Lecture 1 Introduction to IC Design

Outline

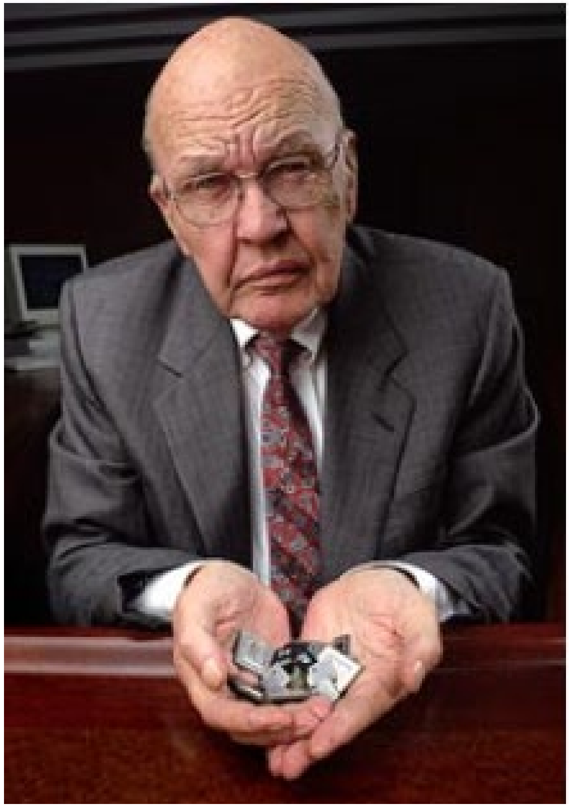
- 1.1 History and Road Map of IC Industry
- 1.2 Introduction to IC Industry
- 1.3 FPGA and ASIC Design
- 1.4 Fundamentals of Verilog HDL Design
- 1.5 Attributes of Verilog HDL Design

Outline

- 1.1 History and Road Map of IC Industry
- 1.2 Introduction to IC Industry
- 1.3 FPGA and ASIC Design
- 1.4 Fundamentals of Verilog HDL Design
- 1.5 Attributes of Verilog HDL Design

1.1.1 A Brief History of IC Design Industry

- A. The First IC (1958)
  - Jack Kilby
  - Nobel Prize in Physics -2000



https://en.wikipedia.org/wiki/Jack_Kilby

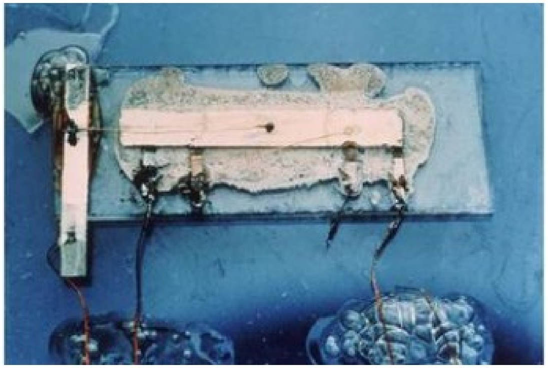


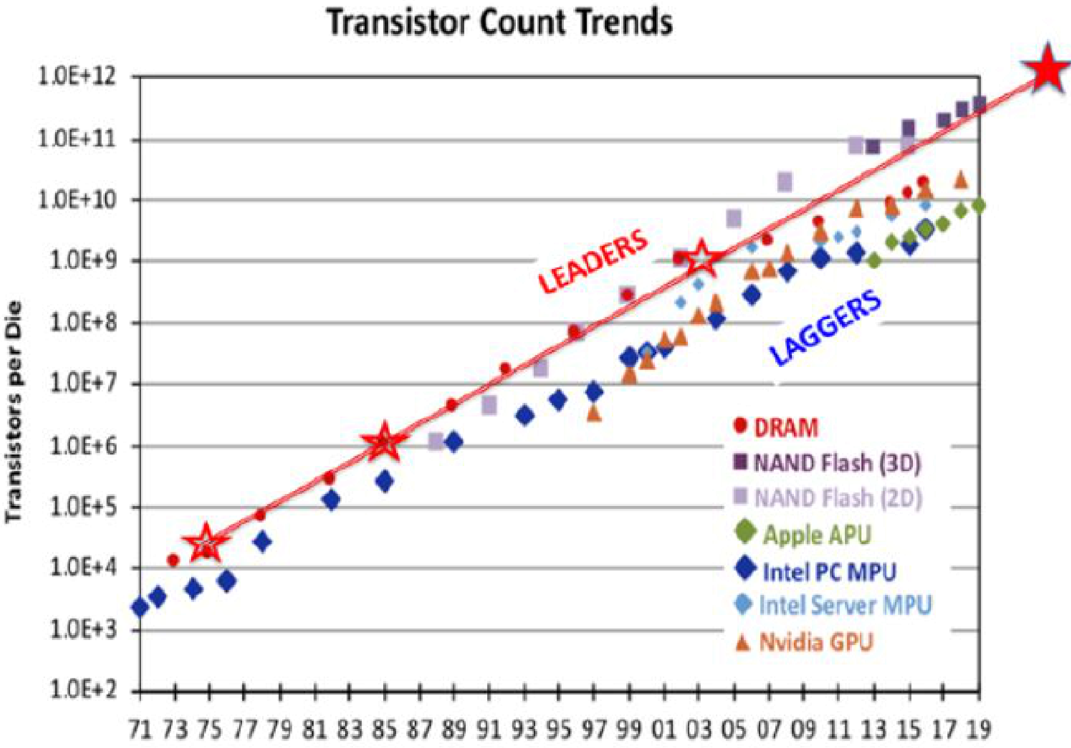
FIGURE 1.1
First IC [10].

1.1.1 A Brief History of IC Design Industry

- B. Moore’s Law (1965)

Moore's law is the observation that the number of transistors in a dense integrated circuit (IC) doubles about every two years.

https://en.wikipedia.org/wiki/Robert_Noyce



Sources: Intel, SIA, Wikichip, IC Insights

FIGURE 1.2
Transistor Count Trends [11].

