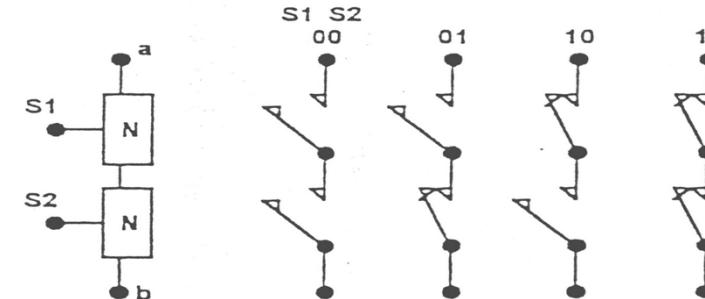
Chapter 1

1.6.2 Combinational logic

- Series CMOS switch combination

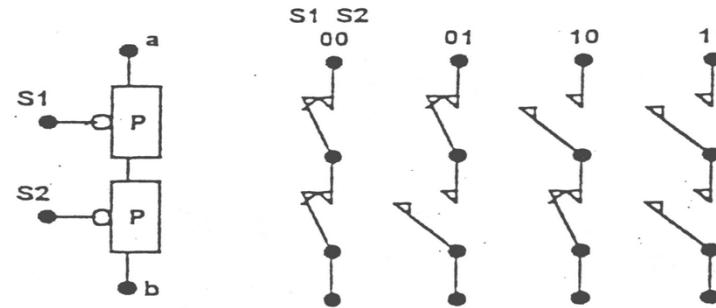
Series NMOS as
the AND switch

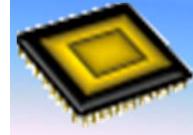
F	S1	
0	0	1
S2	OFF	OFF
1	OFF	ON



Series PMOS as
the NOR switch

F	S1	
0	0	1
S2	ON	OFF
1	OFF	OFF





Chapter 1

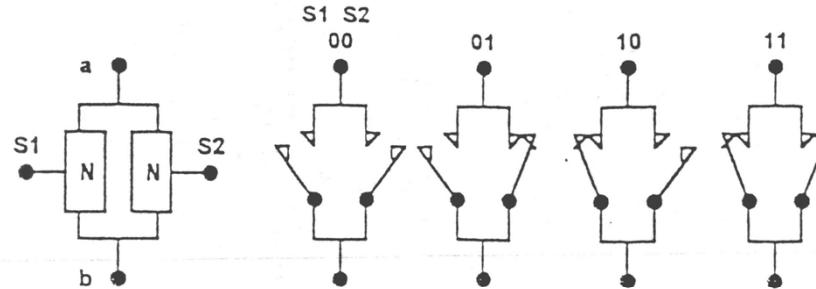
1.6.2 Combinational logic

- Parallel CMOS switch combination

- ♦ Parallel CMOS switch combination

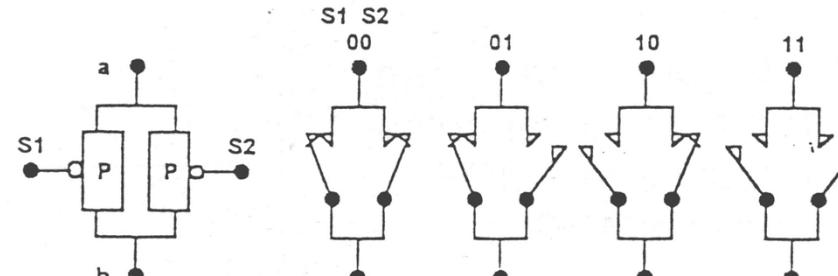
Parallel NMOS as the OR switch

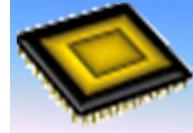
F	S1	0	1
S2	0	OFF	ON
1	ON	ON	



Parallel PMOS as the NAND switch

F	S1	0	1
S2	0	OFF	ON
1	ON	ON	

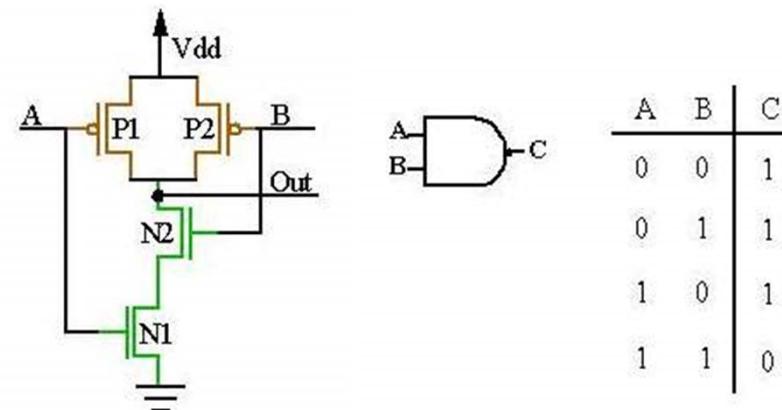
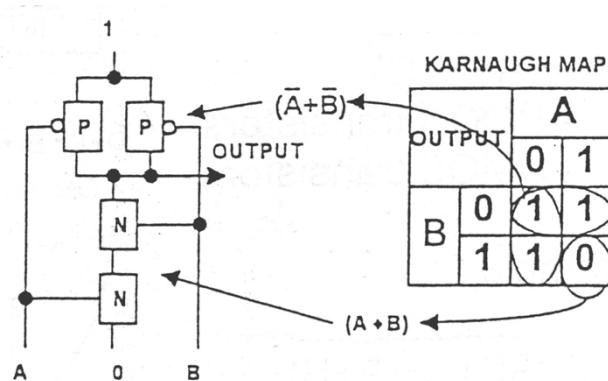


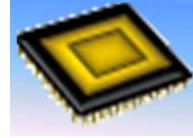


Chapter 1

1.6.3 The CMOS NAND Gate

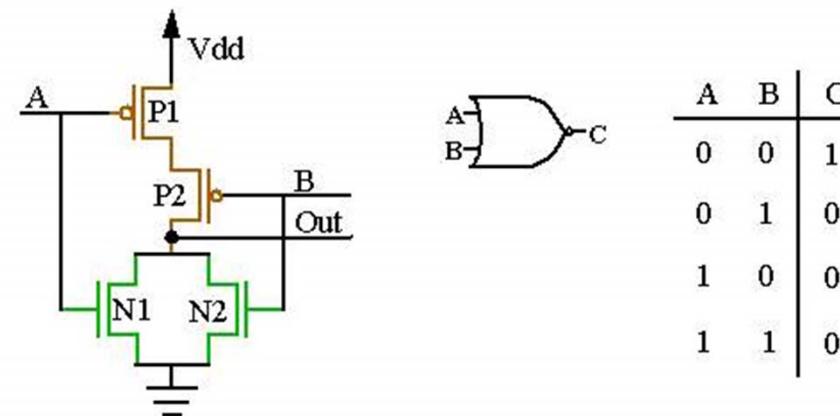
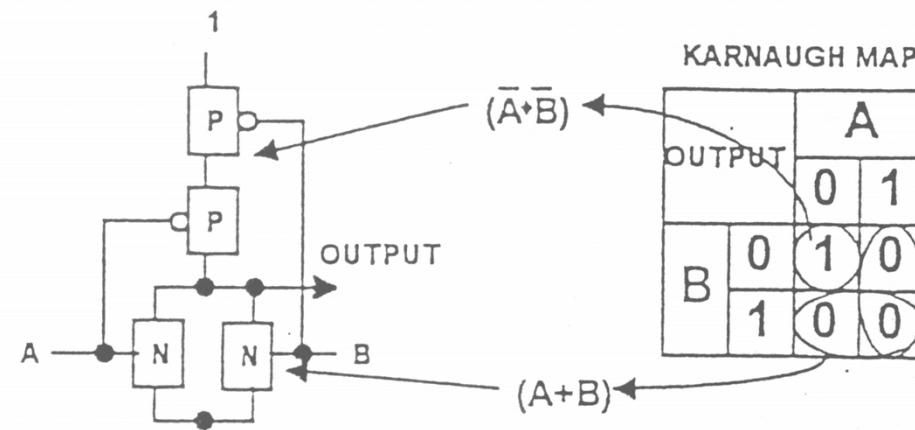
- Parallel PMOS transistors are connected between VDD and output as the NAND switch.
- Series NMOS transistors are connected between output and VSS as the AND switch (transition logic value is “0”)

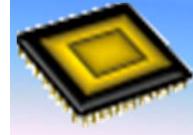




Chapter 1

1.6.4 The CMOS NOR Gate





Chapter 1

1.6.5 Compound Gate

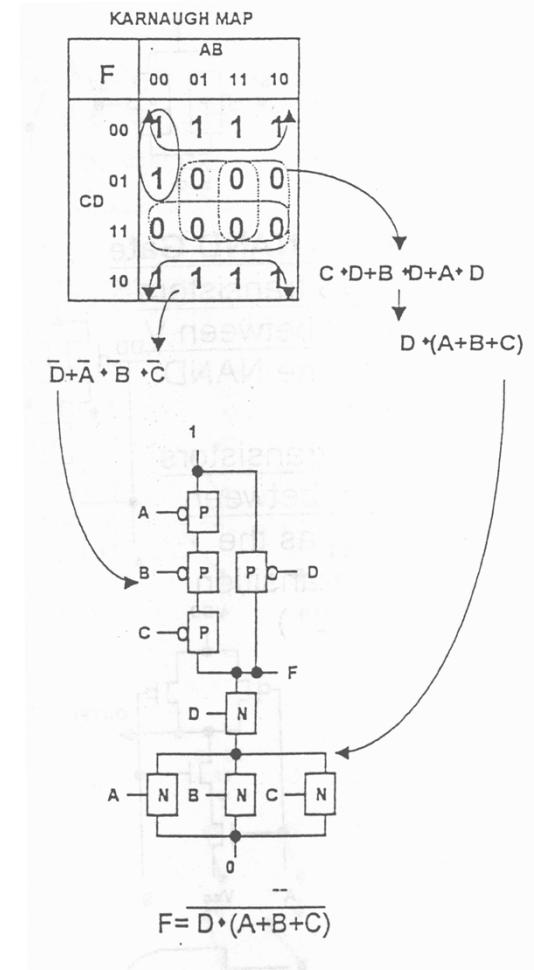
➤ For m input variables, it require m PMOS and m NMOS

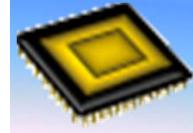
➤ Example 1:

- $F = D \cdot (A + B + C)$
- 4 PMOS transistors
- 4 NMOS transistors

➤ Example 2:

- $F = AB + C \cdot (A + B)$
- 5 PMOS transistors
- 5 NMOS transistors

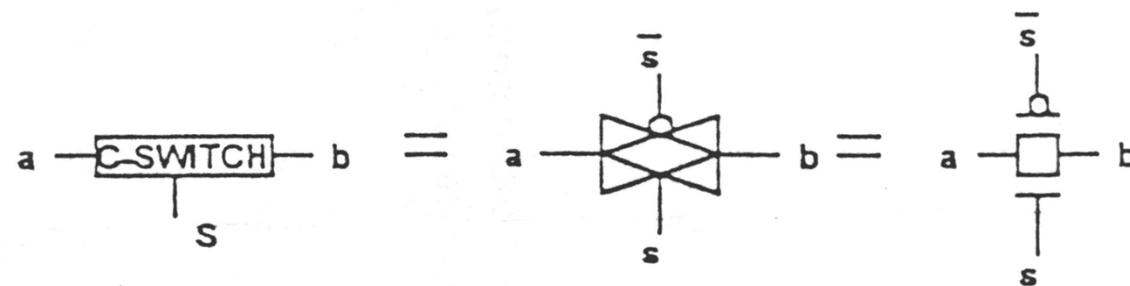
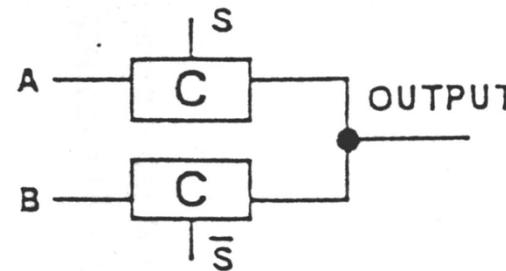


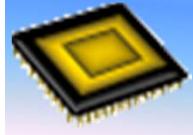


Chapter 1

1.6.6 Multiplexer

- Complementary switches may be used to select between a number of inputs, thus forming a multiplexer function





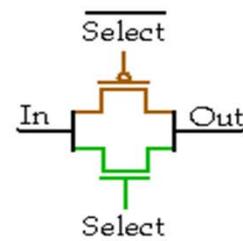
Chapter 1

1.6.6 Multiplexer

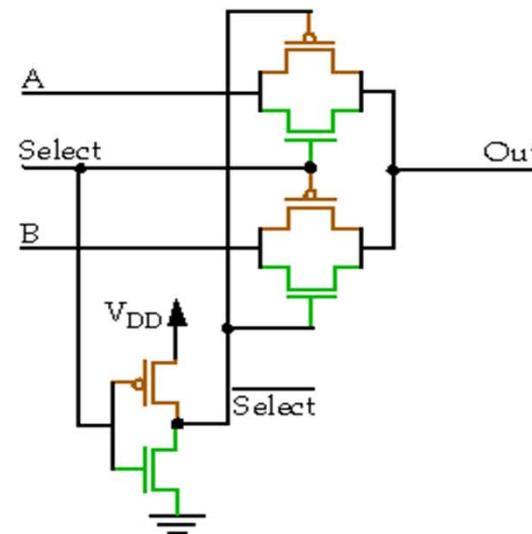
➤ Example:

- 2-input multiplexer

Transmission Gate



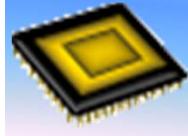
2-to-1 MUX



Truth Table for 2-to-1 MUX

Select	Out
0	B
1	A

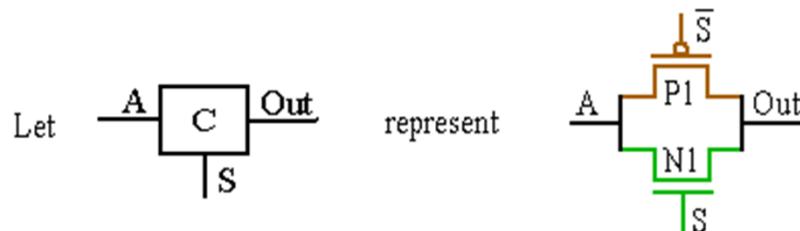
$$\text{Out} = A \cdot S + B \cdot \bar{S}$$



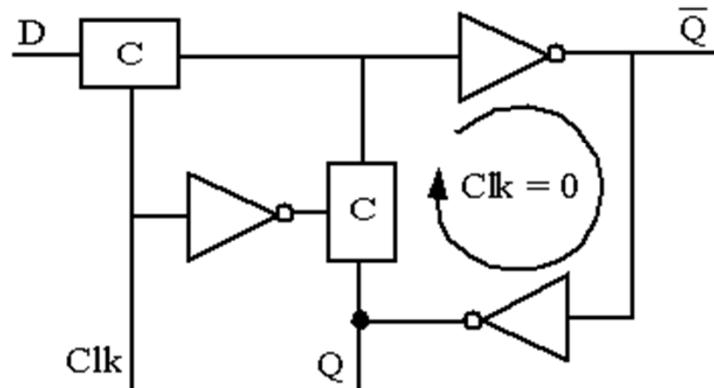
Chapter 1

1.6.7 Memory – Latch and Register

Example: D-latch



A positive *level-sensitive* latch:



State of the output is dependent on the level of the clock.

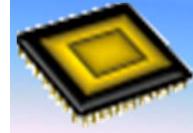
Although \bar{S} is not given in the “black box” abstraction, it must be routed to the pass gate.

The S “black box” terminal connects to the n-channel device.

When $\text{Clk} = 1$, Q set to D and \bar{Q} set to \bar{D}

When $\text{Clk} = 0$, D is ignored,
Feedback path is established.

Note: Other notations for \bar{D} :
-D or DN or D.L.

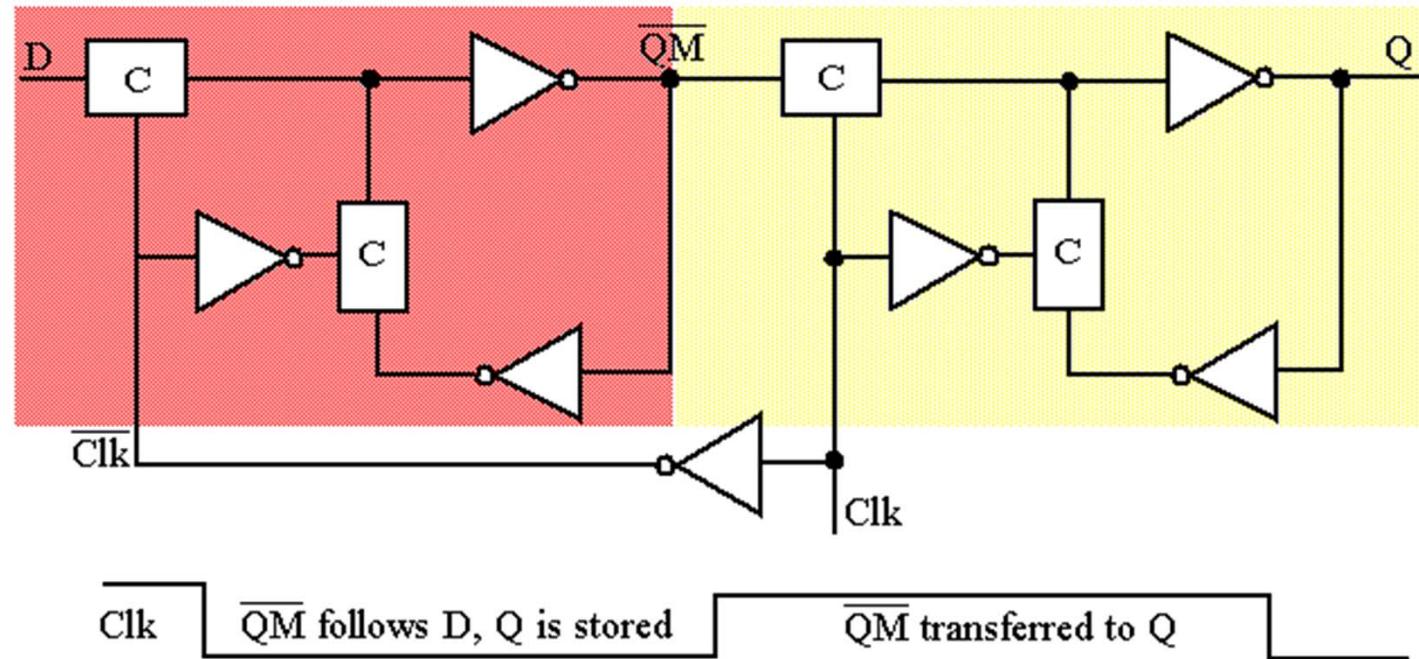


Chapter 1

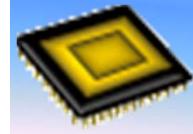
1.6.7 Memory – Latch and Register

Example: D Flip-Flop

Combine one negative (master) and one positive (slave) *level-sensitive* latch.

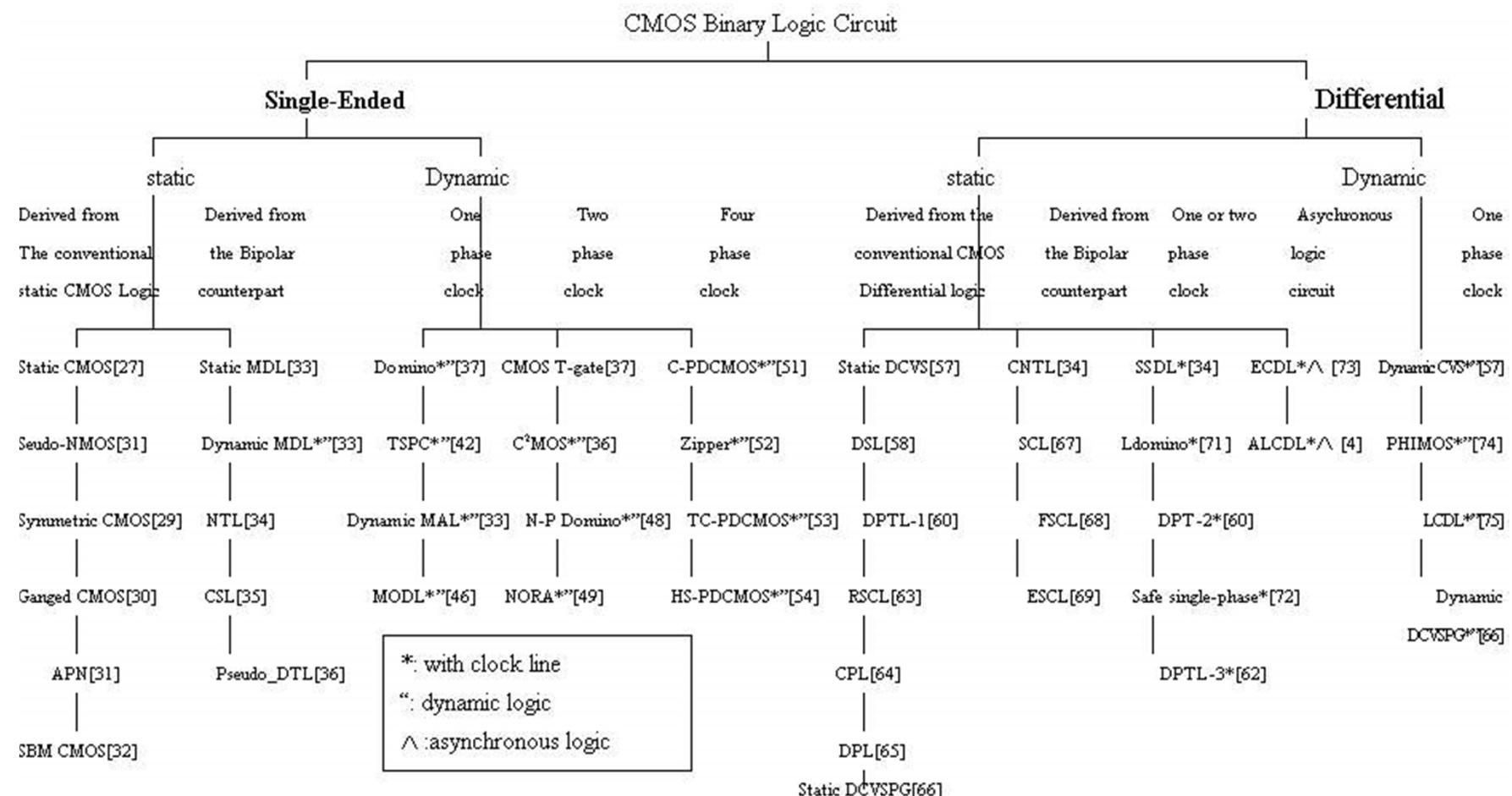


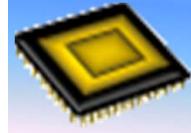
Forms the basis of most CMOS storage elements (EXCEPTIONS: RAM and ROM).