1.1.1 A Brief History of IC Design Industry

- C. Advancements in the 1970s and 1980s
- D. Multi-Core SoCs (1990s and Early 2000s)

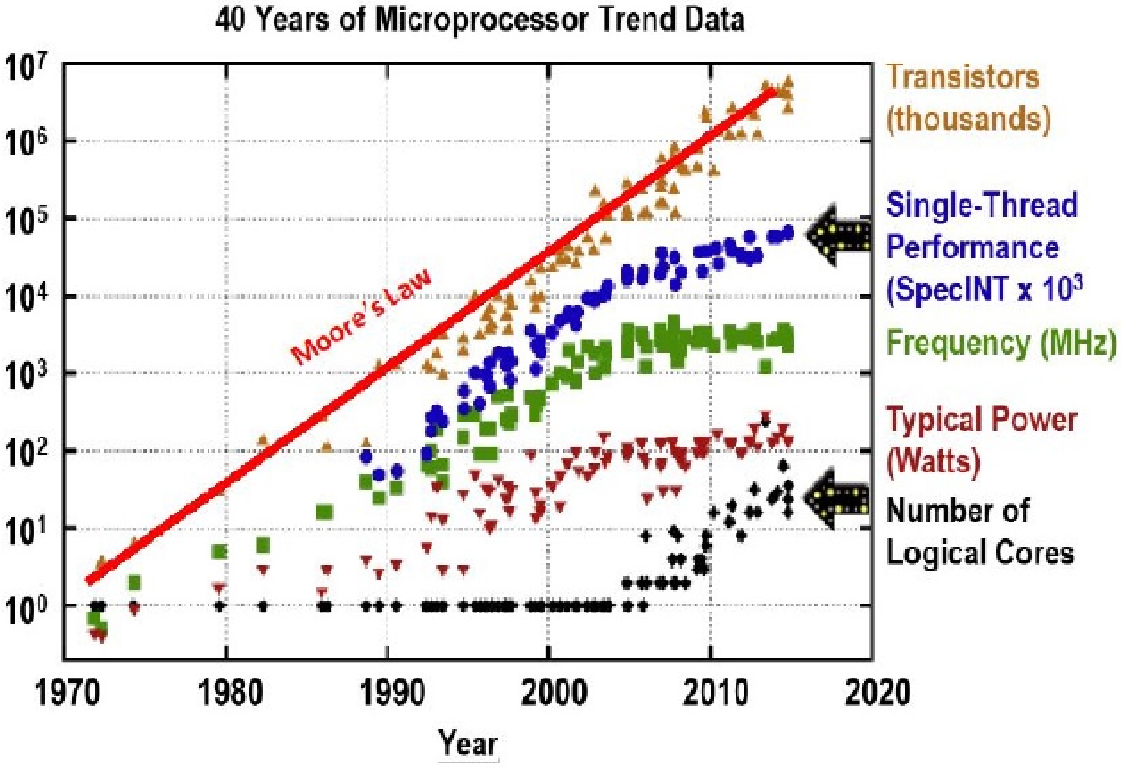


FIGURE 1.3
40 Years of Microprocessor Trend Data: Adoption of Multicore Architecture Allowed Moderate Performance Increase in CPU Design Within Power Limits [2].

1.1.2 Today's IC Industry and Technology

- A. Maximum CPU Clock Rate
- B. GPUs and Neural Engines

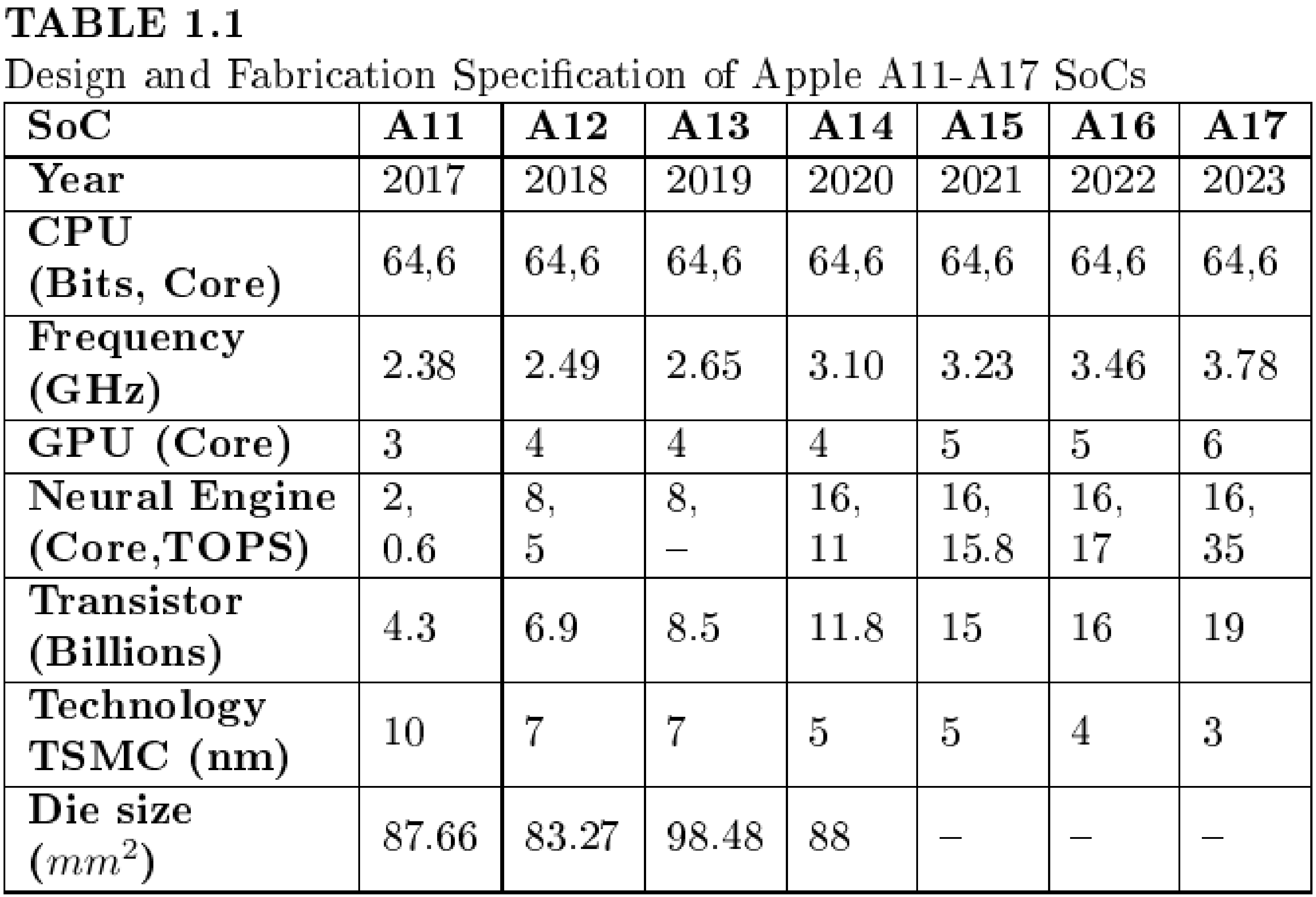


TABLE 1.1

Design and Fabrication Specification of Apple A11-A17 SoCs

SoC	A11	A12	A13	A14	A15	A16	A17
Year	2017	2018	2019	2020	2021	2022	2023
CPU (Bits, Core)	64,6	64,6	64,6	64,6	64,6	64,6	64,6
Frequency (GHz)	2.38	2.49	2.65	3.10	3.23	3.46	3.78
GPU (Core)	3	4	4	4	5	5	6
Neural Engine (Core,TOPS)	2, 0.6	8, 5	8, –	16, 11	16, 15.8	16, 17	16, 35
Transistor (Billions)	4.3	6.9	8.5	11.8	15	16	19
Technology TSMC (nm)	10	7	7	5	5	4	3
Die size (mm^2)	87.66	83.27	98.48	88	–	–	–