Chapter 1

1.7 CMOS Binary Logic Circuits

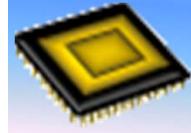




Chapter 1

1.7 CMOS Binary Logic Circuits

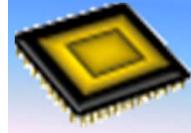
- The demand for high – performance CMOS VLSI circuits :
 - _____
 - _____
 - _____
- Logic Structure :
 - _____ : implement only one logic function at a **single output**
 - _____ : implement a given logic function and its complement simultaneously at **two output nodes**
 - _____ : without the clock control or with clock lines but **no floating output nodes** involved or no minimum clocking speed requirement
 - _____ : operated by storing charges on **parasitic capacitance** and require a lower bound on the speed of the **clock**



Chapter 1

1.7 CMOS Binary Logic Circuits

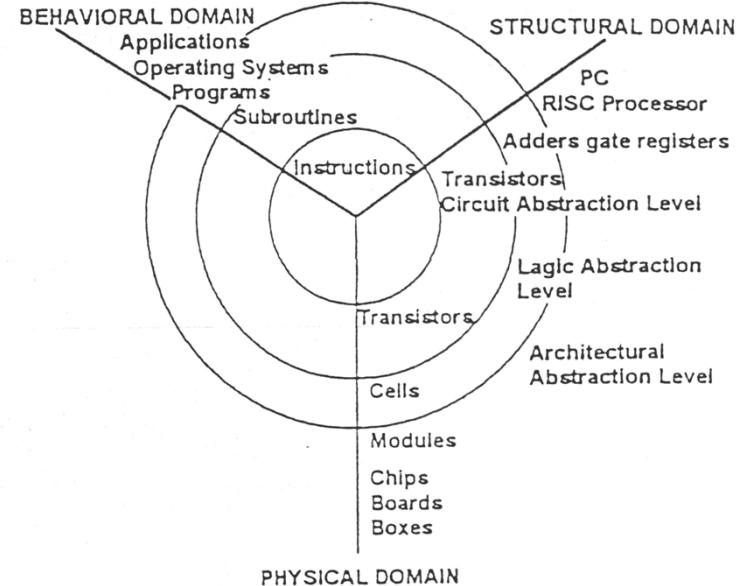
- The present-day CMOS binary logic circuits :
 - No “ ultimate” logic circuits can fit all the requirement
- New static binary logic circuits with good trade-off among speed, layout area, noise margins, and power dissipation
 - High-speed
 - Low power-delay product for complex logic gate
- New dynamic binary logic circuits :
 - True-single-phase pipelined system
 - No race and charge sharing problem
 - Clock slope insensitive
 - Reduce the clock loading
 - Improve the speed

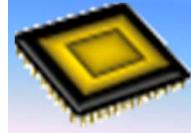


Chapter 1

1.8 Circuit and System Representations

- The _____ specifies what a particular system does
- The _____ specifies how entities are connect together to effect the prescribed behavior
- The _____ specifies how to actually build a structure that has the required connectivity to implement the prescribed behavior
- Each domain may be specified at a variety of levels

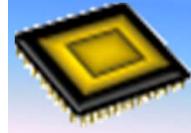




Chapter 1

1.9 CMOS Attributes

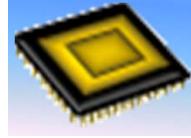
- For CMOS a brief summary of the main attributes are provided below :
 - **Fully restored Logic Levels** ; i.e. ,output settles at VDD or VSS
 - **Transition Time** – Rise and Fall time are of the same order
 - Memories are implemented both densely and with low power dissipation
 - **Transmission Gate** pass both logic levels well, allowing use of efficient, widely used logic structures such as multiplexers, latches, and registers



Chapter 1

1.9 CMOS Attributes

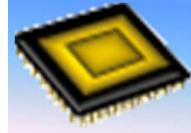
- **Power Dissipation** – Almost zero static power dissipation for fully complementary circuits. Power is dissipated during logic transitions
- **Precharging Characteristics** – Both n-type and p-type devices are available for precharging a bus to VDD and VSS. Nodes can be charged fully to VDD or alternatively to VSS in a short time.
- **Power Supply** – Voltage required to switch a gate is a fixed percentage of VDD. Variable range is 1.5 to 15 volts



Chapter 1

1.9 CMOS Attributes

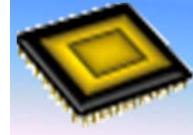
- **Packing Density** – Requires $2n$ devices for n inputs for complementary static gates. Less for dynamic gates or ratio logic form
- **Layout** – CMOS encourages regular and easy automated layout styles



Chapter 1

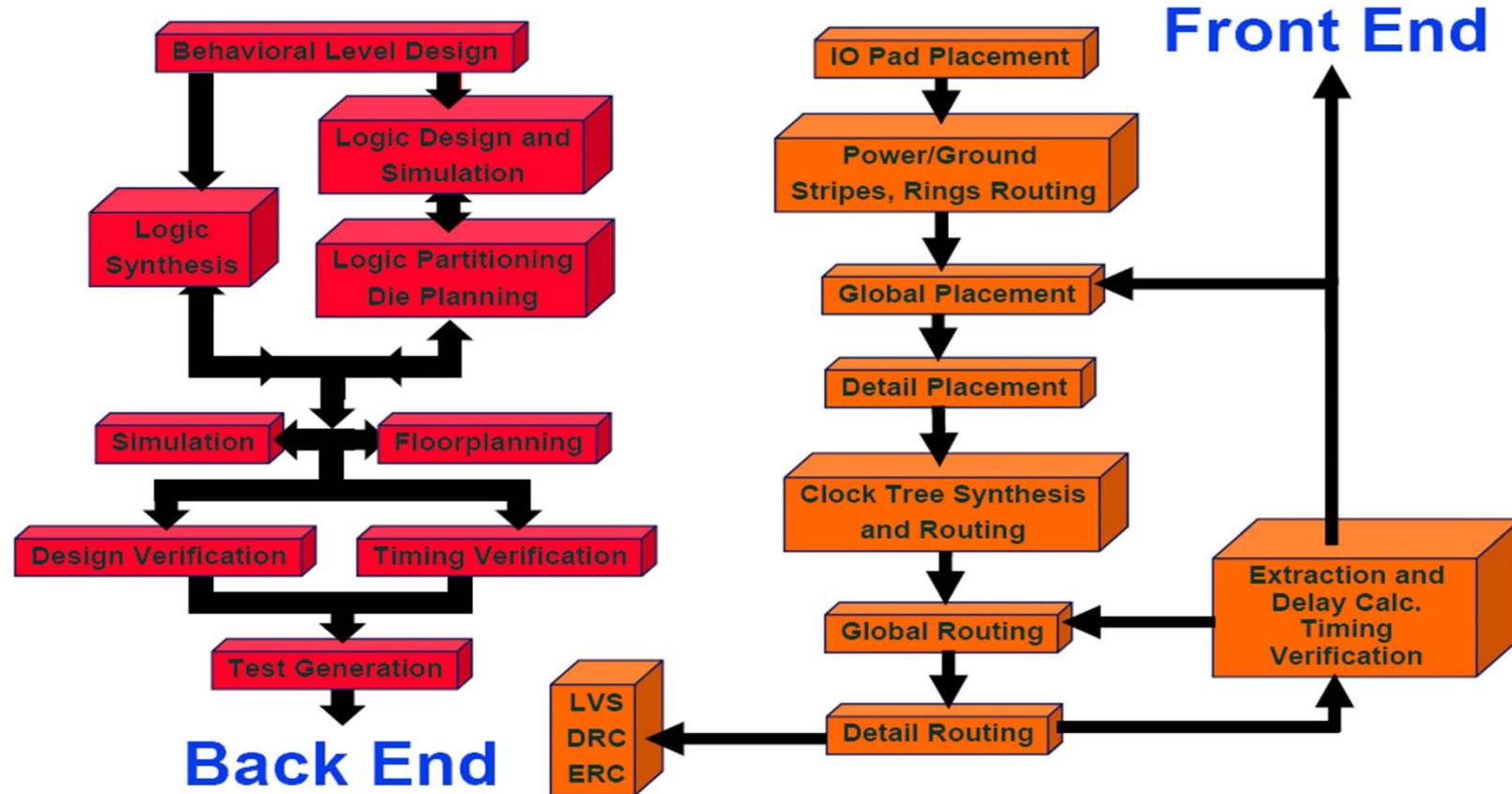
Summary

- As digital design engineers we need to focus on the economics of the design and of the eventual manufacture
 - _____
 - _____
 - _____

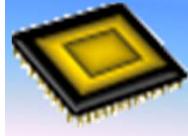


Chapter 1

Traditional Taxonomy

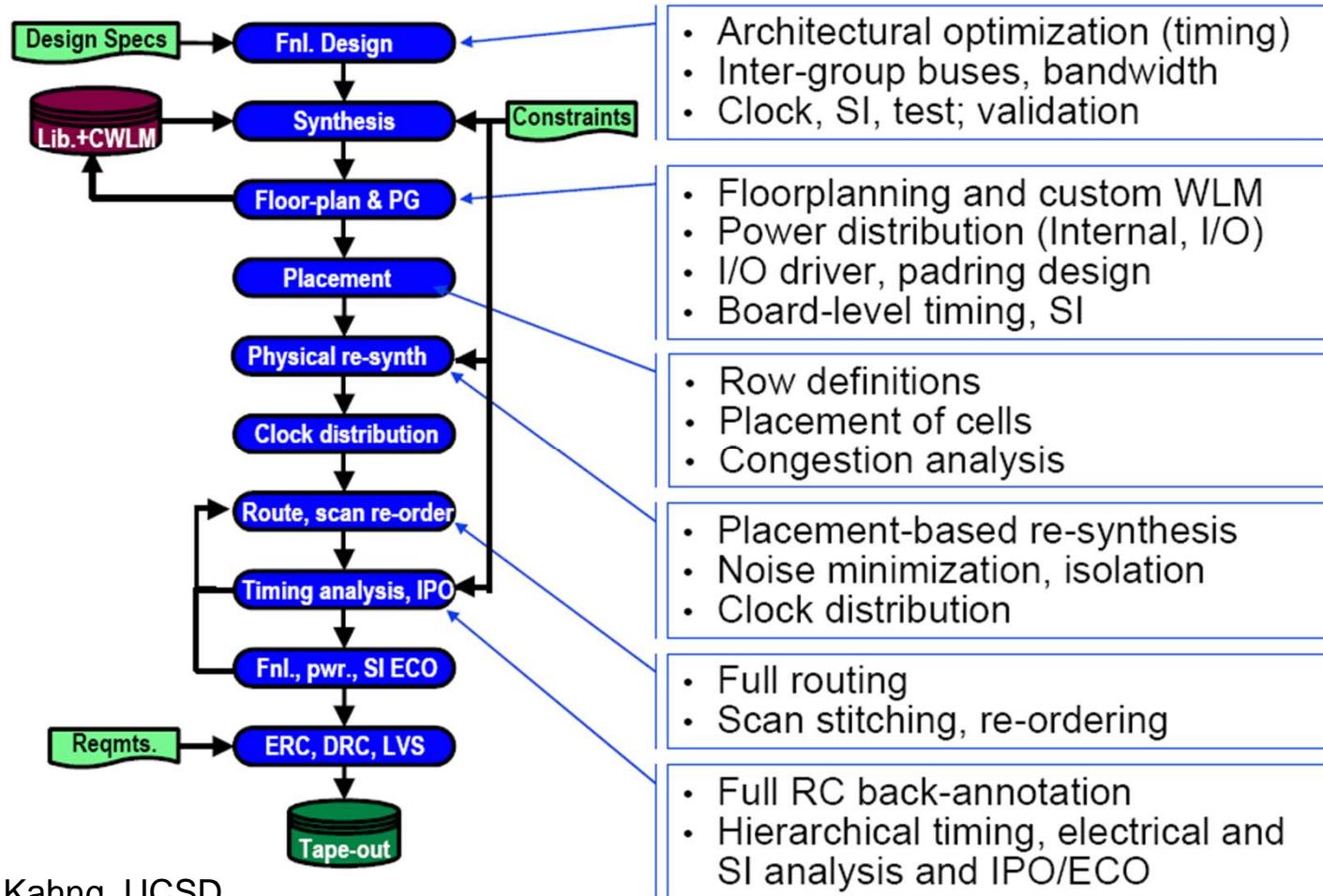


Source: Andrew B. Kahng, UCSD

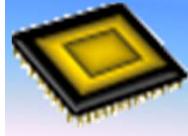


Chapter 1

Another Version (Back-End Flow)

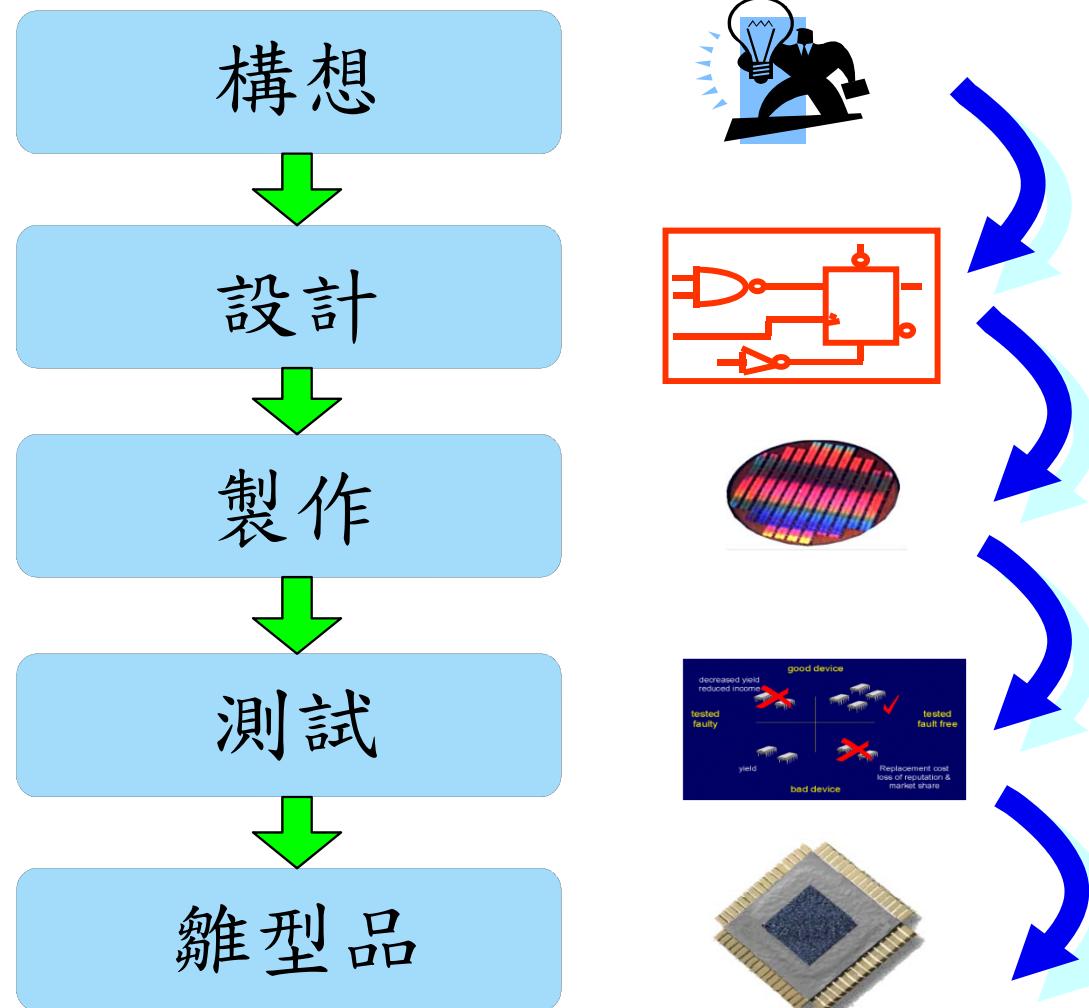


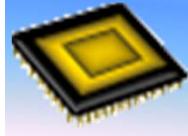
Source: Andrew B. Kahng, UCSD



Chapter 1

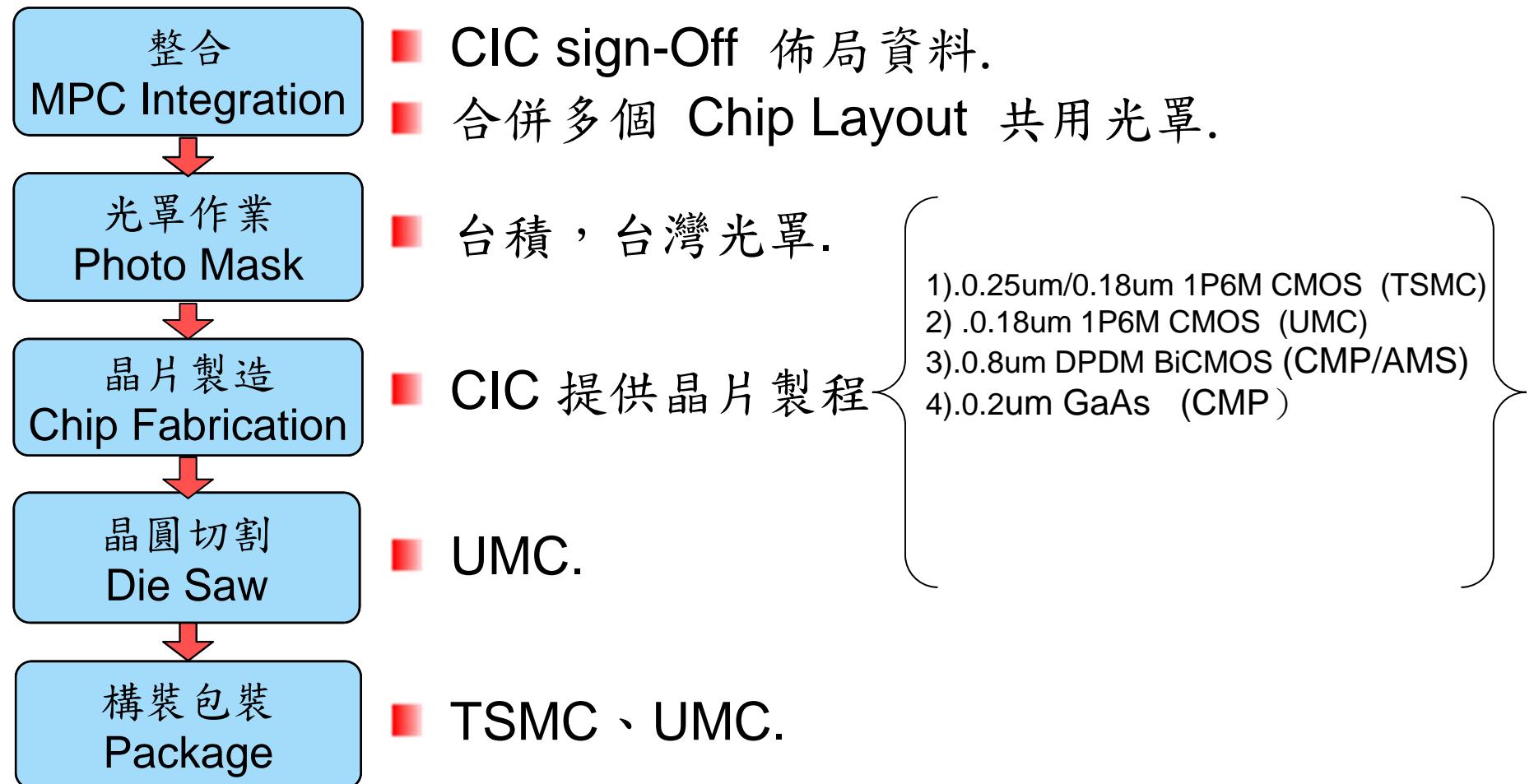
晶片及系統設計製作流程

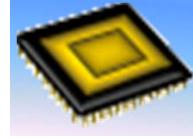




Chapter 1

Multi-Purpose Chip (MPC) 晶片製作流程



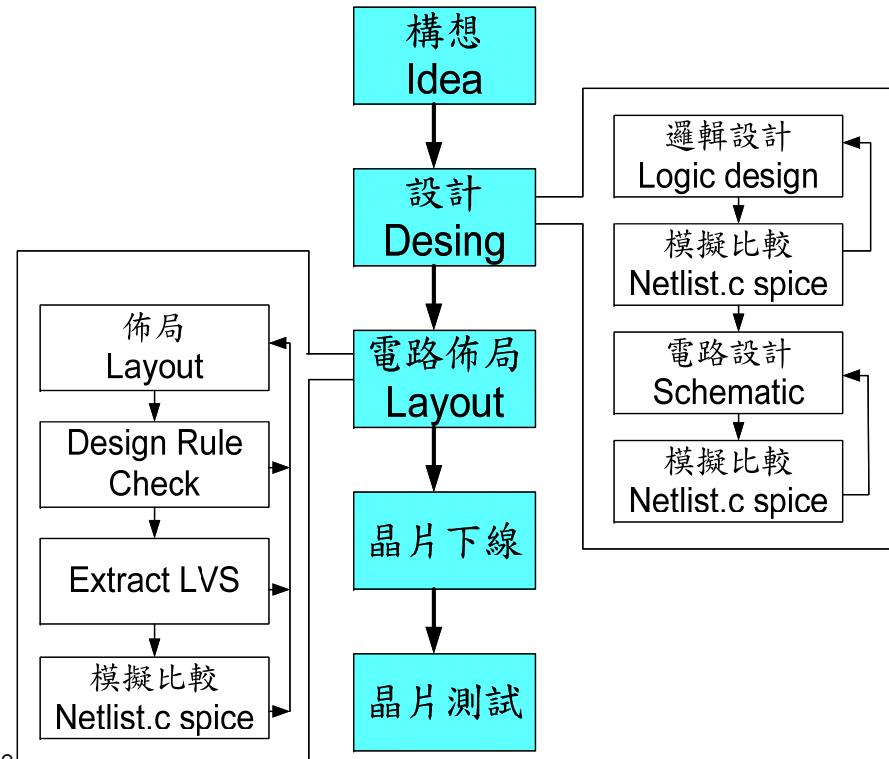


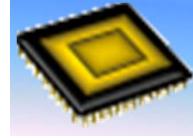
Chapter 1

Full-Custom Design Flow

➤ Full-custom: 客戶端從IC的最基
本元件 (Primitive Cell) 設計開始，
直到整個IC設計完成，客戶端皆保
有最多的自主性，其優點有：

- 較好的專利保護。
- 相同電路會有較低的電源消耗。
- 相同電路 Function 會有較小的面積。
- 較好的設計彈性，及相同電路會有較好的Performance等好處。



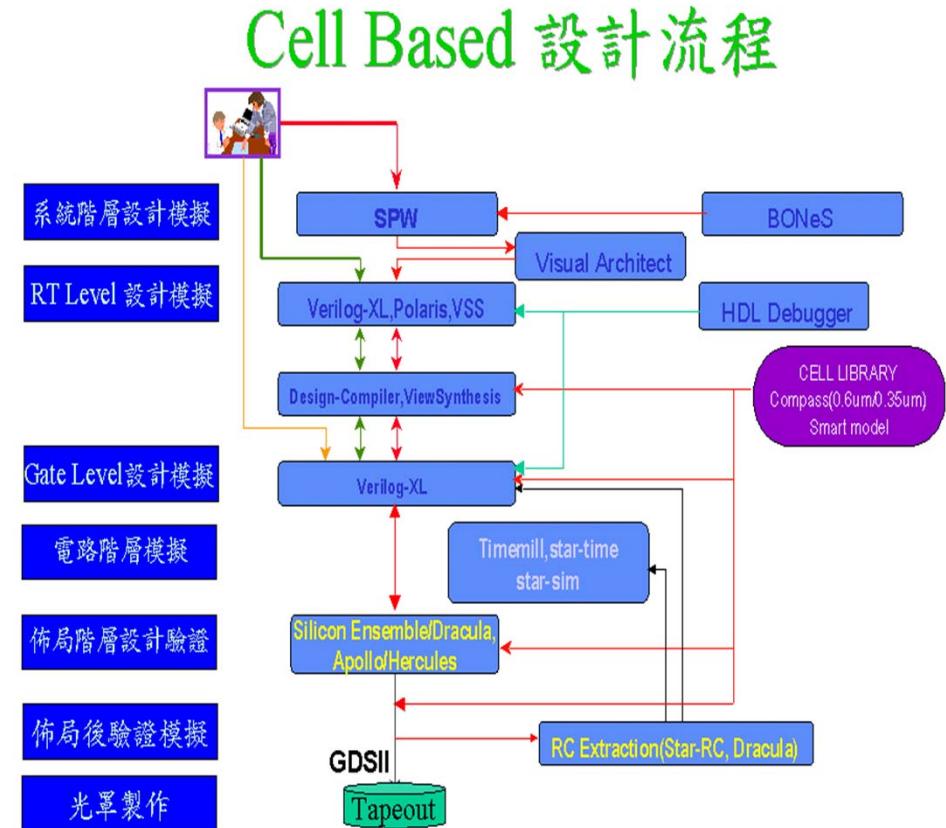


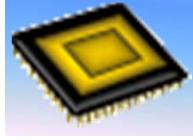
Chapter 1

Semi-Custom Design Flow

➤ Semi-Custom:

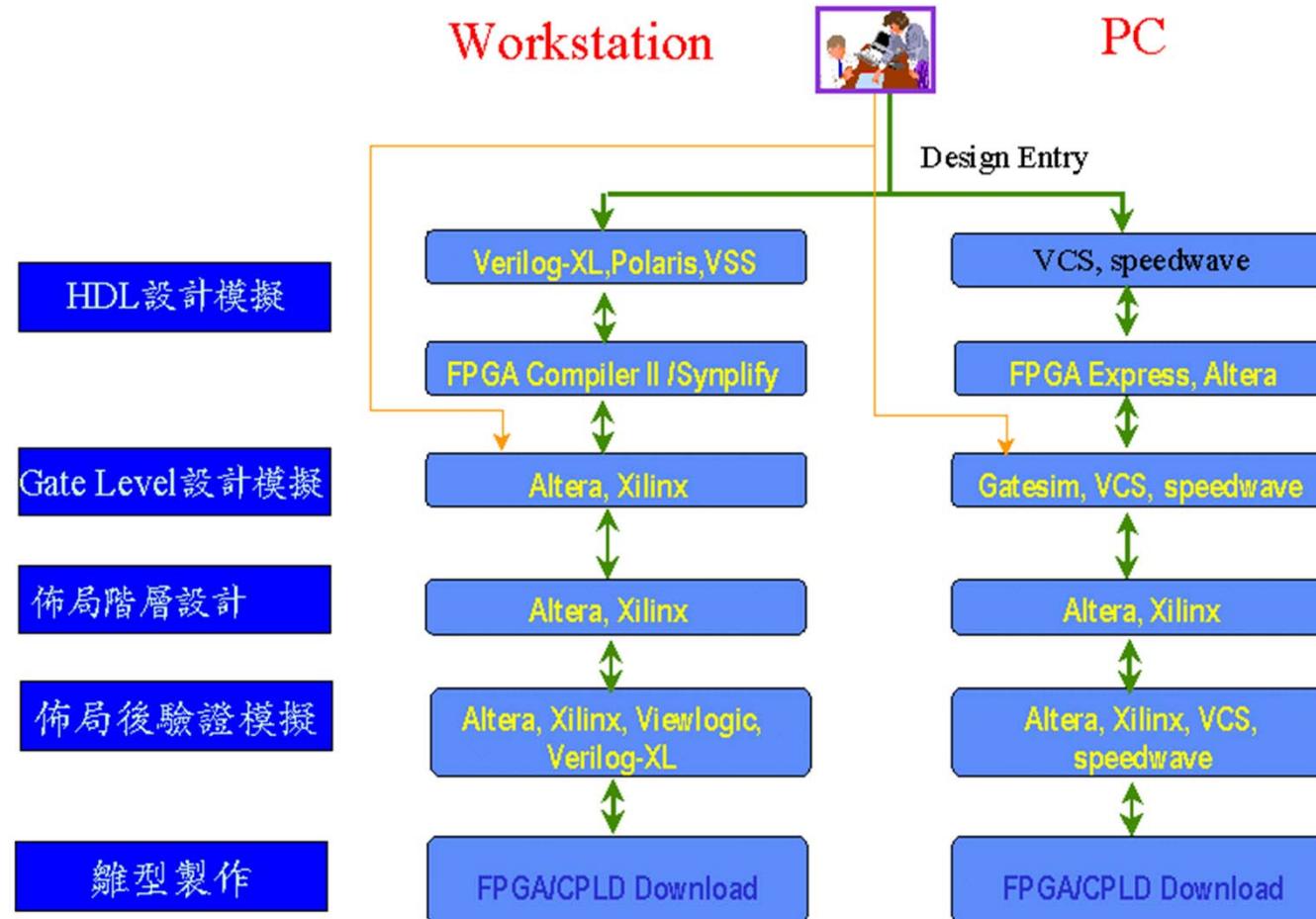
- IC廠除將某一製程參數交由客戶外，也將預先設計好具有某些簡單功能的單元 (Cell)，如 NAND, NOR Gate 等，建立成一個元件庫 (Cell Library)，一併交給客戶。
- Cell包含了元件佈局、元件圖示 (Symbol)、元件Timing Data、負載效應及驅動能力而 Placement及 Routing (簡稱 P&R) 大部份可用自動佈局APR (Automatic Placement & Routing) 完成

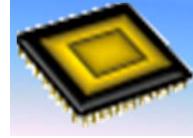




Chapter 1

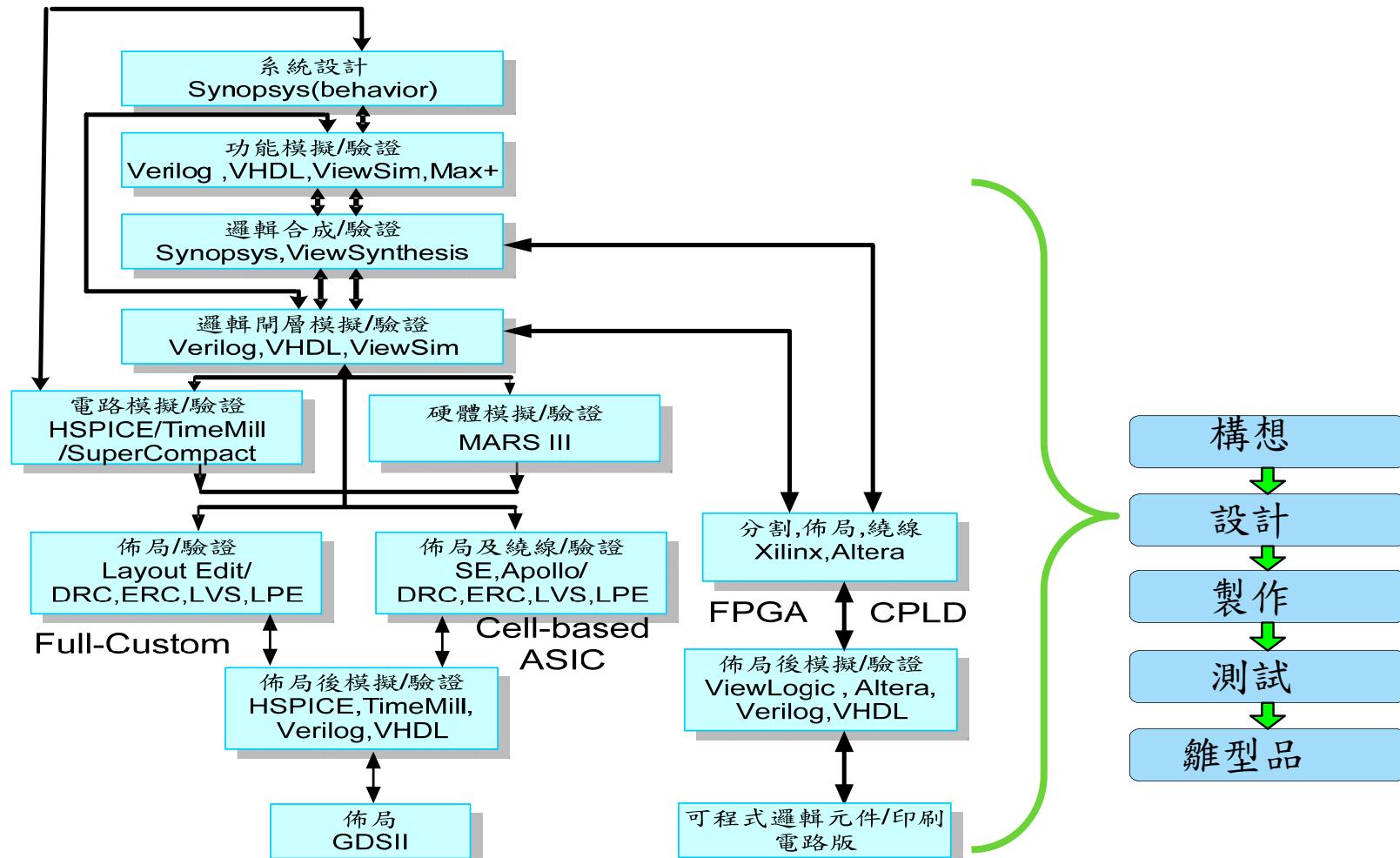
FPGA Design Flow

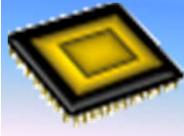




Chapter 1

系統及晶片設計流程

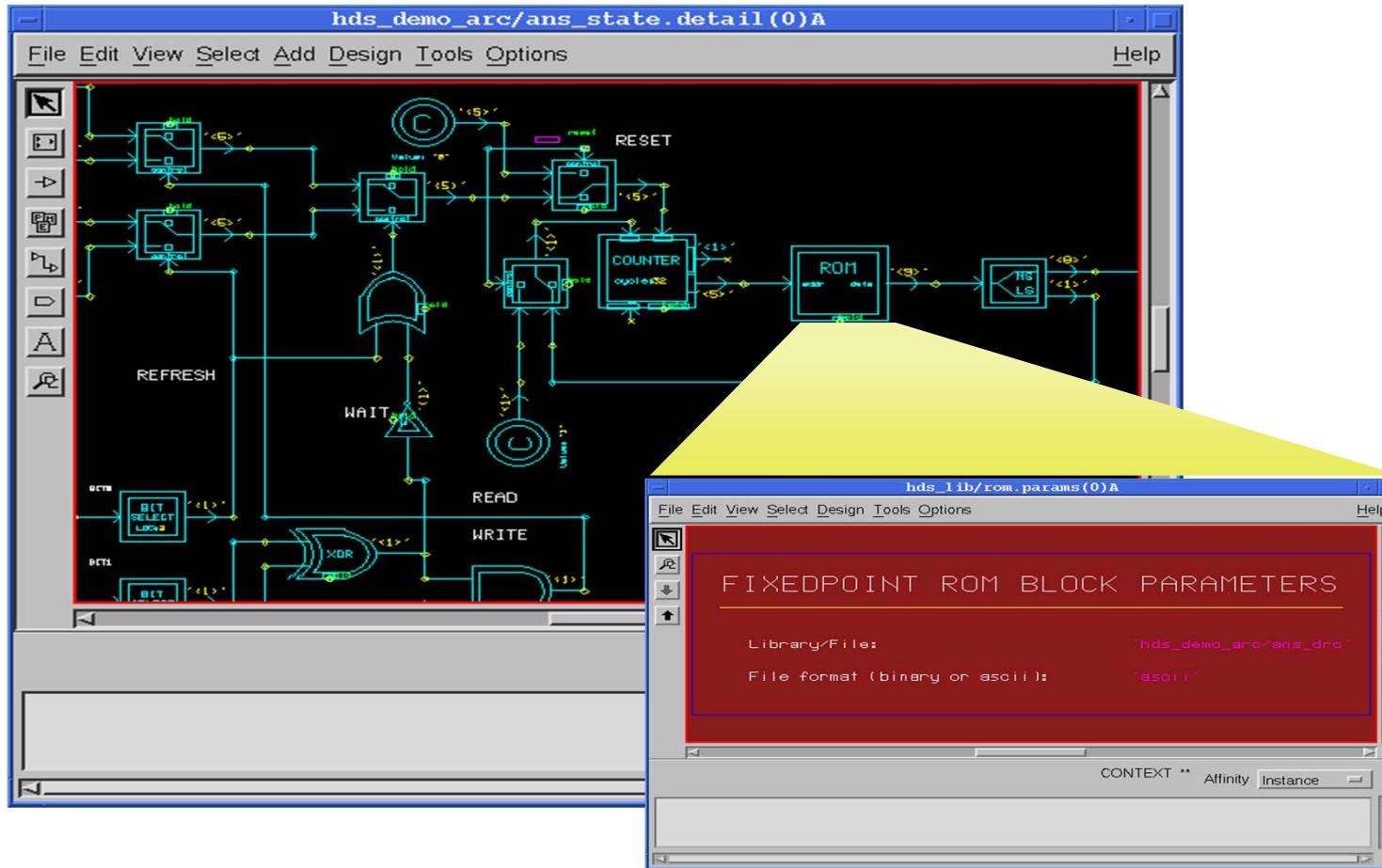


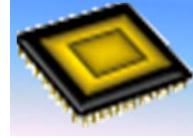


Chapter 1

系統功能設計

- ## ➤ 利用方塊流程圖描述系統功能 (以SPW Tool為例)