1.1.2 Today's IC Industry and Technology

- C. Transistor Technology
- D. Moore's Wall

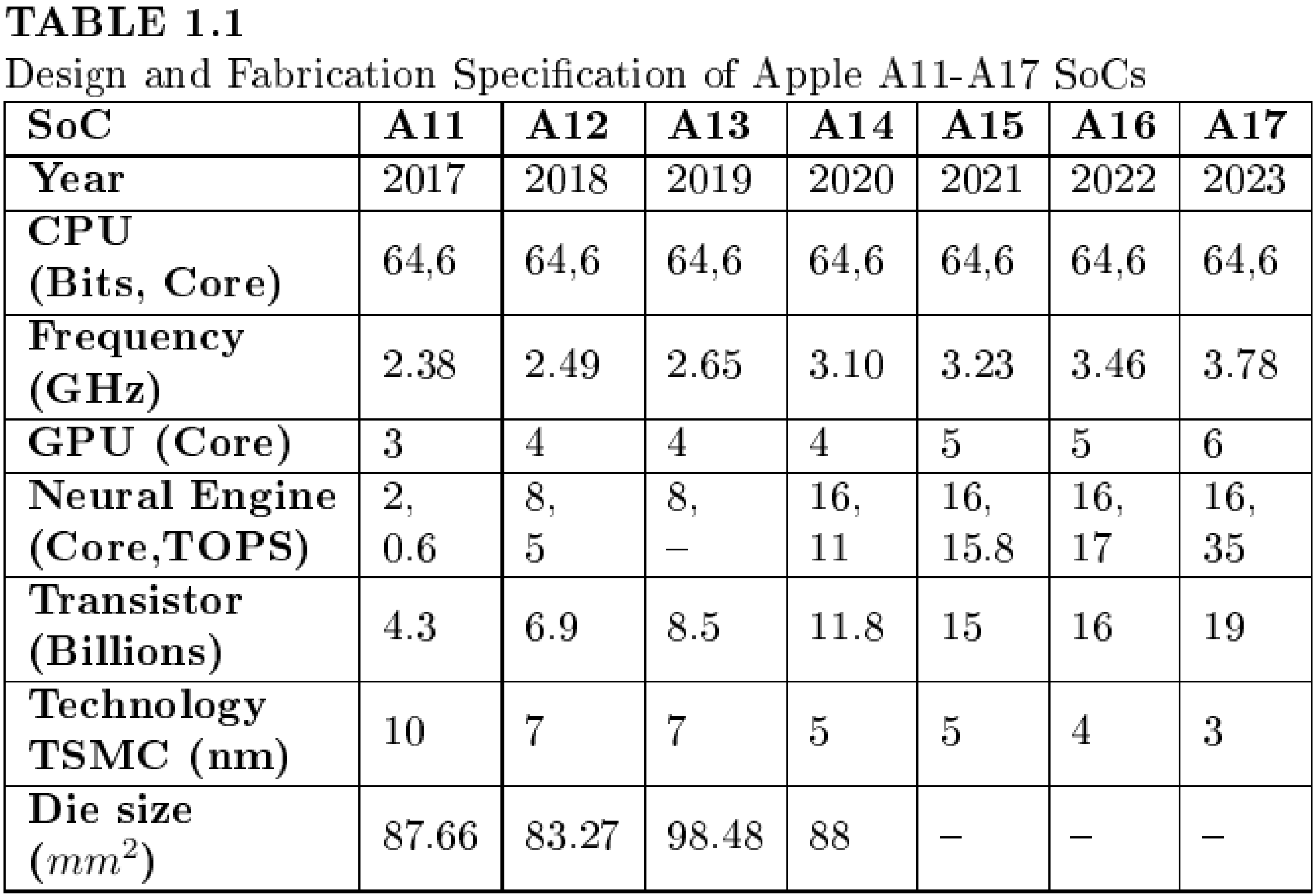


TABLE 1.1

Design and Fabrication Specification of Apple A11-A17 SoCs

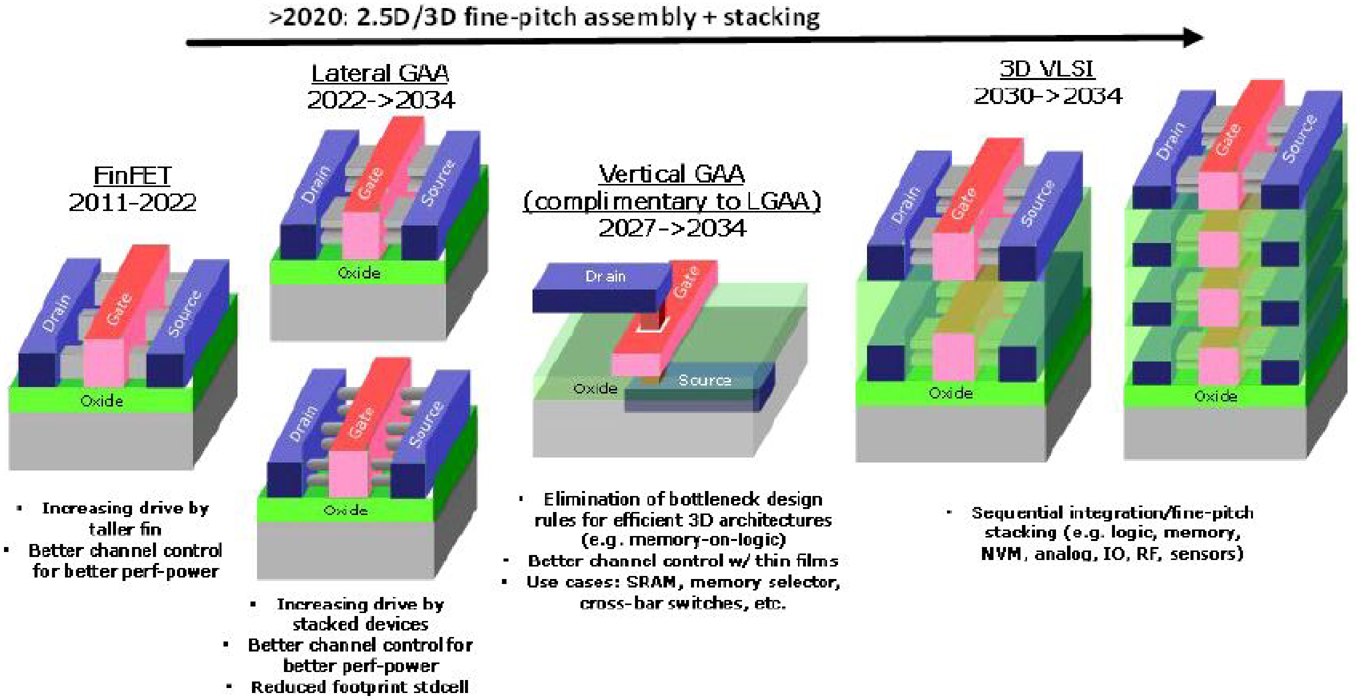
SoC	A11	A12	A13	A14	A15	A16	A17
Year	2017	2018	2019	2020	2021	2022	2023
CPU (Bits, Core)	64,6	64,6	64,6	64,6	64,6	64,6	64,6
Frequency (GHz)	2.38	2.49	2.65	3.10	3.23	3.46	3.78
GPU (Core)	3	4	4	4	5	5	6
Neural Engine (Core,TOPS)	2, 0.6	8, 5	8, –	16, 11	16, 15.8	16, 17	16, 35
Transistor (Billions)	4.3	6.9	8.5	11.8	15	16	19
Technology TSMC (nm)	10	7	7	5	5	4	3
Die size (mm^2)	87.66	83.27	98.48	88	–	–	–

1.1.3 Beyond Moore's Law

- The end of Moore's law
  - International Technology Roadmap for Semiconductors (ITRS), v2.0, 2015: the physical gate length of transistors would shrink until at least 2028
  - ASML (the world's supplier and leader in the semiconductor industry, leading corp. of the photolithography systems.): the ability to reduce features (line and space) would reach final limits around 7-8 nm by the end of this decade
- Beyond Moore's law
  - New Materials and New Transistor Structures
  - Specialized Design on Future High-Performance Computing
  - Chiplet and Heterogeneous Integration
  - Neuromorphic Computing and Quantum Computing

1.1.3 Beyond Moore’s Law

- New materials
  - Materials having high-hole mobility (III-V) and high electron mobility (germanium) are mainly carried out from 2003 to 2024
- New transistor structures
  - 2011 to 2022, 2D -> 2.5D of FinFET -> gate-all-around (GAA)
  - 2025 to 2040, 3D VLSI and heterogeneous integration



>2020: 2.5D/3D fine-pitch assembly + stacking

FinFET
2011-2022

- Increasing drive by taller fin
- Better channel control for better perf-power

Lateral GAA
2022->2034

- Increasing drive by stacked devices
- Better channel control for better perf-power
- Reduced footprint stdcell

Vertical GAA
(complimentary to LGAA)
2027->2034

- Elimination of bottleneck design rules for efficient 3D architectures (e.g. memory-on-logic)
- Better channel control w/ thin films
- Use cases: SRAM, memory selector, cross-bar switches, etc.

3D VLSI
2030->2034

- Sequential integration/fine-pitch stacking (e.g. logic, memory, NVM, analog, IO, RF, sensors)

Figure ES48 [ITRS V2.0] Change in the MOSFET device architecture from the 2D planar through 2.5D FinFETs to 3D monolithic VLSI with GAA

1.1.3 Beyond Moore’s Law

- Specialized Design on Future High-Performance Computing
  - Industry:
    - ASIC Accelerators: Google Tensor Processing Unit (TPU) and Tesla Dojo D1
    - FPGA Accelerators: Microsoft’s FPGA Configurable Cloud, Project Catapult for FPGA-accelerated search, etc.
  - Science-target accelerators:
    - D.E. Shaw’s Anton (accelerates molecular dynamics simulations nearly 180× over contemporary HPC systems),
    - GRAPE series of specialized accelerators for cosmology and molecular dynamics
    - Berkeley DFT (Dense Functional Theory) accelerator, etc.
  - Architecture specialization
    - International Symposium on Computer Architecture (ISCA) workshop on the future of computing research beyond 2030 concluded that architecture specialization is nearly inevitable given current architecture trends.

1.1.3 Beyond Moore’s Law

- New chip architectures
  - Neuromorphic computing
    - Mimic neuro-biological architectures present in the nervous system which is intrinsically 100% parallel computing
    - Using electronic analog circuits and devices such as oxide-based memristors
  - Quantum logic and circuits
    - Quantum von Neumann architecture
    - Qubit with superposition of 0 and 1
    - Quantum entanglement to allow the use of massive quantum parallelism on a single quantum core
    - Today’s Quantum computing: works with classical computing

1.1.4 Preview

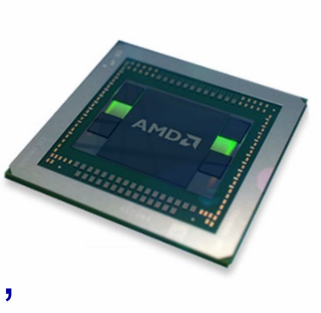
- Preview
  - With a history spanning over 50 years, the IC design industry has witnessed substantial progress.
    - This evolution encompasses various design platforms, increased automation within the design process through the use of comprehensive Electronic Design Automation (EDA) toolsets, and innovative hardware description solutions.
  - This chapter will provide an overview of the fundamental knowledge and background necessary for hardware descriptions and IC design.
    - This includes essential information and prerequisites for register-transfer level (RTL) design using Verilog Hardware Description Language (HDL).

Outline

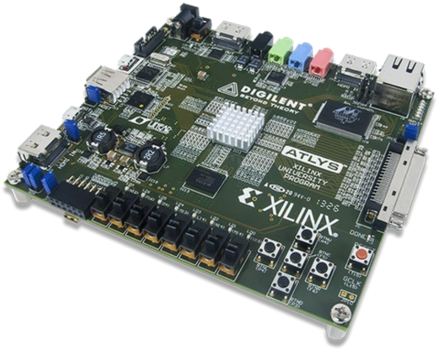
- 1.1 History and Road Map of IC Industry
- 1.2 Introduction to IC Industry
- 1.3 FPGA and ASIC Design
- 1.4 Fundamentals of Verilog HDL Design
- 1.5 Attributes of Verilog HDL Design

1.2.1 What are FPGA and ASIC

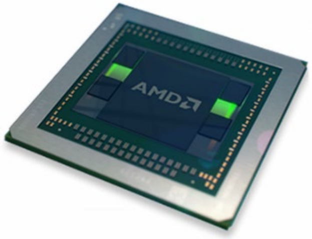
- Application-specific Integrated Circuit (ASIC)
  - An ASIC is a specialized chip designed for a specific purpose, tailored to perform dedicated functions.
  - Most IC design companies we see today, such as Intel, AMD, NVIDIA, Qualcomm, and Broadcom, are primarily known for their ASIC design capabilities, but many of them also have dedicated FPGA development teams.
  - In modern applications, ASICs often take the form of **system-on-chip (Soc)** designs, integrating multiple components like CPUs, memory blocks, interfaces, and peripherals into a single chip.