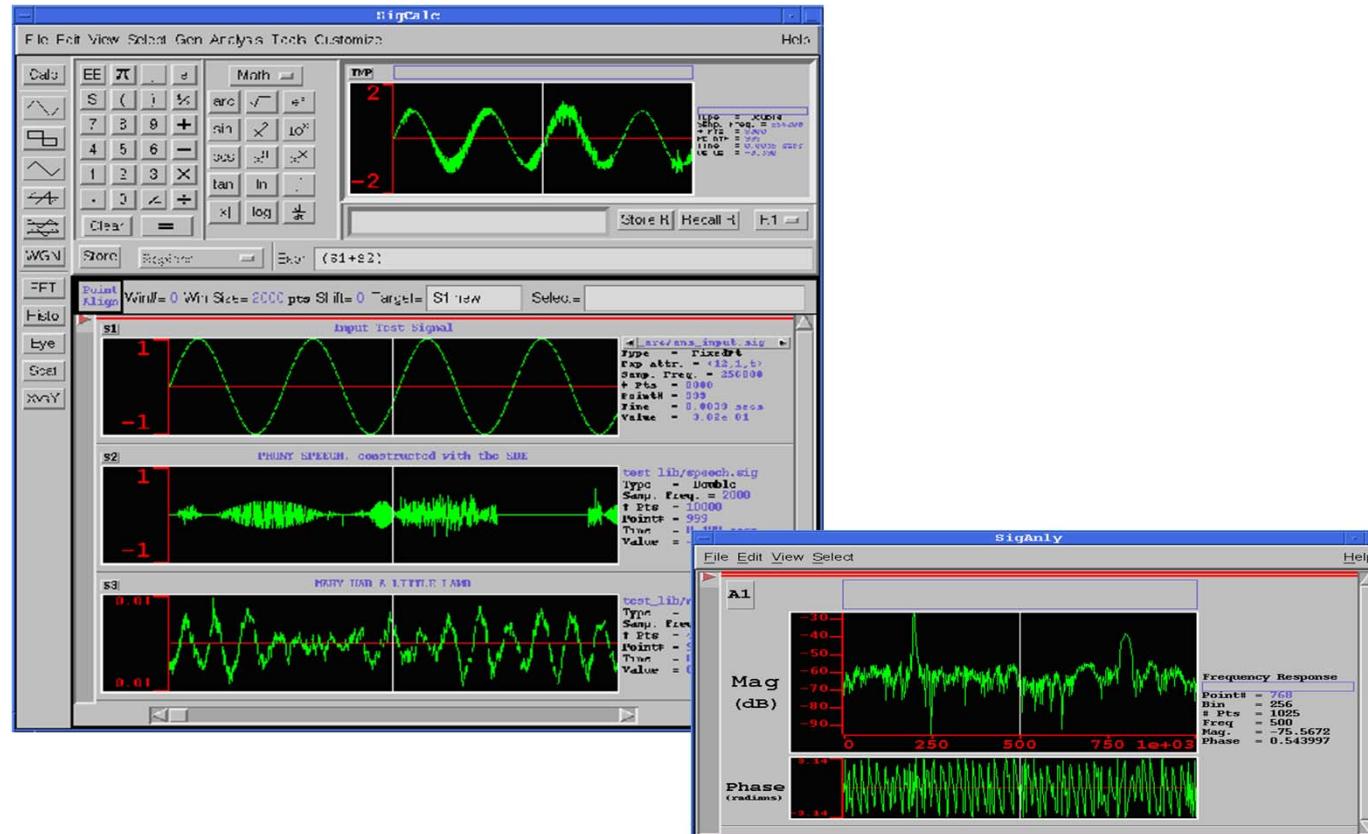


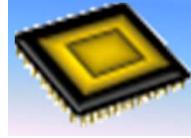


Chapter 1

系統功能模擬與分析

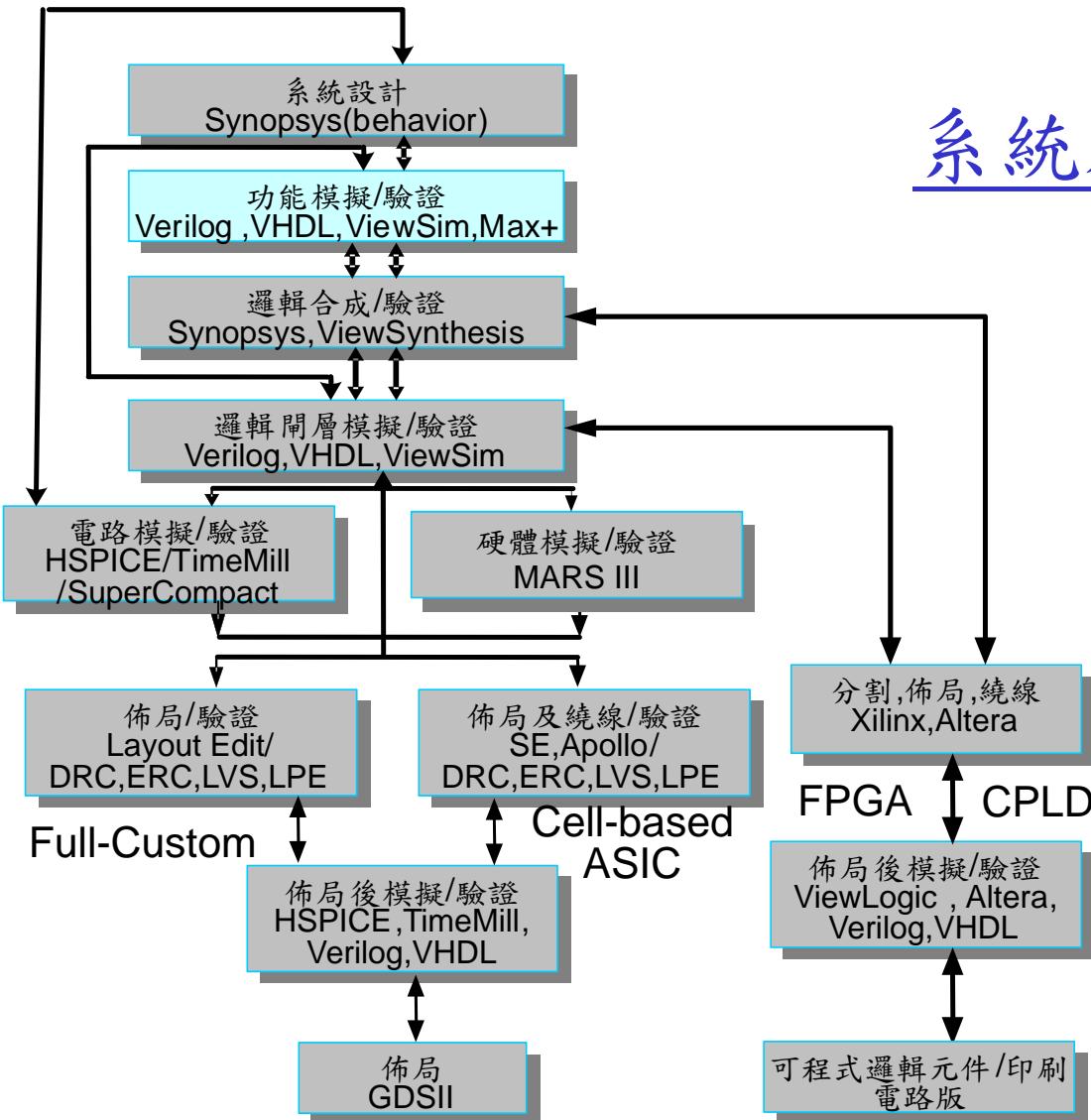
- 利用 Signal Calculator 產生輸入信號做系統功能模擬，並分析系統之輸出信號。

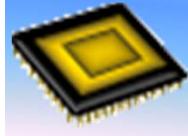




Chapter 1

系統及晶片設計流程

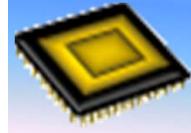




Chapter 1

Behavior (Function) Simulation

- Behavioral (functional) representation
 - usually is the first step in designing a circuit
 - involve a description of how the circuit should communicate with the outside world
 - allow the designer to optimize a design by choosing a circuit from a set of structurally different, yet functionally identical ones that conform to the desired behavioral representation
- Hardware description language (HDL)
 - is a programming language with a syntax specially designed to allow the description of circuits currently, the most popular HDLs
 - VHDL: Very high speed integrated circuit Hardware Description Language
 - Verilog



Chapter 1

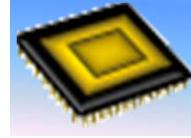
CAD tools vendor

➤ Structural design

- **CAD tools** typically allow a circuit to be represented at multiple abstraction levels



Viewlogic

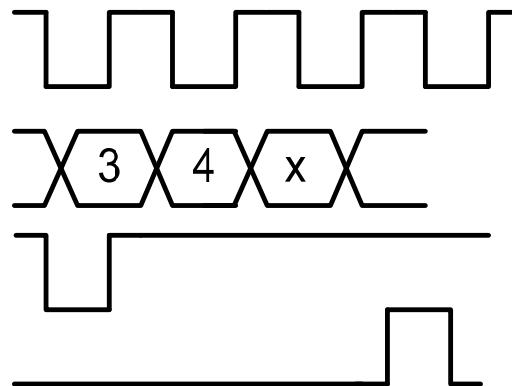


Chapter 1

Behavior (Function) Simulation

- 使用 Hardware Description Language (VHDL or Verilog)。
- 為了驗證所描述的 Behavior (function) 之正確性，必須做 Behavior Simulation.
- 當模擬結果產生後，我們拿它與所希望的結果相比對，如果結果不滿意就修改 Behavior Description，直到滿意為止。

輸入 Testfixture

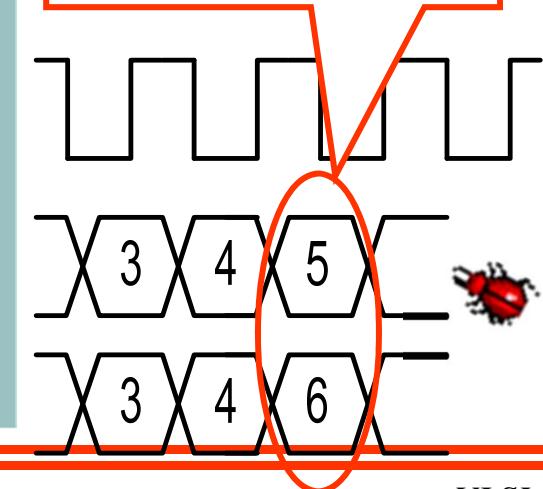


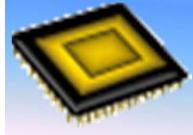
Behavior 之描述

```
library ieee;
use ieee.std_logic_1164.all;
entity onebit_full_adder_4 is
port(cin    : in std_logic ;
     a       : in std_logic ;
     b       : in std_logic ;
     sum    : out std_logic ;
     carry  : out std_logic );
end onebit_full_adder_4;
```

比對輸出波形與
預期波形

Function 不正確



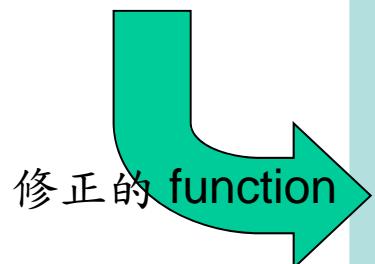


Chapter 1

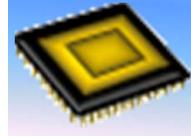
修改 Description

- 找出Behavior Description 中錯誤的地方並修改它.

```
library ieee;
use ieee.std_logic_1164.all;
entity onebit_full_adder_4 is
port(cin      : in std_logic ;
     a          : in std_logic ;
     b          : in std_logic ;
     sum        : out std_logic ;
     carry      : out std_logic );
end onebit_full_adder_4;
```



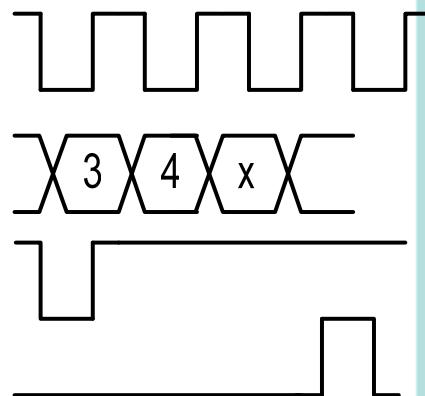
```
library ieee;
use ieee.std_logic_1164.all;
entity onebit_full_adder_4 is
port(cin : in std_logic ;
     a      : in std_logic ;
     b      : in std_logic ;
     sum   : out std_logic ;
     carry : out std_logic );
end onebit_full_adder_4;
architecture synth_RTL of
onebit_full_adder_4 is
signal wire1 :std_logic;
begin
    wire1 <= ((a or b) nand (a nand b));
    carry <= ((a nand b) nand ((a or b) nand
cin));
    sum   <= ((wire1 nand cin) nand
(wire1 or cin));
end synth_RTL;
```



Chapter 1

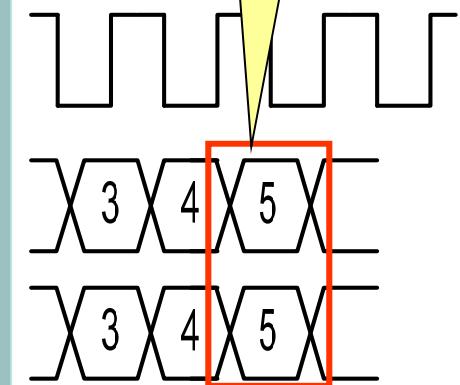
Re-Simulation

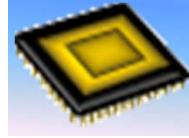
➤修改完之Behavior Description 要再做 Simulation ，檢查其輸出波形是否符合要求。



```
library ieee;
use ieee.std_logic_1164.all;
entity onebit_full_adder_4 is
port(cin : in std_logic ;
      a          : in std_logic ;
      b          : in std_logic ;
      sum        : out std_logic ;
      carry      : out std_logic );
end onebit_full_adder_4;
architecture synth_RTL of
onebit_full_adder_4 is
signal wire1 :std_logic;
begin
    wire1 <= ((a or b) nand (a nand b));
    carry <= ((a nand b) nand ((a or b)
nand cin));
    sum    <= ((wire1 nand cin) nand
(wire1 or cin));
end synth_RTL;
```

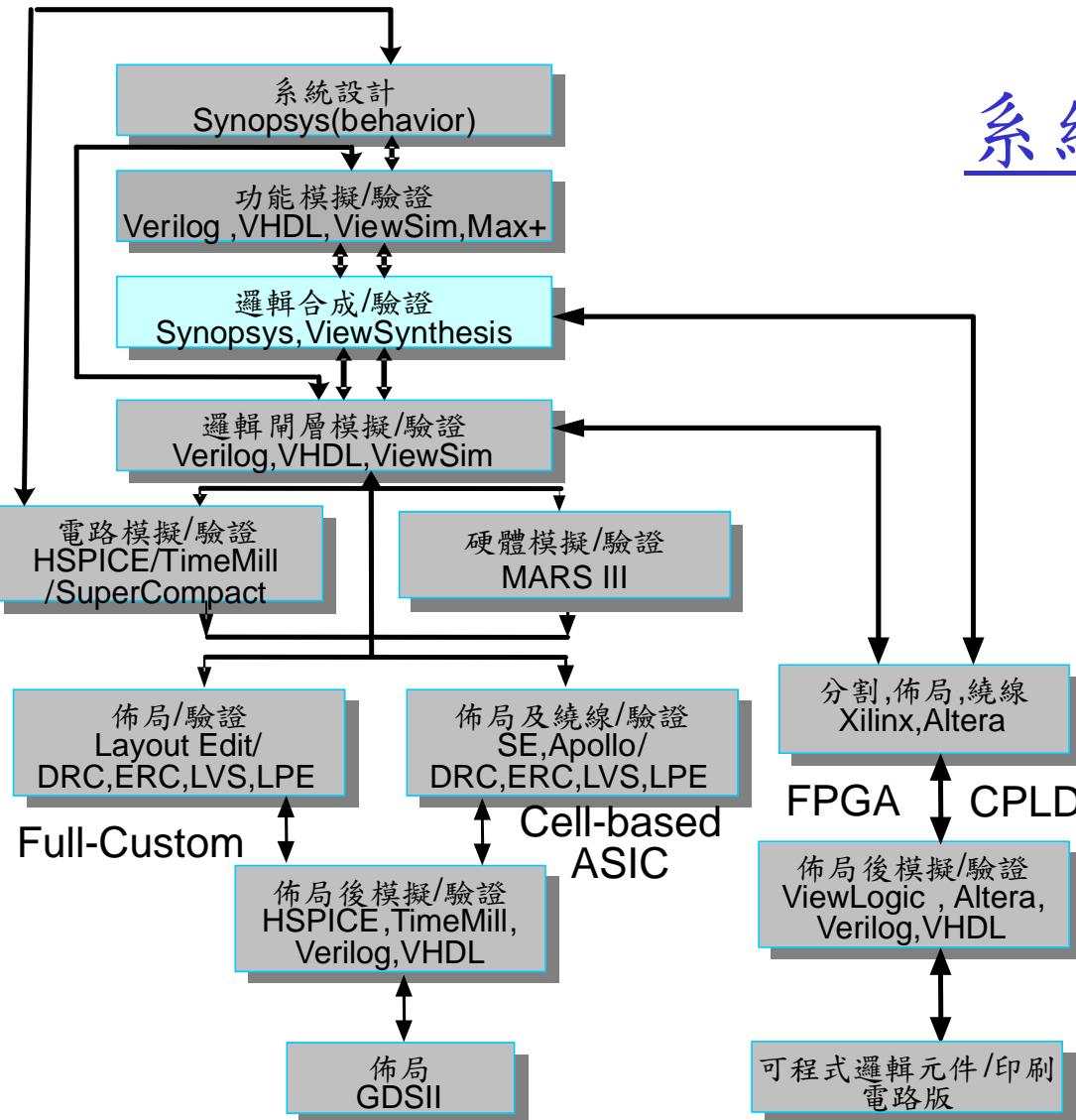
Function
正確

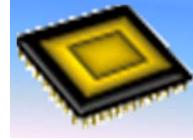




Chapter 1

系統及晶片設計流程

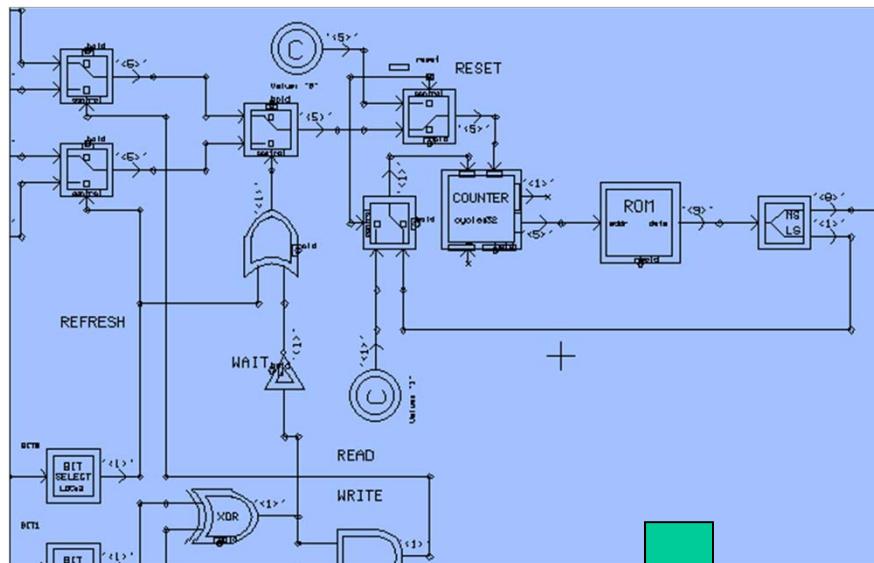




Chapter 1

產生可合成之VHDL Code

➤ VHDL Link 為驗證過之系統設計，可做邏輯合成之



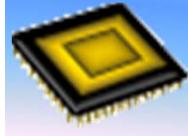
對應的Code

```
library IEEE;
library comdisco_mv19;
use IEEE.std_logic_1164.all;
use comdisco_mv19.hds_arith_synth.all;
use comdisco_mv19.hds_fxp_synth.all;
use comdisco_mv19.hds_proc_synth.all;

entity hds_demo_arc_ans_state is
    port(SM_IN_PORT : in UN(2 downto 0);
          HOLD_PORT : in UN(0 downto 0);
          RESET_PORT : in UN(0 downto 0);
          SM_OUT_PORT : out UN(7 downto 0);
          system_clock: in FXP_BIT);
end hds_demo_arc_ans_state;

architecture hds_body of hds_demo_arc_ans_state
is

procedure hds_lib_rom_536 (hold : UN(0 downto 0); addr : UN(4 downto 0); hold_state : UN(8 downto 0); O : out UN(8 downto 0)) is
```



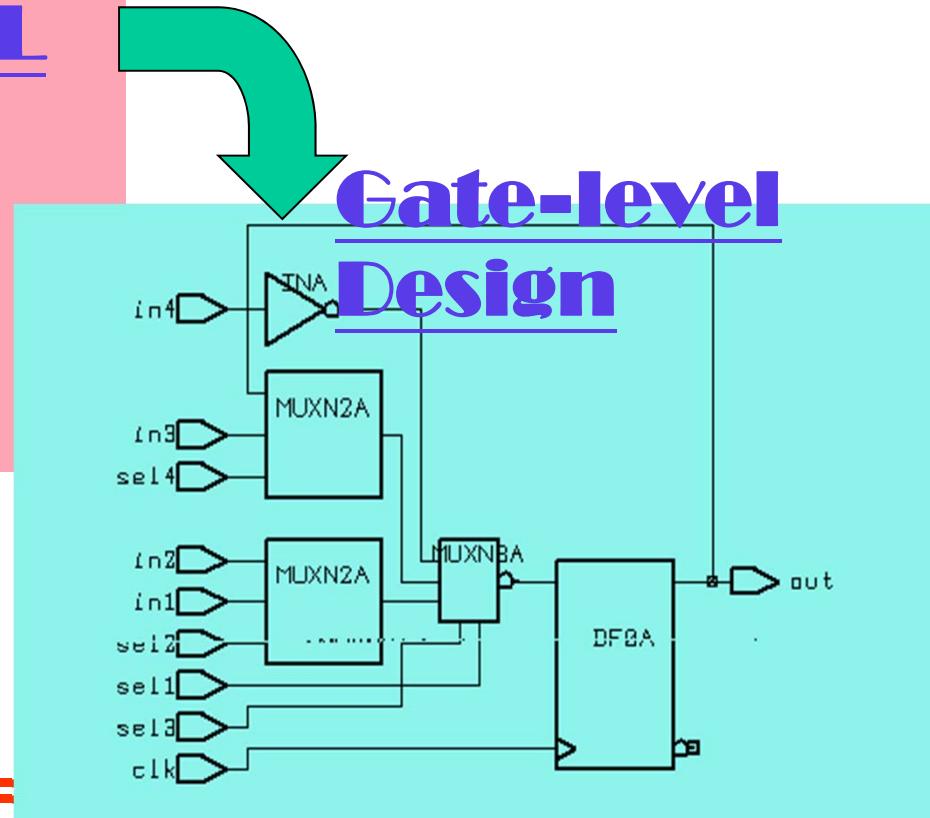
Chapter 1

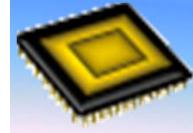
邏輯合成 Logic Synthesis

```
always @(posedge clk) begin
    if (sel1) begin
        if (sel2)
            out=in1;
        else
            out=in2;
    end
    else if (sel3) begin
        if (sel4)
            out=in3;
    end
    else
        out=in4;
end
```

RTL

➤ Logic Synthesis translates RTL design to gate-level design.