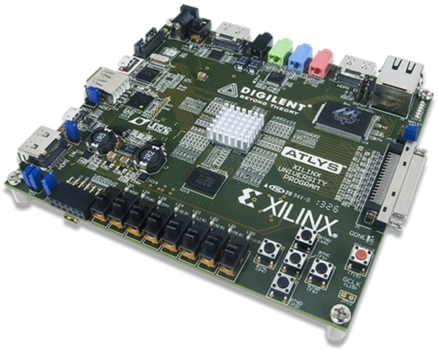
- Field-programmable Gate Array (FPGA)



1.2.1 What are FPGA and ASIC



- Application-specific Integrated Circuit (ASIC)
- Field-programmable Gate Array (FPGA)
  - An FPGA is a semiconductor device that boasts configurable logic components, which encompass programmable logic blocks (CLBs), reconfigurable interconnects, and dedicated hard-silicon blocks designed for specific functions.
  - The fundamental building blocks within CLBs typically consist of look-up tables (LUTs), multiplexers, and flip-flops, providing FPGAs with an exceptionally versatile and adaptable nature.



1.2.1 What are FPGA and ASIC

- ASIC vs. FPGA
  - Applications:
    - ASICs offer the advantage of high performance and power efficiency for specific tasks
    - FPGAs provide flexibility and reconfigurability, making them suitable for prototyping, rapid development, and applications that require frequent updates or changes to the design.
  - Design Flow:
    - The ASIC design and fabrication process involves a very complex procedure, starting from the register-transfer level coding to the final chip manufacture steps.
    - FPGA development leverages existing hardware resources, making the FPGA design flow generally simpler compared to the more complex ASIC design flow.
    - In practice, an ASIC design flow often includes the utilization of FPGAs during development for functional and timing validation, allowing engineers to test their designs in a real hardware environment.

1.2.2 Brief Introduction to IC Industry

- ASIC Industry:
  - Integrated Device Manufacturer (IDM)
    - Offers IC products of its own design as well as operates semiconductor fabrication
    - Intel, Samsung, Texas Instruments, etc.
  - Semiconductor foundry
    - Only manufacture devices for other companies, without designing them
    - TSMC, Globalfoundries, UMC, and SMIC
  - Fabless design company
    - Focuses on the research and development of an IC product
    - Majority part of IC industry are fabless companies, or named design house, such as AMD, NVIDIA, Qualcomm, etc.
- FPGA Industry:
  - AMD/Xilinx and Intel/Altera.
  - In 2015, Intel acquired Altera; In 2020, AMD acquired Xilinx.

1.2.3 EDA Tools and Vendors

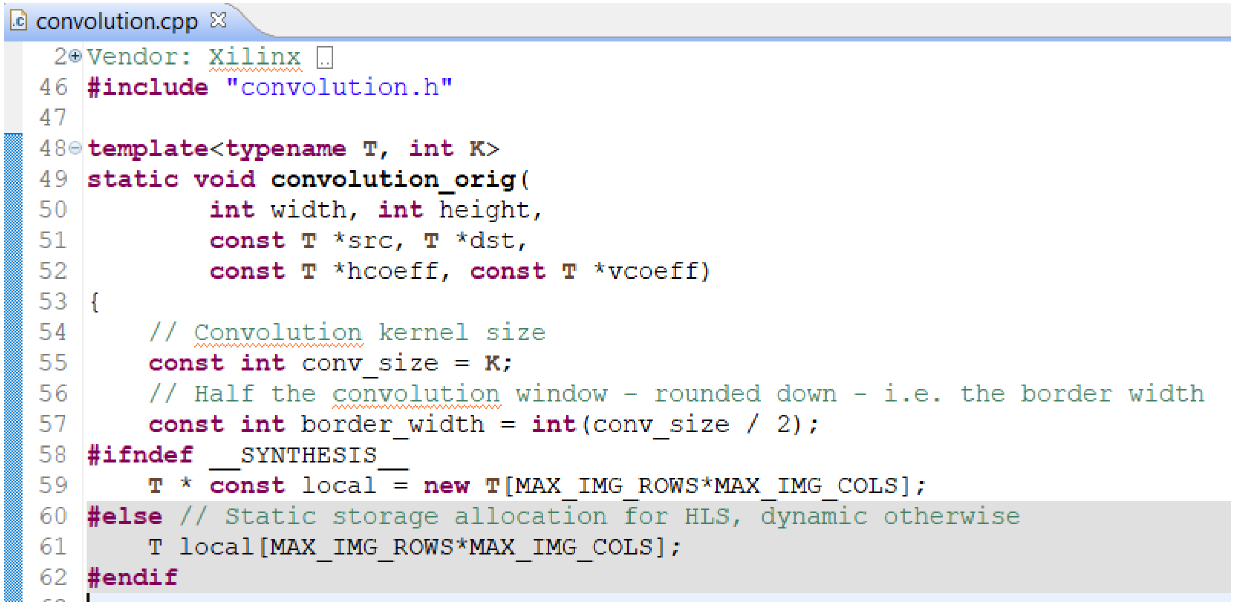
- ASIC Simulation and Synthesis
  - EDA Vendors
    - Cadence: www.cadence.com
    - Synopsys: www.synopsys.com
    - Mentor Graphics (Acquired by Siemens in 2017)
  www.mentorgraphics.com
  - Simulators
    - Synoposys VCS, Cadence NC, Siemens/Mentor Graphics ModelSim. (Advanced version QuestaSim)
  - Synthesis
    - Synopsys Design Compiler, Cadence Genus Synthesis Solution, Mentor Graphics Precision
- FPGA Simulation and Synthesis and Implementation
  - Simulation: the same as ASIC simulation
  - Synthesis
    - AMD/Xilinx Vivado/ISE
    - Intel/Altera Quartus

1.2.4 Hardware Design Solutions

- 1.2.4.1 Hardware Description Language
  - Verilog HDL
    - Introduced by Gateway Design Automation in 1985
    - In 1995, Verilog became the IEEE Standard 1364
    - In 2003, IEEE 1364-2001
  - VHDL (Very High Speed IC HDL)
    - In 1987, was standardized as IEEE 1076-1987
    - In 1994, IEEE 1076-1993
  - Verilog V.S. VHDL
    - Verilog is similar in form of C and VHDL is similar to Ada.
    - Verilog is simpler for typical design, VHDL is more academic.
    - Some industries like aerospace and governments use VHDL, but most other leading IC design companies use Verilog.