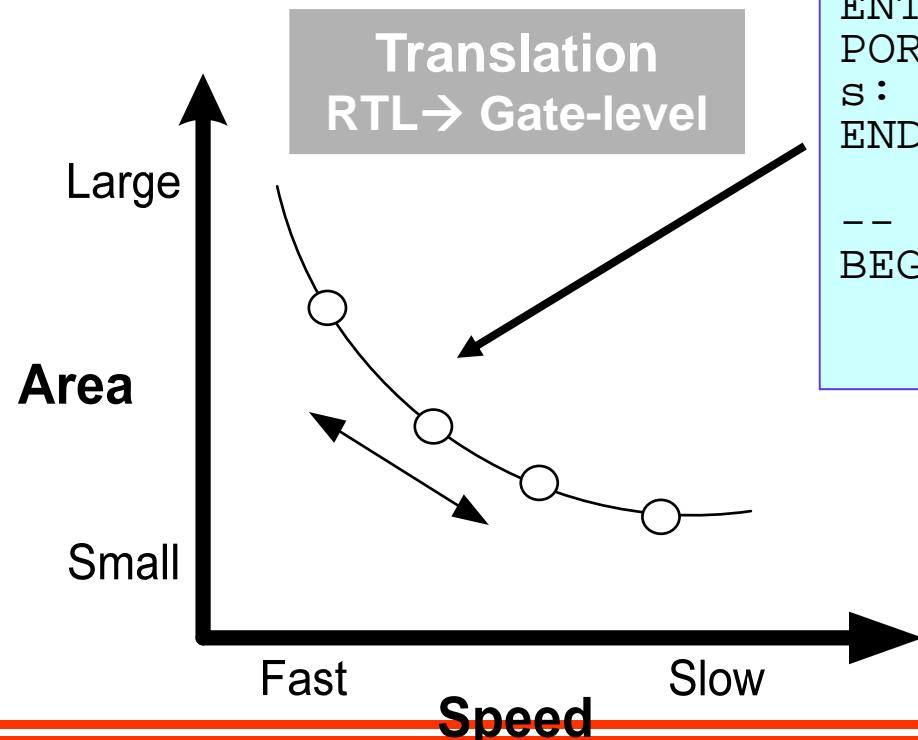




Chapter 1

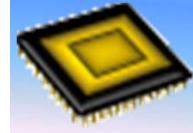
Synthesis is Constraint Driven

- Logic Synthesizer first translates RTL design to an **intermediate gate-level design** , then optimize according to the **area** and **timing** constraint.



```
-- Interface
ENTITY adder IS
PORT (a, b:IN unsigned(0 to 31);
      s: OUT unsigned (0 to 32));
END adder;

-- Behavioral representation
BEGIN
      s <= a + b;
END behavioral;
```

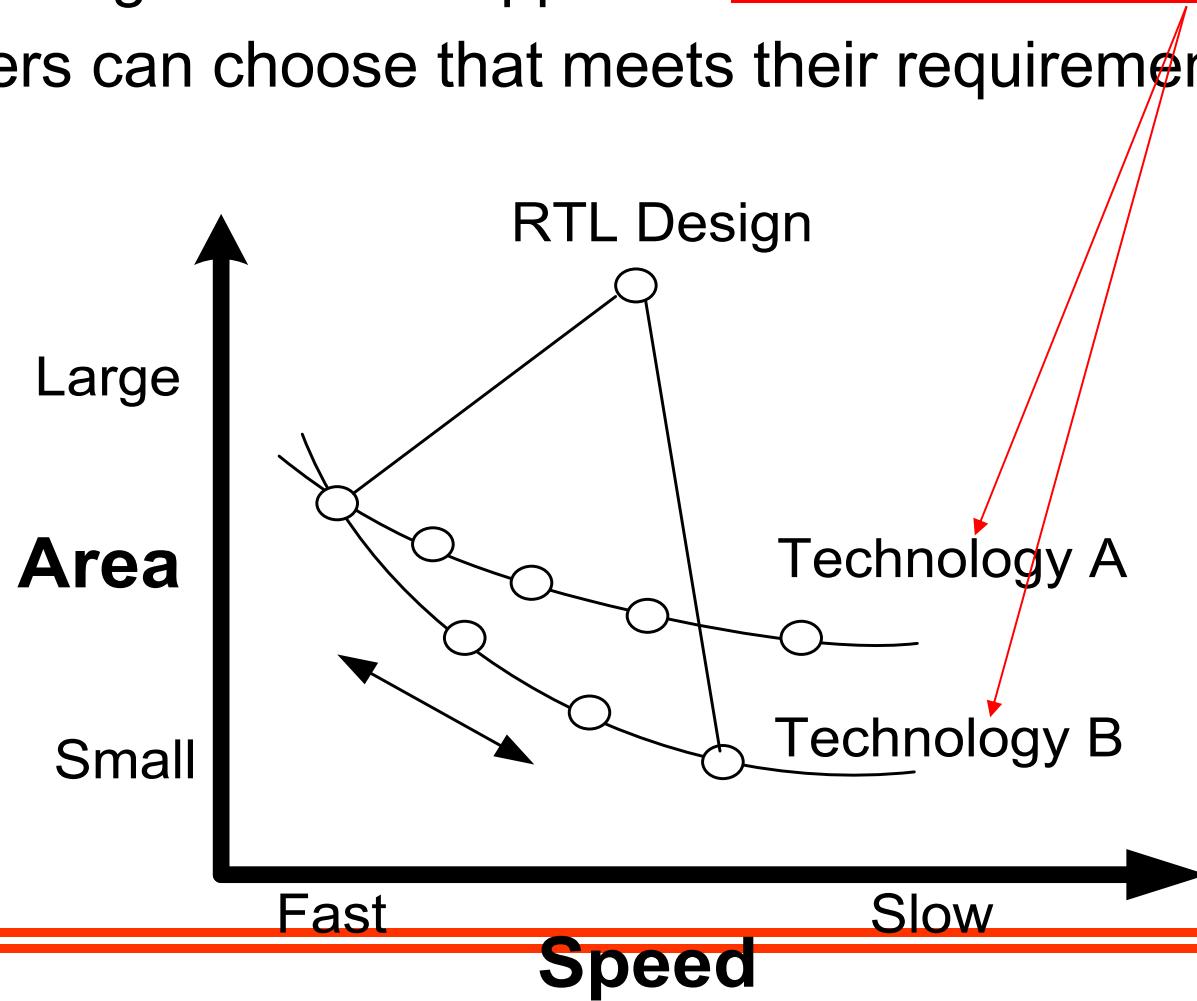


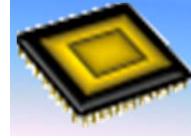
Chapter 1

Synthesis Handles Technology

Independent Design.

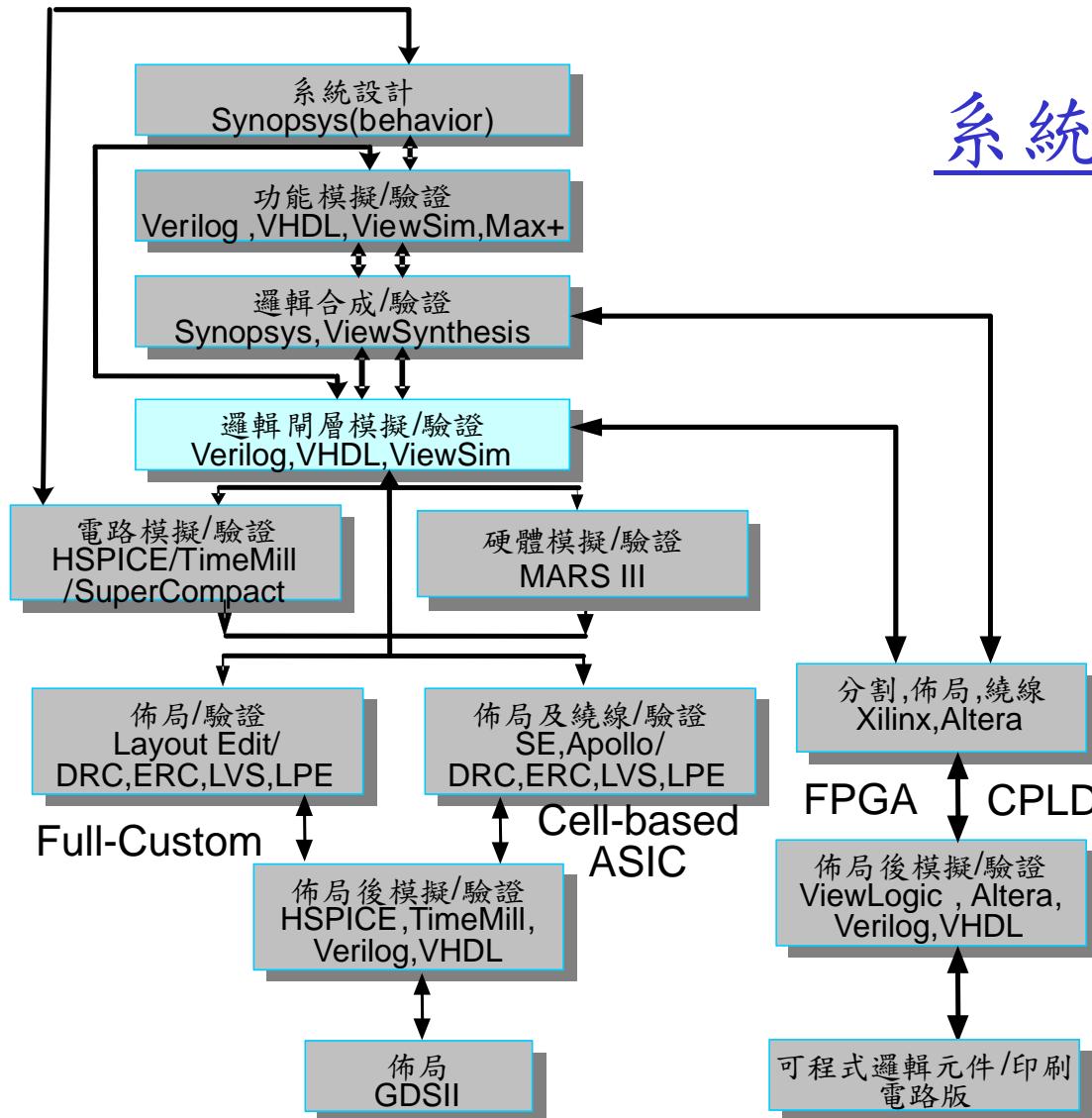
- A RTL design can be mapped to _____, designers can choose that meets their requirement.

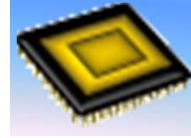




Chapter 1

系統及晶片設計流程



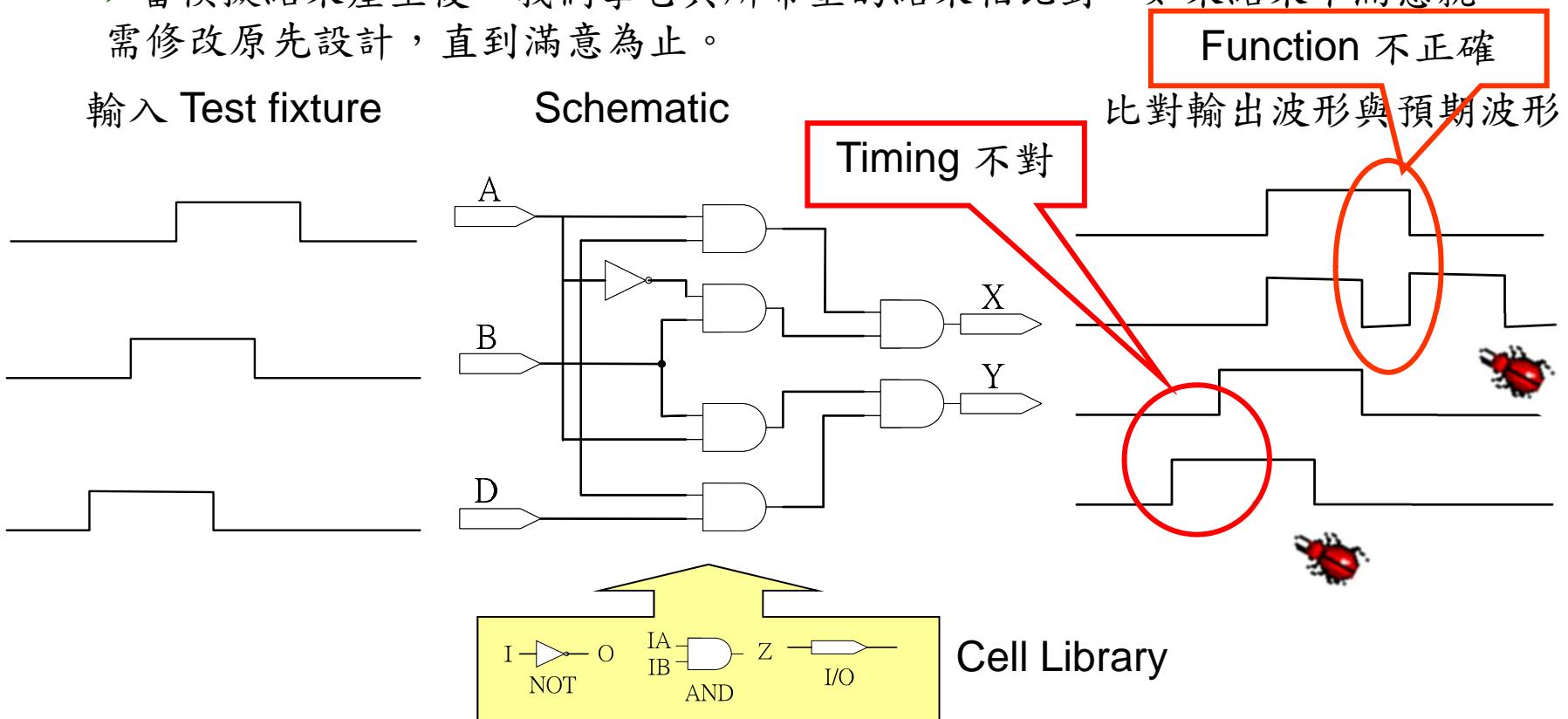


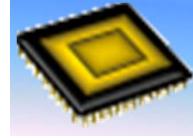
Chapter 1

Gate Level Simulation

➤ 在產生 Schematic 之後，必須做 Gate Level Simulation。其目的在驗證電路的 function 之正確性及 timing 是否符合設計時的要求。

➤ 當模擬結果產生後，我們拿它與所希望的結果相比對，如果結果不滿意就需修改原先設計，直到滿意為止。

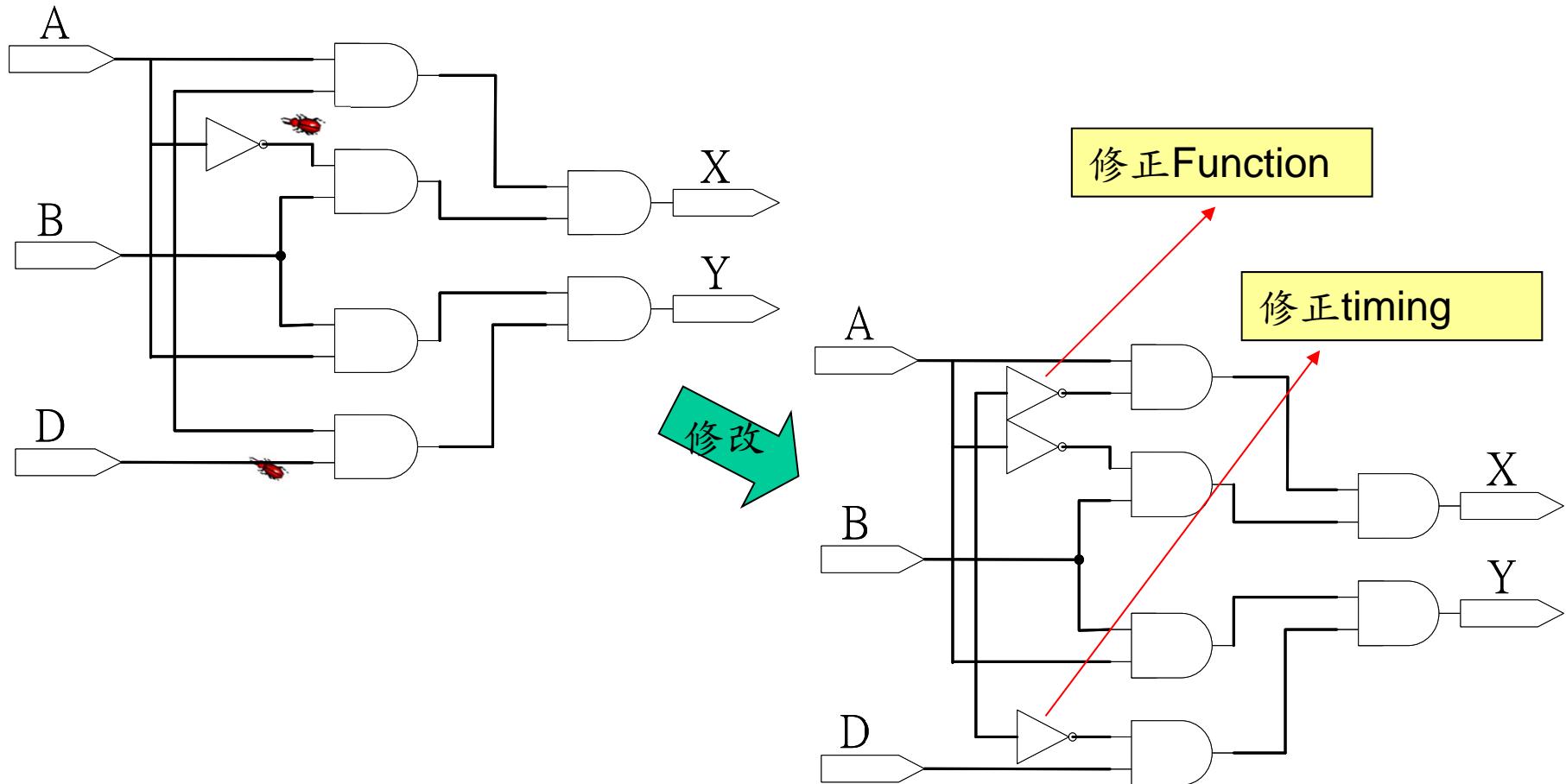


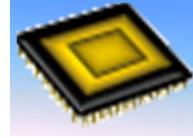


Chapter 1

修改Schematic

➤找出Schematic中造成輸出波形錯誤的地方並修改它。



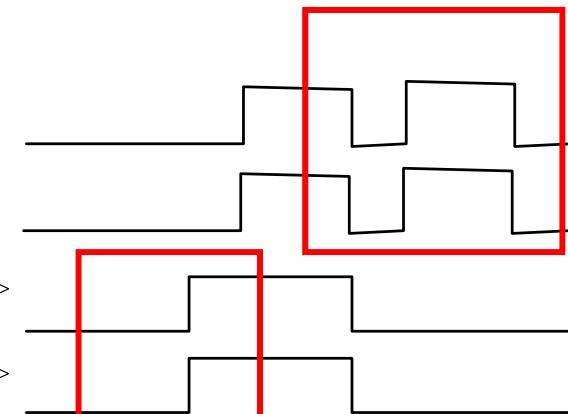
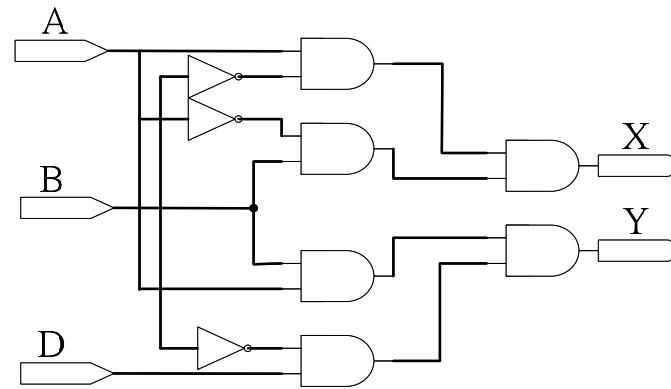
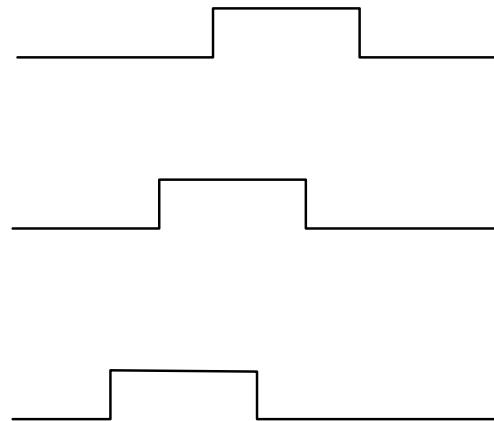