1.2.4 Hardware Design Solutions

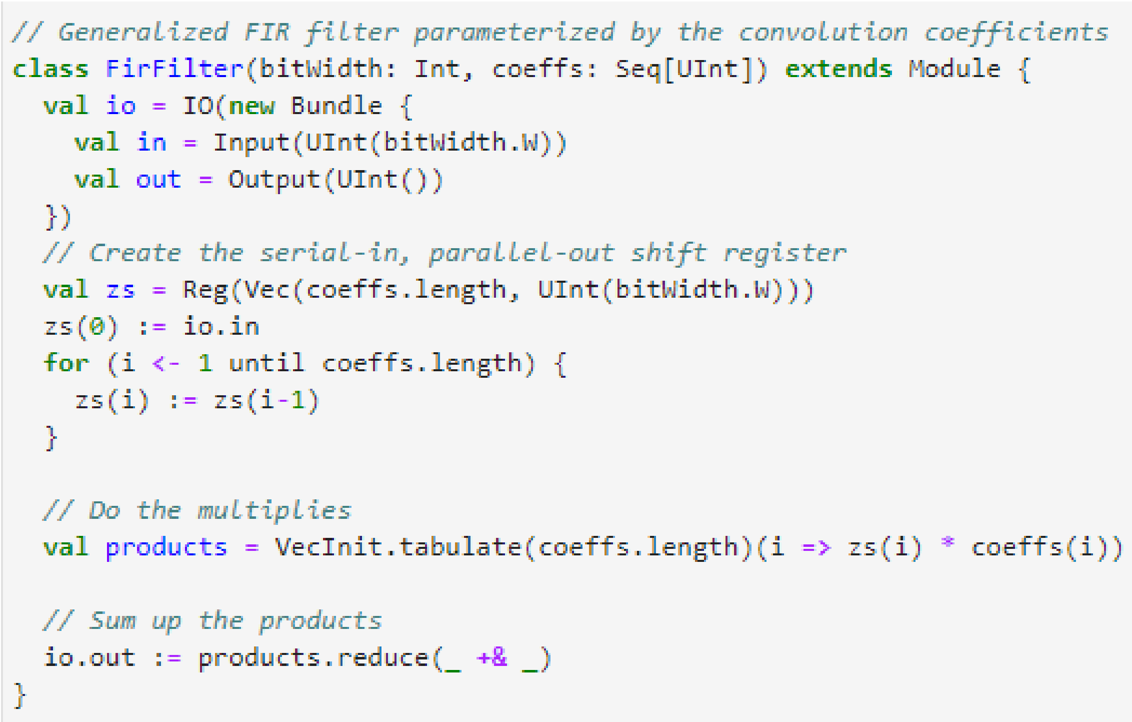
- 1.2.4.2 High-Level Synthesis
  - Motivation: from a complex hardware problem into a much easier computer science problem and popularized chip design in academia.
  - C or C++ and then convert the design into RTL code using the High-Level Synthesis (HLS) tools:
    - Ex. AMD Vivado HLS 2D convolution with line buffer (convolution.cpp)
  - Concerns: a lack of hardware-related descriptions such as the timing and performance considerations



```
Vendor: Xilinx 
#include "convolution.h"

template<typename T, int K>
static void convolution_orig(
        int width, int height,
        const T *src, T *dst,
        const T *hcoeff, const T *vcoeff)
{
    // Convolution kernel size
    const int conv_size = K;
    // Half the convolution window - rounded down - i.e. the border width
    const int border_width = int(conv_size / 2);
#ifndef __SYNTHESIS__
    T * const local = new T[MAX_IMG_ROWS*MAX_IMG_COLS];
#else // Static storage allocation for HLS, dynamic otherwise
    T local[MAX_IMG_ROWS*MAX_IMG_COLS];
#endif
```

1.2.4 Hardware Design Solutions

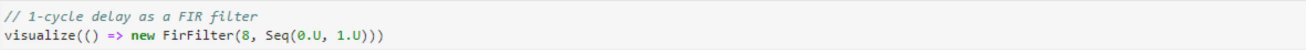
- 1.2.4.3 HDL Code Generators
  - Motivation: Instead of describing hardware directly, there were many explorations for producing RTL with code generators in industry.
  - Concerns: a lack of powerful libraries behind the solutions
  - Chisel HCL
    - An open-source embedded domain-specific language developed at UC Berkeley.
    - Benefits:
      - Scala language
      - Parameterizable
      - Primitives



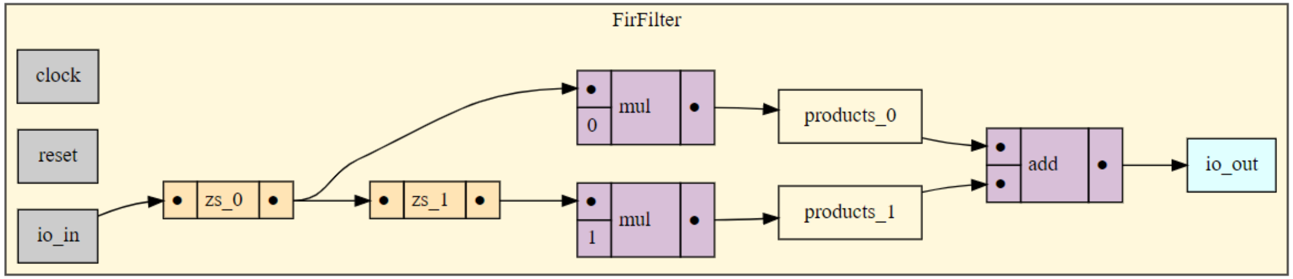
```
// Generalized FIR filter parameterized by the convolution coefficients
class FirFilter(bitWidth: Int, coeffs: Seq[UInt]) extends Module {
  val io = IO(new Bundle {
    val in = Input(UInt(bitWidth.W))
    val out = Output(UInt())
  })
  // Create the serial-in, parallel-out shift register
  val zs = Reg(Vec(coeffs.length, UInt(bitWidth.W)))
  zs(0) := io.in
  for (i <- 1 until coeffs.length) {
    zs(i) := zs(i-1)
  }

  // Do the multiplies
  val products = VecInit.tabulate(coeffs.length)(i => zs(i) * coeffs(i))

  // Sum up the products
  io.out := products.reduce(_ +& _)
}
```