Chapter 1

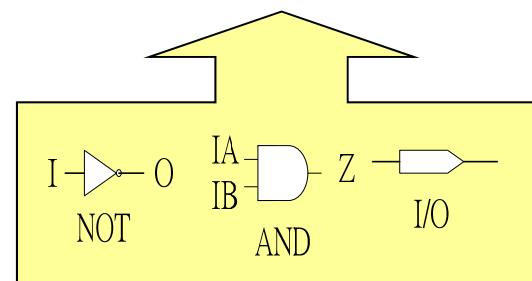
Re-Simulation

➤修改完之 Schematic 要再做 Simulation，檢查其輸出波形是否符合要求。

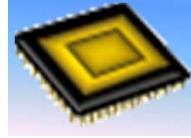
Function 正確



timing 正確

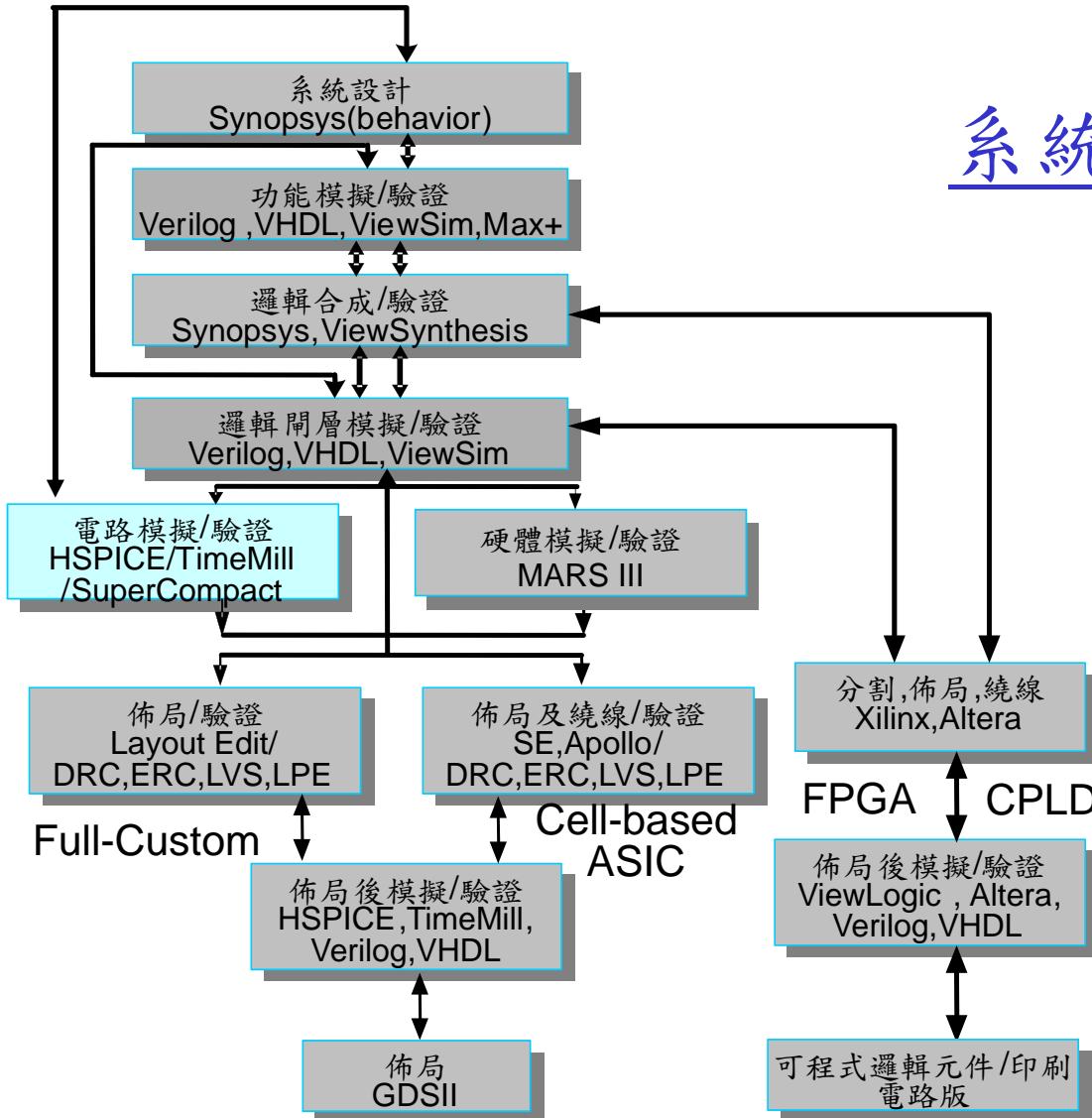


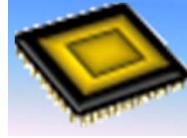
Cell Library



Chapter 1

系統及晶片設計流程

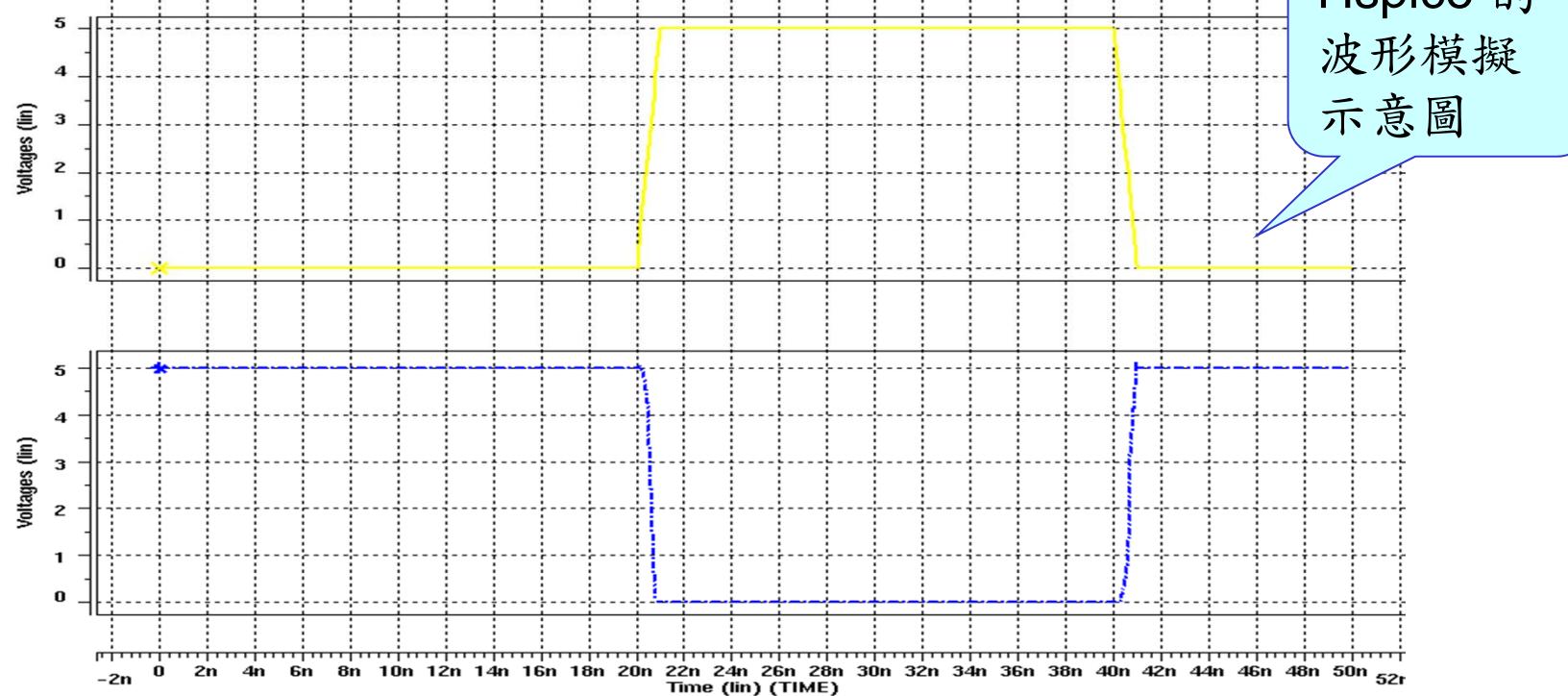


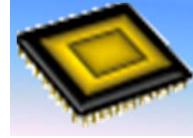


Chapter 1

SPICE 電路模擬軟體

- 模擬欲輸入的波形，以HSPICE作電路的時序分析，並觀察相對的輸出波形。
- 若電路輸出無法達到規格要求，則需重新設計電路後再模擬，直到輸出波形符合預定規格為止。





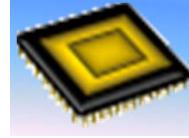
Chapter 1

SPICE電路模擬軟體(cont.)

- 以 netlist 表達電路的連接，再配合電晶體模型，用 HSPICE 軟體可做精確的時序模擬 (Timing Simulation)。

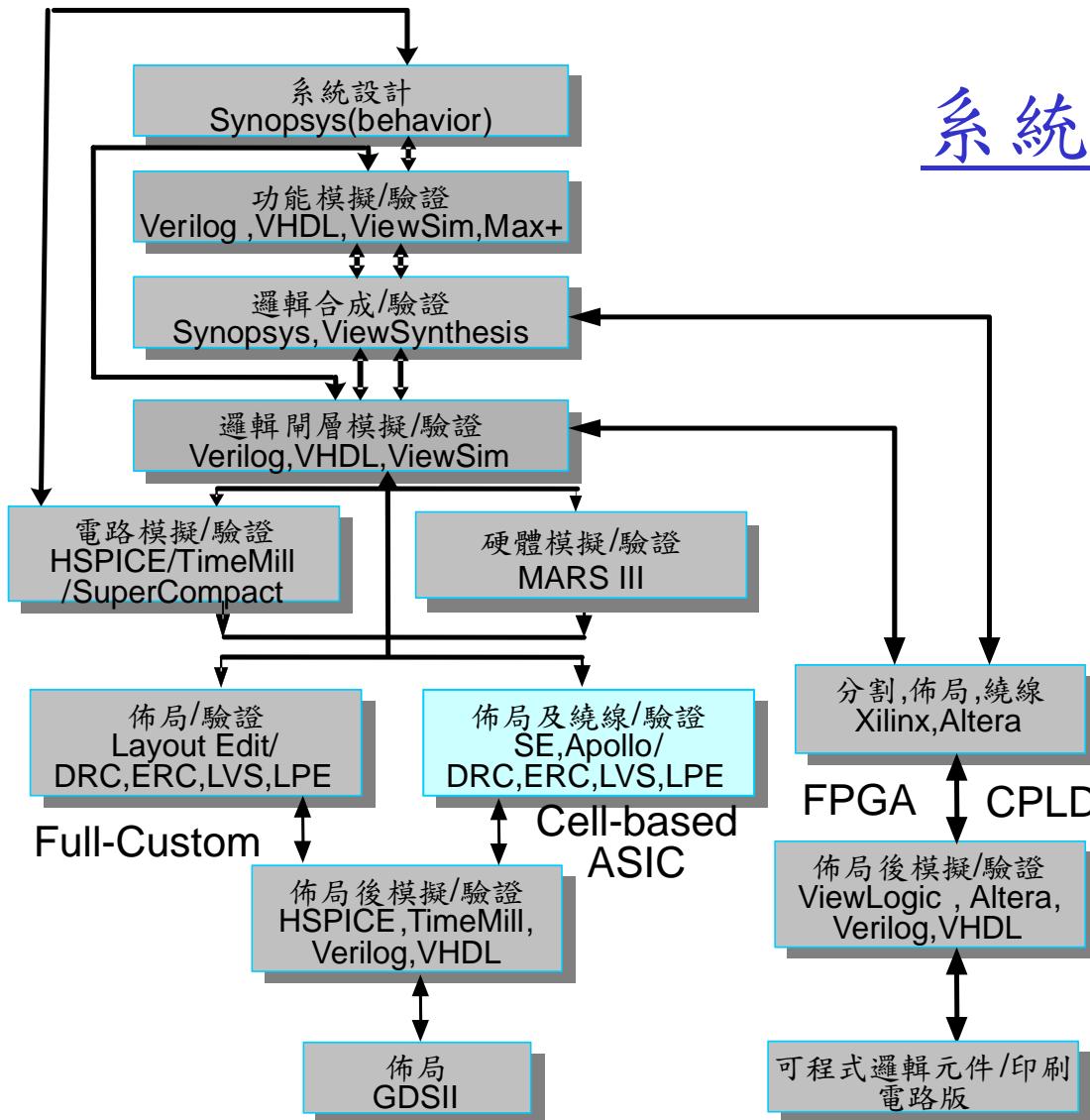
電路圖以 netlist 表示

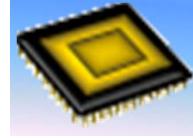
```
*This circuit is a NOR gate.  
MM5 N19 N27 vdd vdd PM W=76.8e-6 L=0.6u  
MM4 Z N19 vdd vdd PM W=315e-6 L=0.6u  
MM2 N31 I vdd vdd PM W=4.8e-6 L=0.6u  
MM0 N27 N31 vdd vdd PM W=19.2e-6 L=0.6u  
MM7 N19 N27 0 0 NM W=38.4e-6 L=0.6u  
MM6 Z N19 0 0 NM W=146e-6 L=0.6u  
MM3 N31 I 0 0 NM W=2.4e-6 L=0.6u  
MM1 N27 N31 0 0 NM W=9.6e-6 L=0.6u  
VIN1 I 0 PWL(0 0V 20ns 0V 21ns 5V 40ns 5V  
41ns 0V)  
.OP  
.TRAN 2NS 50NS  
.OPTIONS POST  
.END
```



Chapter 1

系統及晶片設計流程

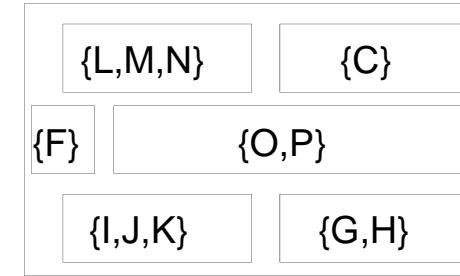
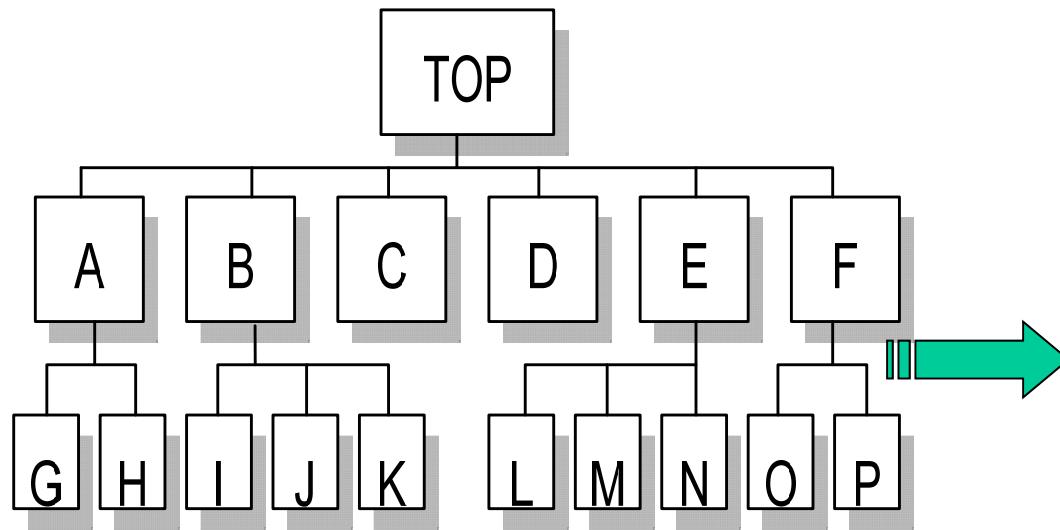




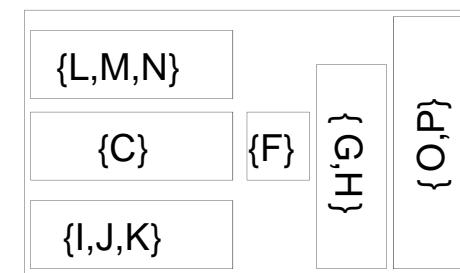
Chapter 1

Floorplan

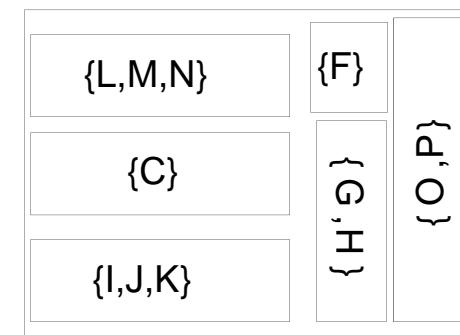
Design Hierarchy



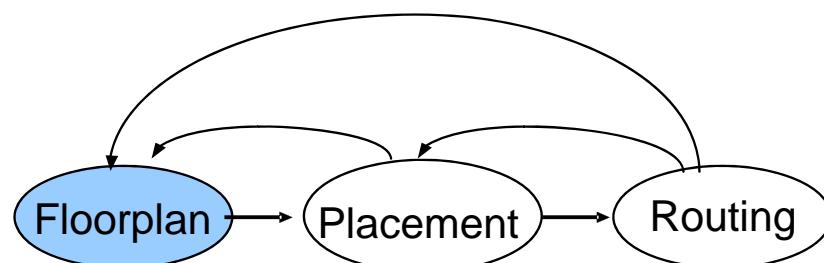
方式一

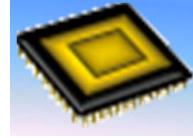


方式二



方式三



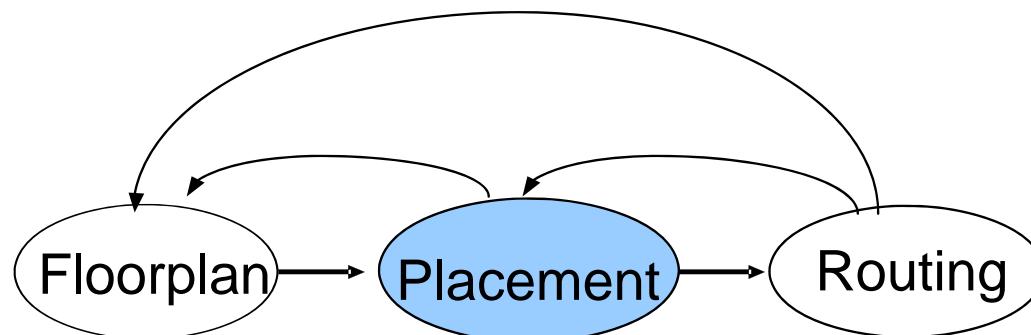
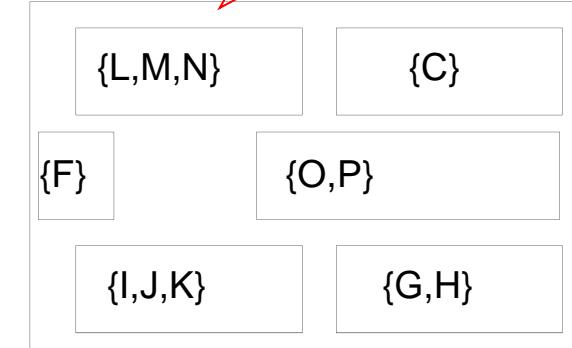
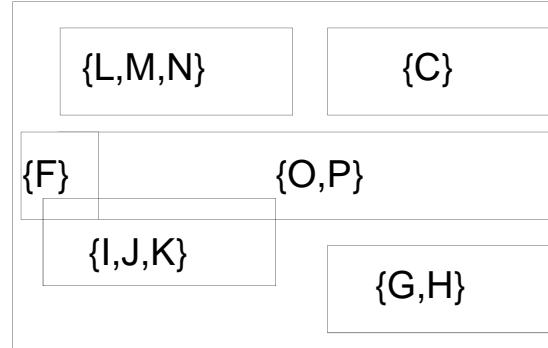


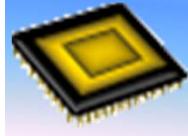
Chapter 1

Placement

- 經過Placement之後，可能由於Black Size改變，使結果變差。
此時就需重做Floorplan，然後再一次Placement。

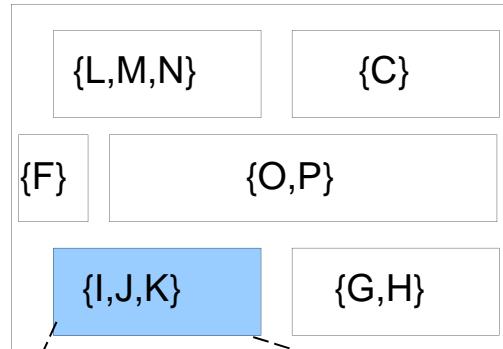
評估結果是否需要
Re-Floorplan



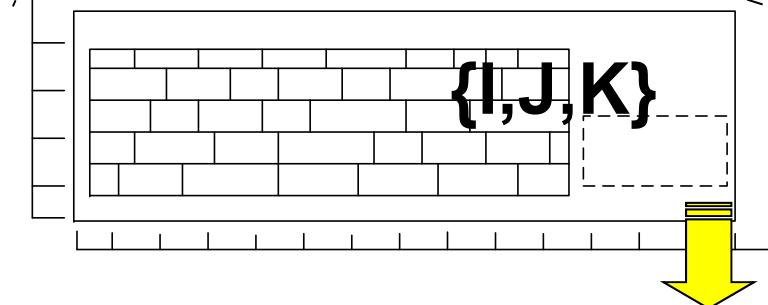


Chapter 1

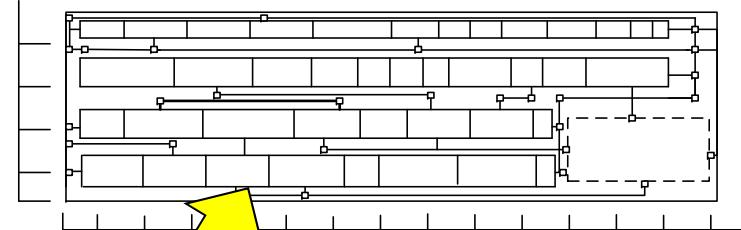
Sub-block Floorplan, Place, Route



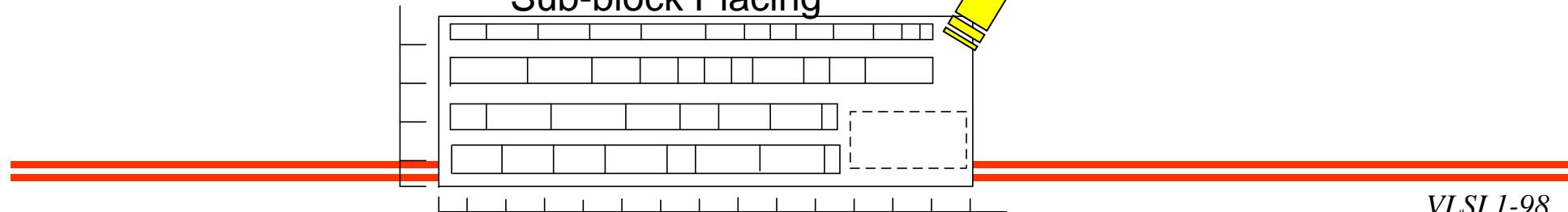
➤ 經過Floorplan 之後，決定Partition 方式及各Block的形狀，大小，放置方式等。再進行Placement將其固定。

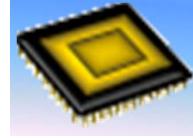


Sub-block Placing



Sub-block Routing

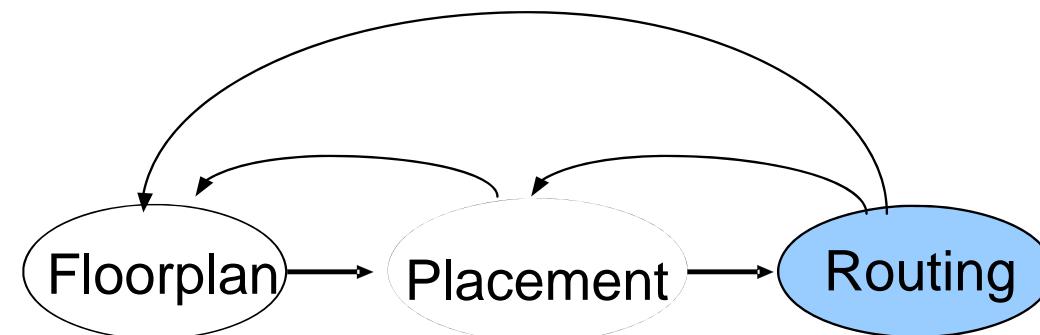
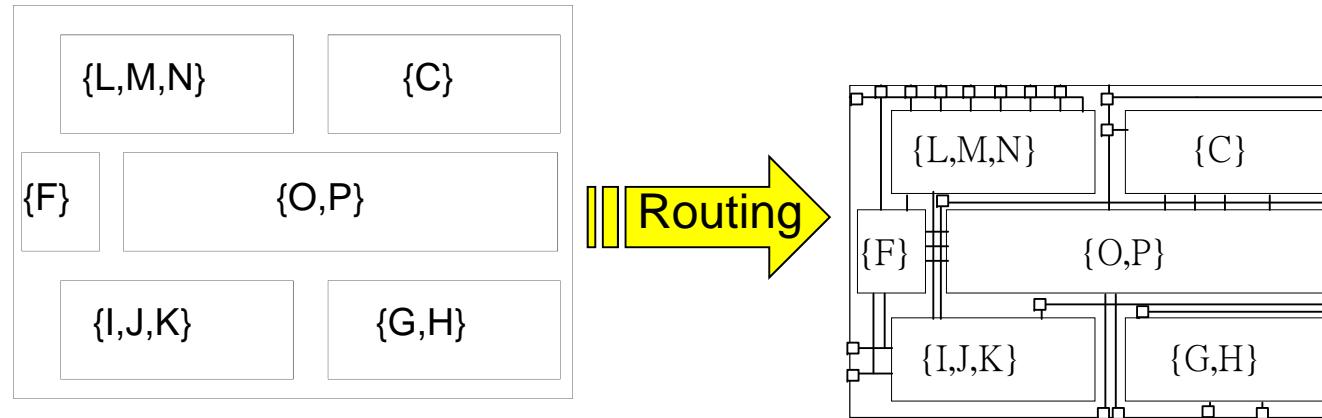


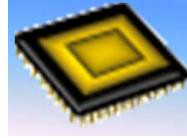


Chapter 1

Routing

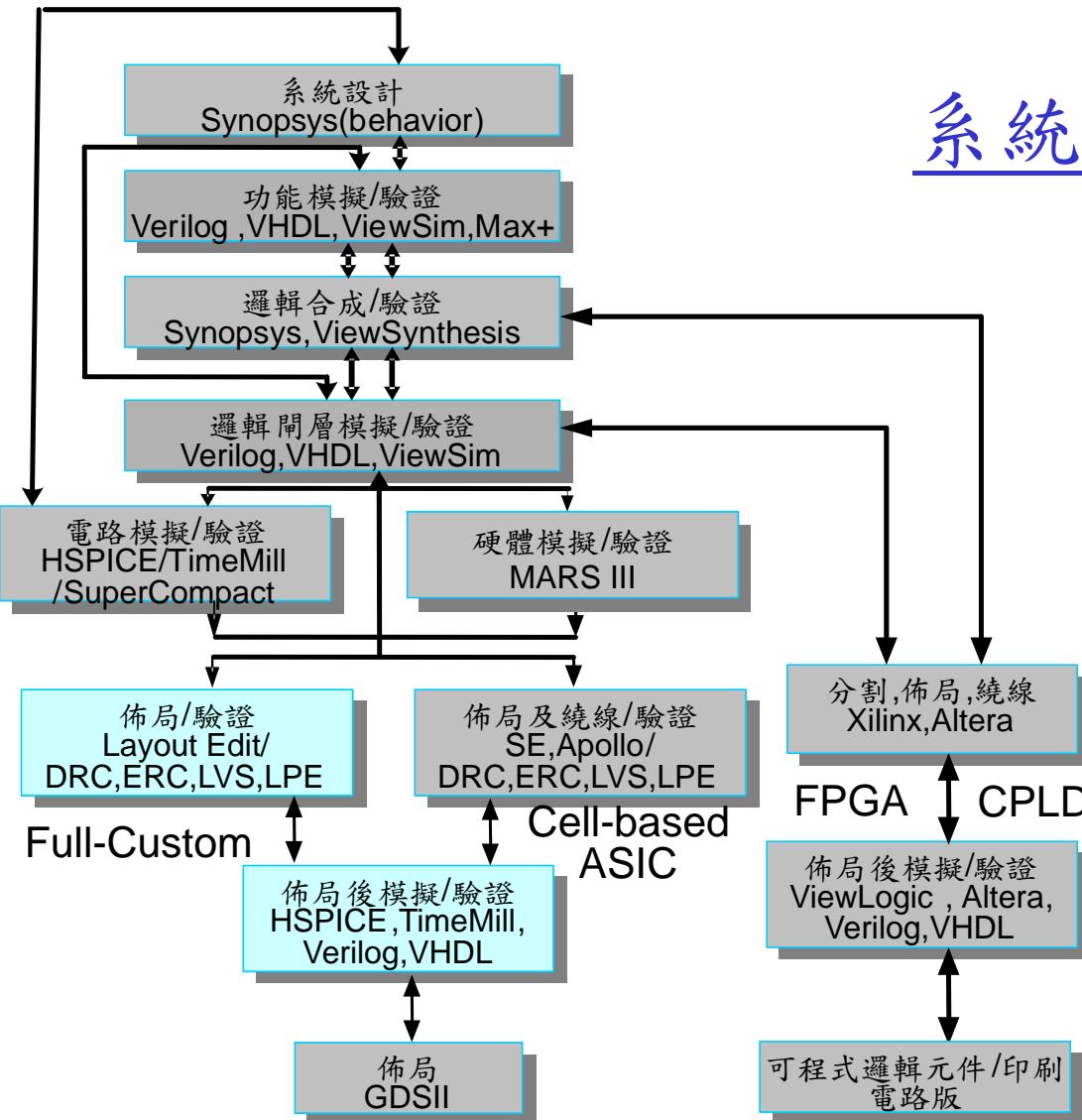
➤ Place 完成後就可以進行Routing，連接各Block以及I/O Pin。

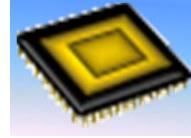




Chapter 1

系統及晶片設計流程

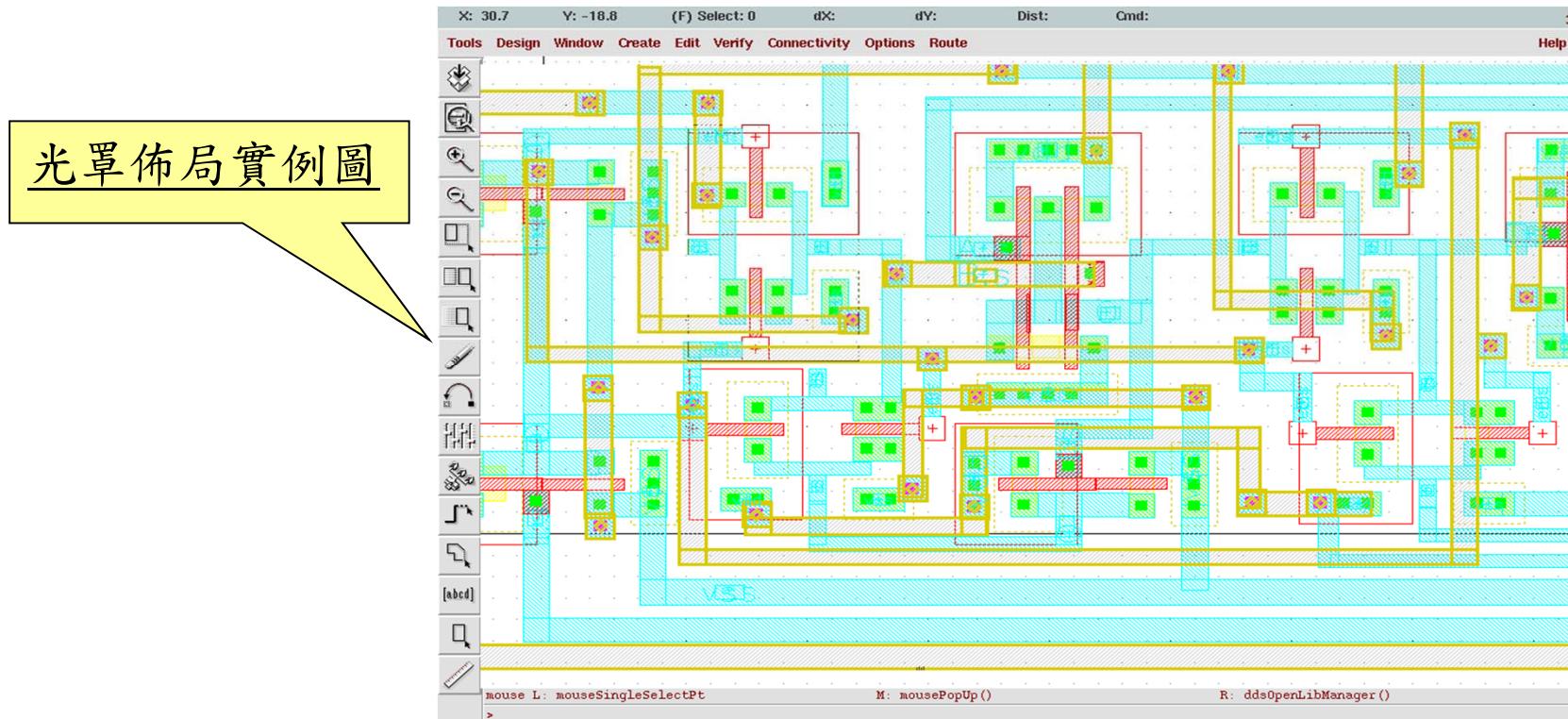


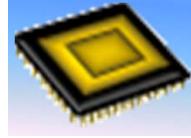


Chapter 1

光罩佈局編輯軟體 Layout Edit

➤電路模擬合乎規格後，就開始根據電路連線繪製佈局，並用 Layout Editor 繪製實體電路的佈局。繪製的佈局是光罩的依據，而光罩為積體電路晶片製作時必備品。





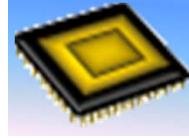
Chapter 1

DRACULA 佈局驗證軟體

➤ DRACULA用以檢查光罩佈局正確與否，包括：

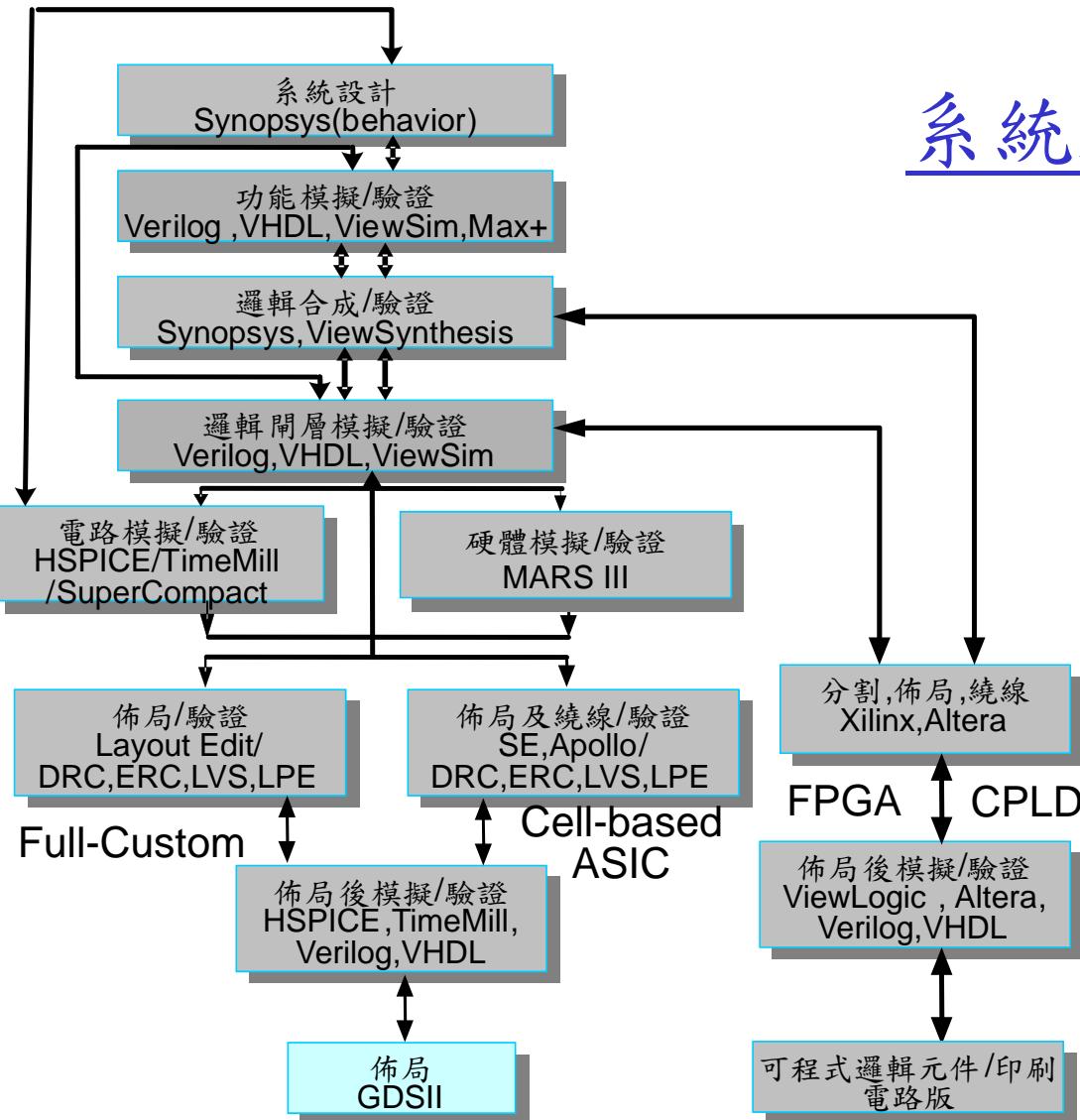
Design Rule Check, Electrical Rule Check, Layout Versus
SLayout Parameter Extraction.

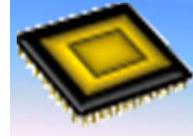
- (1) **DRC** >> 檢查佈局是否合乎工廠所要求的佈局幾何空間規定。
 - (2) **ERC** >> 檢查佈局是否有不正常的電氣特性。如短路或浮接等。
 - (3) **LVS** >> 檢查佈局是否與原先的設計線路一致。
 - (4) **LPE** >> 由佈局抽取出電晶體及寄生電路元件如寄生電容等，以 netlist 形式表示。
- 完成(3)(4)之後，應再使用 HSPICE 或 TimeMill 電路模擬軟體作最後的驗證（post layout simulation）。



Chapter 1

系統及晶片設計流程

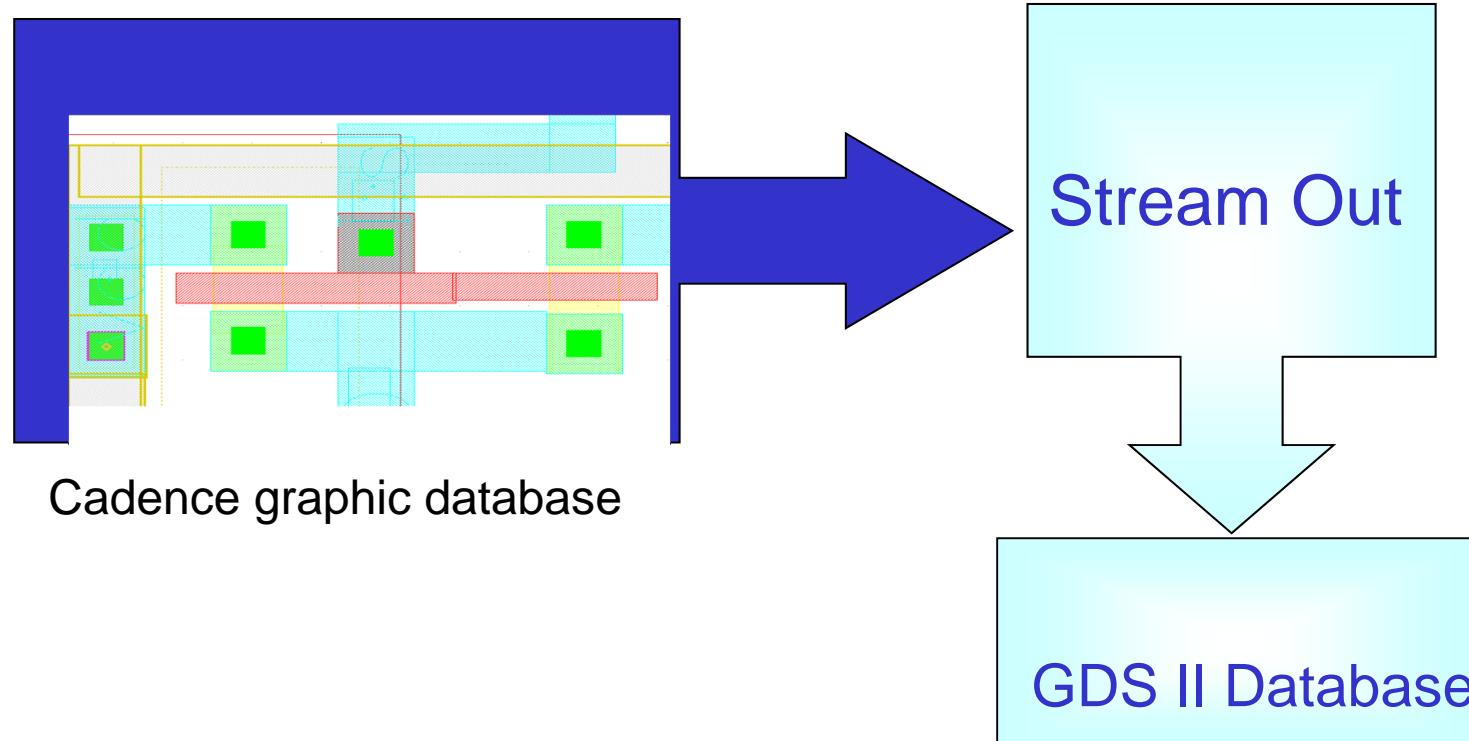


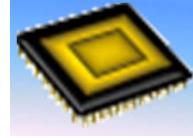


Chapter 1

GDS II OUT

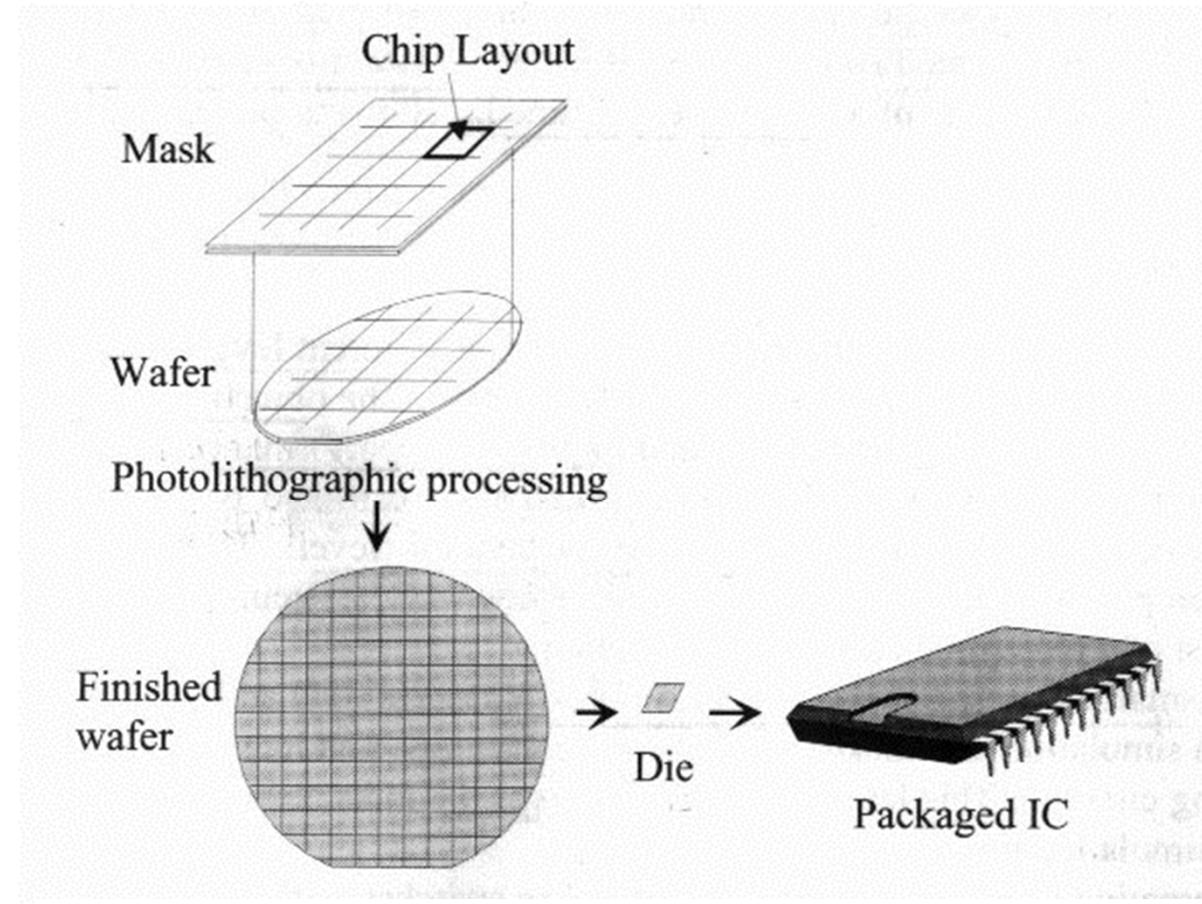
- GDS II format 是一種工業界佈局資料交換的標準格式。當佈局設計完成後，須先將佈局設計轉換成GDS II format，以便製作光罩。在OPUS中可利用 Stream Out 方式將graphic database轉換成GDS II format database。