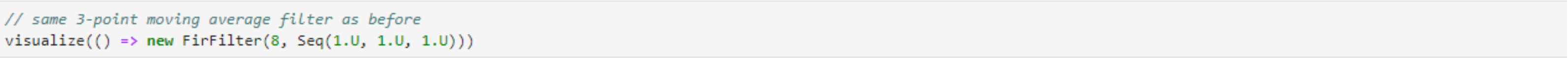
1.2.4 Hardware Design Solutions

- Chisel HCL Example: Parameterizable Design

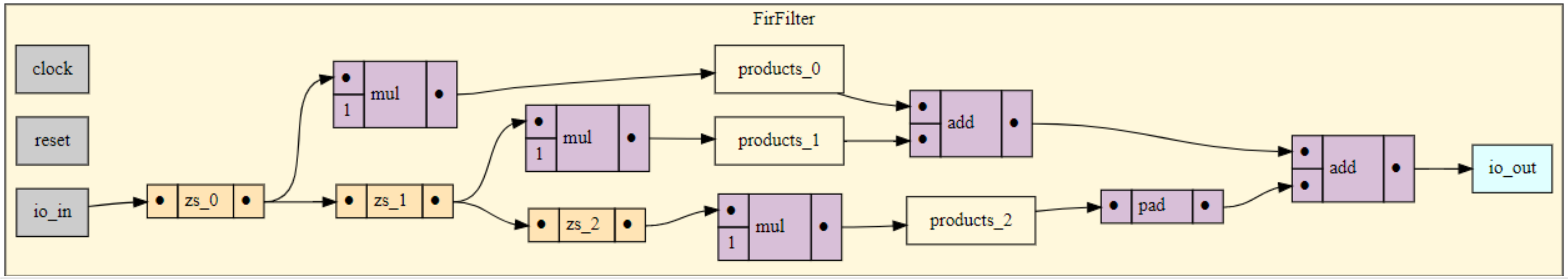


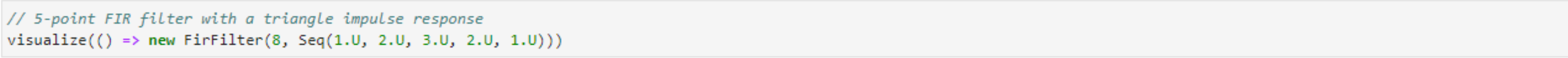
```
// 1-cycle delay as a FIR filter
visualize(() => new FirFilter(8, Seq(0.U, 1.U)))
```



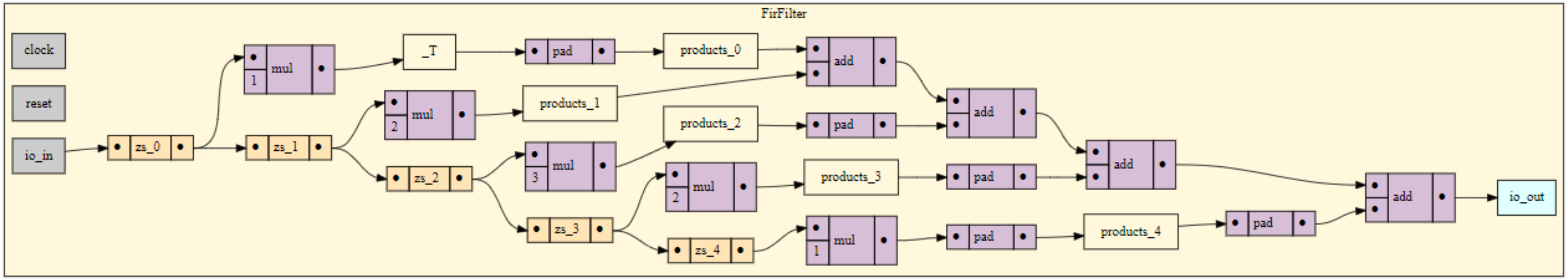


```
// same 3-point moving average filter as before
visualize(() => new FirFilter(8, Seq(1.U, 1.U, 1.U)))
```





```
// 5-point FIR filter with a triangle impulse response
visualize(() => new FirFilter(8, Seq(1.U, 2.U, 3.U, 2.U, 1.U)))
```



1.2.4 Hardware Design Solutions

- 1.2.4.4 SystemVerilog
  - A superset of Verilog-2005 focusing on many new features and capabilities to aid design verification.
  - It became IEEE standard 1800 in 2005 and updated in 2012 as IEEE 1800-2012.
  - Supports structured and object-oriented paradigm based on Verilog and C++. SystemVerilog is based on class level testbench which is more dynamic and reusable in nature.
  - Verification:
    - Coverage driven verification, constrained random verification, and assertion-based verification.
    - VMM and UVM are the main verification methodologies widely used in industry today. UVM stands for Universal Verification Methodology and VMM stands for Verification Methodology Manual.

Outline

- 1.1 History and Road Map of IC Industry
- 1.2 Introduction to IC Industry
- 1.3 FPGA and ASIC Design
- 1.4 Fundamentals of Verilog HDL Design
- 1.5 Attributes of Verilog HDL Design

1.3.1 FPGA and ASIC Implementations

- A. FPGA Slice
  - CLB (Configurable Logic Block) contains a pair of slices.
  - Each slice contains LUTs and storage elements used to provide logic, arithmetic, and ROM functions
  - A typical slice of the Xilinx 7 Series FPGAs contains 4 LUTs and 8 storage elements

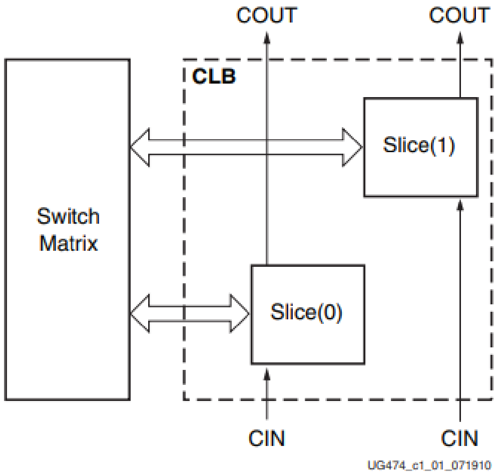
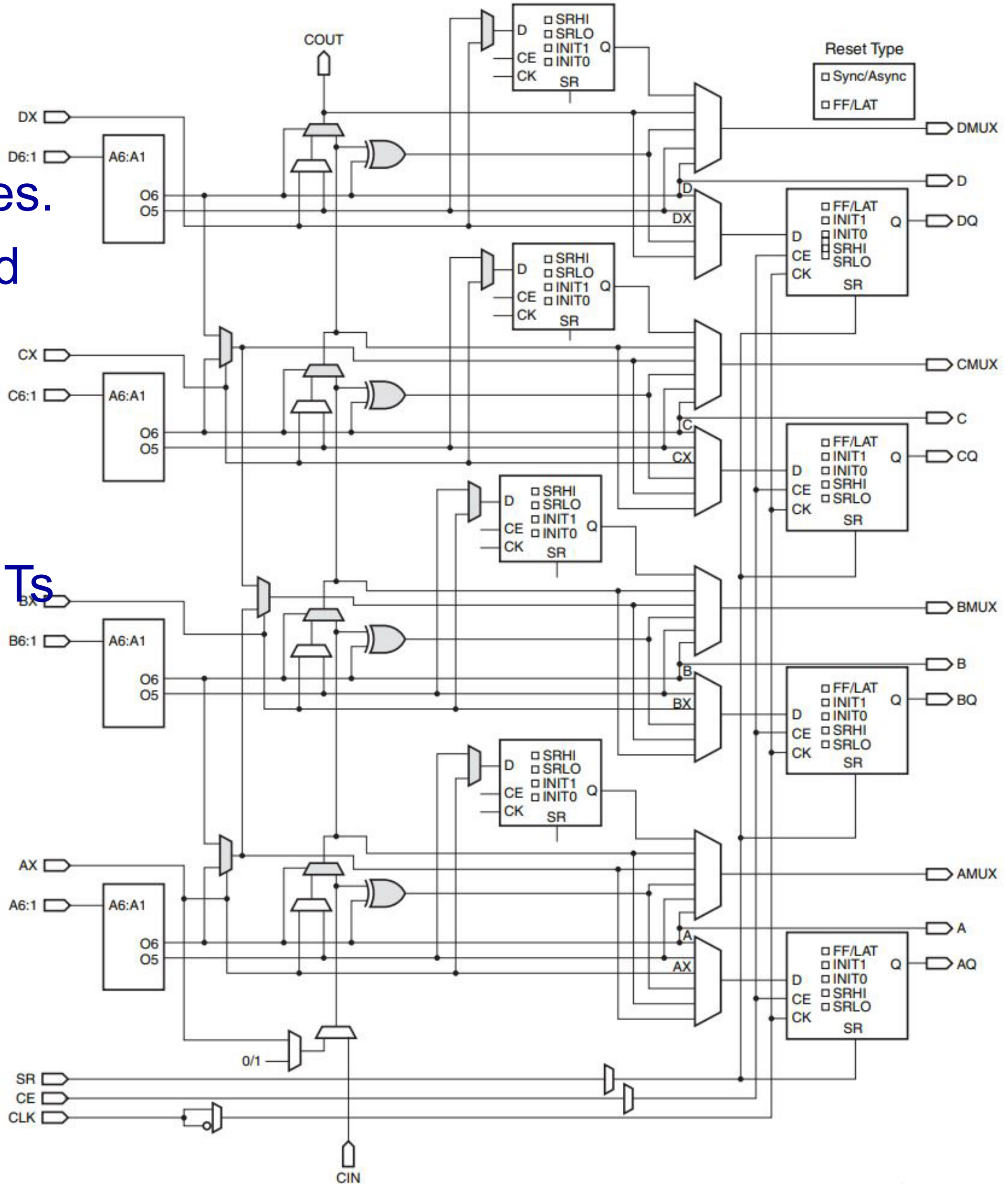