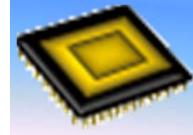




Chapter 1

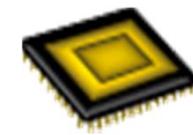
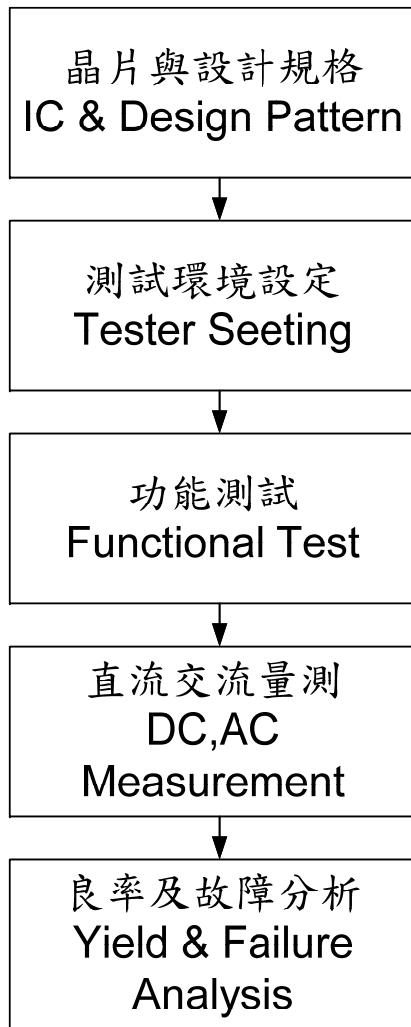
Fabrication





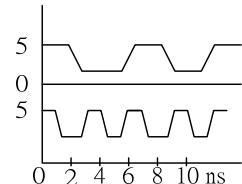
Chapter 1

晶片測試流程

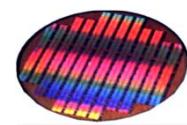
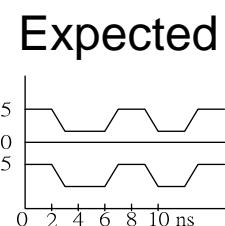


Input 0000
0001
0010

Output 0000
0001
0010



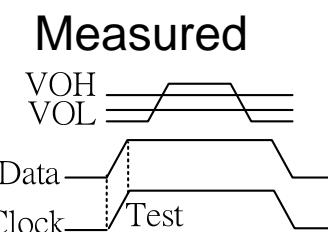
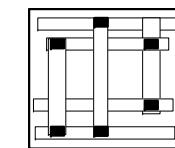
VS

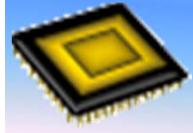


+



=

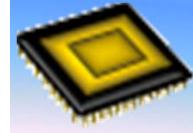




Chapter 1

Application-Specific Integrated Circuits

- an ASIC is a circuit which **performs a specific function** in a particular application
- in the ASIC market, it is important to reduced the manufacturing cost and the time to market
- Mask-programmed **gate-array** (or simply gate-array)
 - one of **special architectures** for these ASIC
 - consist of an array of unconnected gate cells prefabricated on a chip
 - one can **personalize a gate array by adding appropriate metal interconnections** to it
 - **cost is lower** than a full-custom chip: since gate-arrays can be mass produced and used in many different designs
 - designs are **more restrictive than full-custom** designs since all transistors are of **fixed sizes**

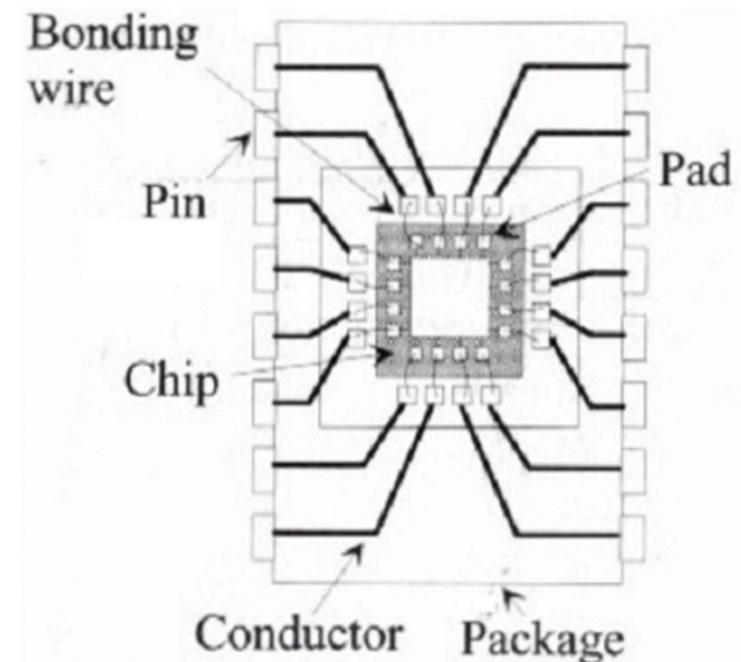


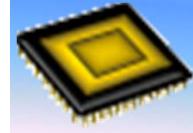
Chapter 1

Packaging

➤ Chip layout including a circuit and a ring of bounding pads (pad frame)

- signal and power connections of the circuit are connected to the bounding pads
- the size of a **bounding pad** is typically $100 \times 100 \mu m^2$
- a signal buffer and a protection circuit are provided for each connection between a bounding pad and the circuit
- the protection circuit protects the circuit against potential damages (e.g., static electricity)

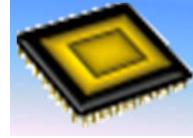




Chapter 1

Chip mounted in an IC package

- Bounding wires connect bounding pads to the pins of an IC package
- To fully utilize the processing power of a chip, enough I/O pins must be provided
- Dual-in-line package: there is a severe limitation on the number of pins available



Chapter 1

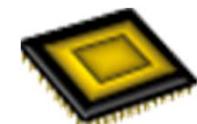
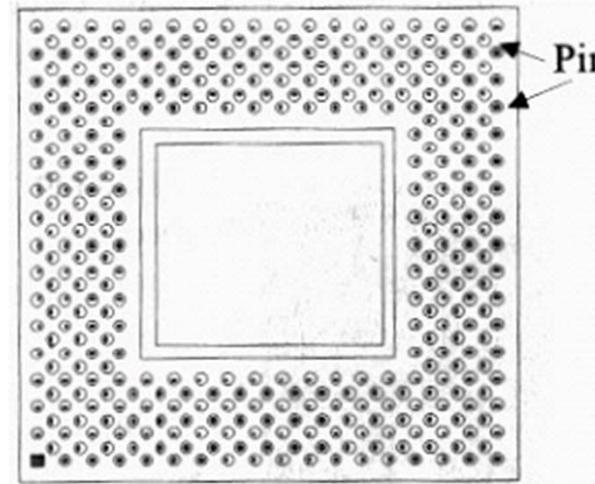
Bottom view of a pin grid array (PGA) package

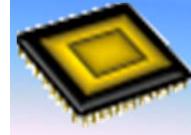
➤ Pin grid array (PGA) package

- arrange pins at the bottom of the package and can provide 400 or more pins

➤ Ball grid array (BGA) package

- Has a similar array but has replaced the pins with solder bumps (balls)

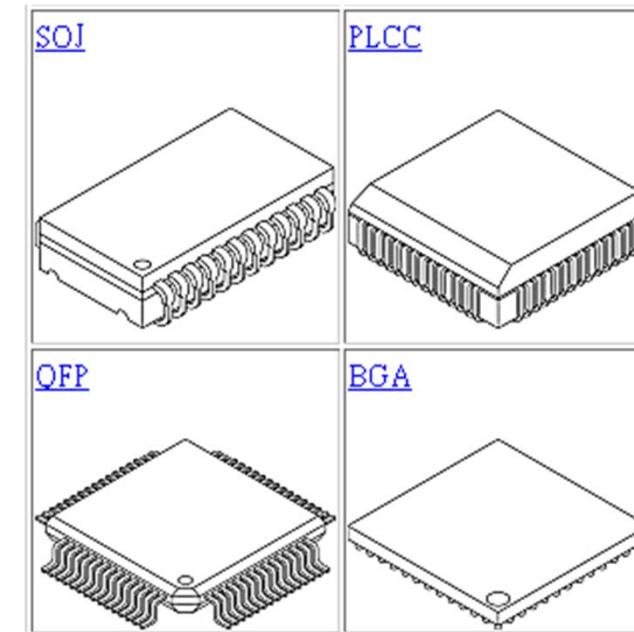
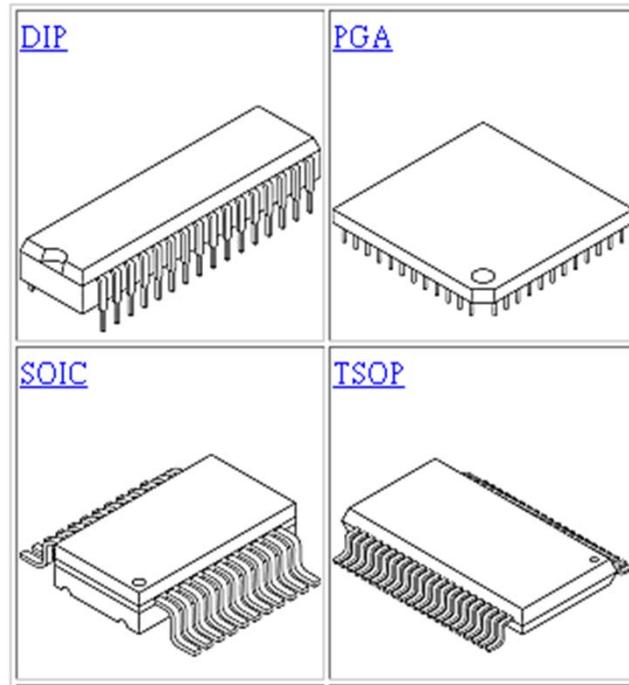


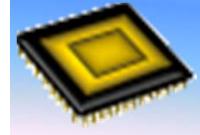


Chapter 1

Another important issue with packaging

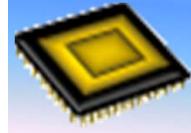
- its capability of **heat dissipation**
- **manufacturer's specification** should be carefully evaluated before a package is selected





Chapter 1

Appendix



Chapter 1

CIC 提供的IC相關設計軟體

Cell library

- ◆ [0.13um TSMC/Artisan Design Kit](#)
- ◆ [0.18um TSMC/Artisan Design Kit](#)
- ◆ [0.18um UMC/Artisan Design Kit](#)
- ◆ [0.35um TSMC/TSMC Design Kit](#)

Debugger/Rule Checker/Testbench Auto

- ◆ [CosmosScope](#)
- ◆ [Debussy](#)
- ◆ [LEDA](#)
- ◆ [VERA](#)

FPGA Suite tool

- ◆ [FPGA Advantage](#)
- ◆ [Foundation Series](#)
- ◆ [ISE Foundation](#)
- ◆ [MaxPlus II](#)
- ◆ [Quartus II](#)

Formal / Code Coverage

- ◆ [Conformal LEC](#)
- ◆ [Formality](#)
- ◆ [Verification Navigator](#)

Circuit Simulator

- ◆ [Analog Artist \(Spectre\)](#)
- ◆ [Eldo](#)
- ◆ [HSPICE](#)
- ◆ [NanoSim](#)
- ◆ [SMASH](#)
- ◆ [Timemill/Powermill](#)

Design For Testability

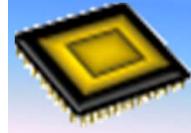
- ◆ [DFT](#)
- ◆ [DFT Compiler](#)
- ◆ [Syntest](#)
- ◆ [TetraMAX](#)
- ◆ [Verifault](#)

FPGA Synthesizer

- ◆ [FPGA-Compiler II](#)
- ◆ [Leonardo Spectrum](#)
- ◆ [Precision Synthesis](#)
- ◆ [Synplify Pro](#)

IP

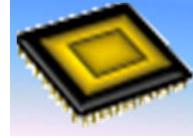
- ◆ [ARM CPU](#)
- ◆ [SUNPLUS CPU](#)
- ◆ [Xtensa CPU](#)



Chapter 1

CIC 提供的IC相關設計軟體

IP Center	Layout Editor
◆ IP Gear	◆ Laker ◆ Virtuoso-XL
Others (Physical Design)	PCB
◆ TCAD	◆ Allegro
Physical Extraction	Physical Verification/Analysis
◆ Calibre xRC(PEX) ◆ Fire&ICE ◆ Star-RCXT	◆ Assura ◆ Calibre DRC/ERC/LVS ◆ Dracula ◆ Hercules ◆ RailMill ◆ VoltageStorm
Place & Route	RF Simulator
◆ Apollo ◆ Astro ◆ Magma Blast Series ◆ SOC Encounter ◆ Silicon Ensemble	◆ ADS ◆ Designer ◆ RFDE ◆ Spectre RF
SOPC flow	STA/SIA/PA
◆ Excalibur-ARM ◆ Excalibur-Nios	◆ Pathmill ◆ PrimePower ◆ PrimeTime ◆ PrimeTime SI
Schematic Editor	Simulator
◆ Composer	◆ ModelSim ◆ NC SIM ◆ NC VHDL ◆ NC-Verilog ◆ Scirocco Simulator ◆ Spexsim ◆ VCS ◆ Verilog-XL



Chapter 1

CIC 提供的IC相關設計軟體

SoC Emulation

- ◆ [System Explorer](#)

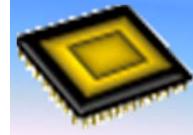
System Level

- ◆ [COSSAP](#)
- ◆ [CoCentric Fixed-Point Designer](#)
- ◆ [CoCentric System Studio](#)
- ◆ [ConvergenSC](#)
- ◆ [SPW](#)
- ◆ [Seamless](#)

Synthesis

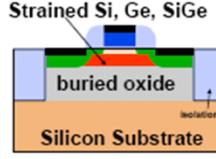
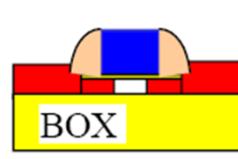
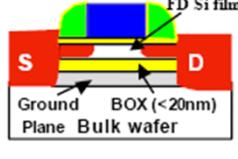
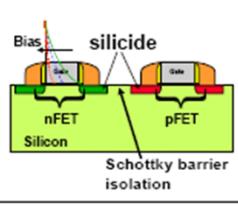
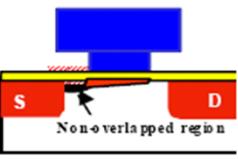
- ◆ [Ambit](#)
- ◆ [CoCentric SystemC Compiler](#)
- ◆ [Design Compiler](#)
- ◆ [Physical Compiler](#)
- ◆ [Power Compiler](#)

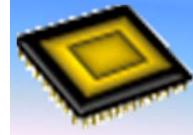
http://www.cic.org.tw/cic_v13/dsd/dsdsoftq01.jsp?ListType=Functionality



Chapter 1

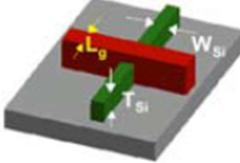
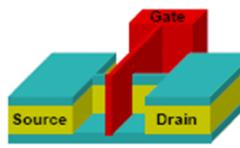
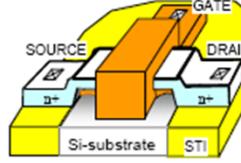
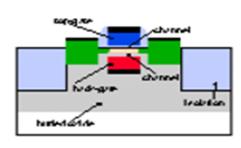
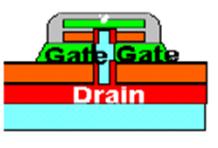
Single-gate Non-classical CMOS Technologies

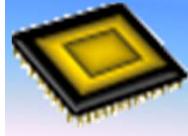
Device	Transport-enhanced FETs	Ultra-thin Body SOI FETs		Source/Drain Engineered FETs	
					
Concept	Strained Si, Ge, SiGe, SiGeC or other semiconductor; on bulk or SOI	Fully depleted SOI with body thinner than 10 nm	Ultra-thin channel and localized ultra-thin BOX	Schottky source/drain	Non-overlapped S/D extensions on bulk, SOI, or DG devices
Application/Driver	HP CMOS	HP, LOP, and LSTP CMOS	HP, LOP, and LSTP CMOS	HP CMOS	HP, LOP, and LSTP CMOS
Advantages	<ul style="list-style-type: none">▪ High mobility	<ul style="list-style-type: none">▪ Improved subthreshold slope▪ No floating body▪ Potentially lower E_{eff}	<ul style="list-style-type: none">▪ SOI-like structure on bulk▪ Shallow junction by geometry▪ Junction silicidation as on bulk▪ Improved S-slope and SCE	<ul style="list-style-type: none">▪ Low source/drain resistance	<ul style="list-style-type: none">▪ Reduced SCE and DIBL▪ Reduced parasitic gate capacitance



Chapter 1

Multiple-gate Non-classical CMOS Technologies

Device	Multiple Gate FETs				
	N-Gate ($N > 2$) FETs	Double-gate FETs			
					
Concept	Tied gates (number of channels >2)	Tied gates, side-wall conduction	Tied gates planar conduction	Independently switched gates, planar conduction	Vertical conduction
Application/ Driver	HP, LOP, and LSTP CMOS	HP, LOP, and LSTP CMOS	HP, LOP, and LSTP CMOS	LOP and LSTP CMOS	HP, LOP, and LSTP CMOS
Advantages	<ul style="list-style-type: none">▪ Higher drive current▪ 2× thicker fin allowed	<ul style="list-style-type: none">▪ Higher drive current▪ Improved subthreshold slope▪ Improved short channel effect	<ul style="list-style-type: none">▪ Higher drive current▪ Improved subthreshold slope▪ Improved short channel effect	<ul style="list-style-type: none">▪ Improved short channel effect	<ul style="list-style-type: none">▪ Potential for 3D integration



Chapter 1

Reference

- Douglas A Pucknell, Kamran Eshraghian ,Basic VLSI Design 3rd Ed, Prentice Hall .
- M Michael Vai, VLSI Design, CRC Press, 2000
- D.A.Pucknell, K.Eshraghian, Basic VLSI Design, 3rd Ed, Prentice Hall,1994
- Weste and Eshraghian,Principles of VLSI Design--A Systems Perspective, Addison-Wesley,2nd,1993
- C.Y. Chang and S.M. Sze,ULSI DEVICES, John Wiely & Sons,2002
- 國家晶片系統設計中心,Dracula Training Manual, 2002.07
- 中央大學電機工程學系,鄭國興老師
- Intel
- Digital Integrated Circuits^{2nd}
- 南台科技大學電子系,楊博惠老師

回本節首頁