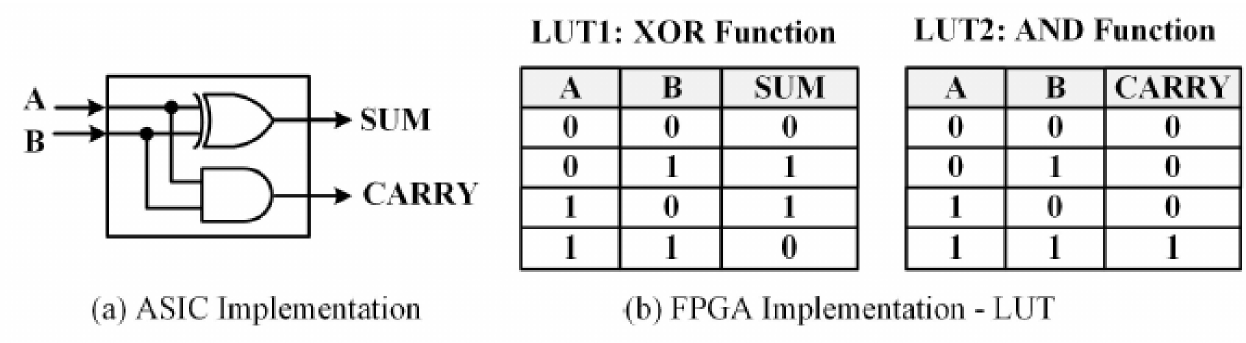
Diagram of CLB and the Slice of Xilinx 7 Series FPGAs



1.3.1 FPGA and ASIC Implementations

- B. Combinational Circuit on ASIC/FPGA

```verilog
//Verilog Description of Single-Bit Half Adder
module Comb_Circuit (input  A, B,
                     output SUM, CARRY);
assign SUM   = A ^ B ;
assign CARRY = A & B ;
endmodule
```



LUT1: XOR Function

A	B	SUM
0	0	0
0	1	1
1	0	1
1	1	0

LUT2: AND Function

A	B	CARRY
0	0	0
0	1	0
1	0	0
1	1	1

(a) ASIC Implementation

(b) FPGA Implementation - LUT

FIGURE 1.1
Half Adder Implementation with ASIC and FPGA

1.3.1 FPGA and ASIC Implementations

- C. Sequential Circuit on ASIC/FPGA
  – ASIC Synthesis Circuit: 2 logic gates of XOR and AND, and a 2-bit D register to store the output of the two gates.
  – FPGA Synthesis Circuit: 2 LUTs and 2 D flip-flops

```
//Verilog Description of Sequential Circuit
//with a Single-Bit Half Adder and a Register
module Seq_Circuit (input                A, B,
                    input                CLK ,
                    output reg [1:0]     Q   );
wire SUM   = A ^ B ;
wire CARRY = A & B ;

always @(posedge CLK) begin
  Q[1] <= CARRY  ;
  Q[0] <= SUM    ;
end
endmodule
```

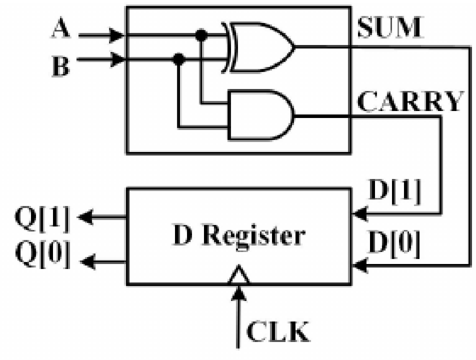
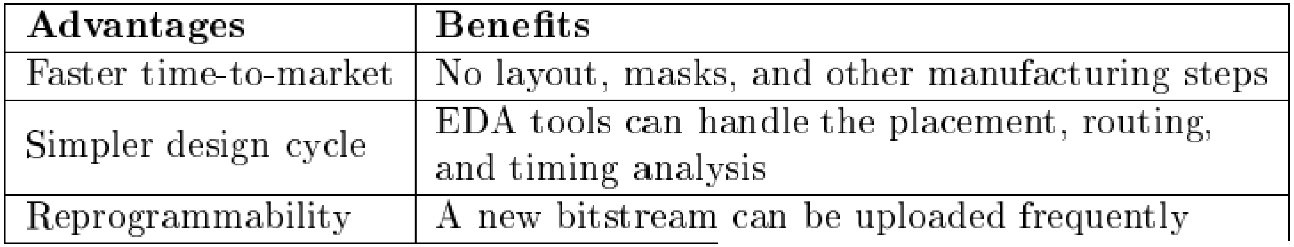


FIGURE 1.2
Half Adder and Register Implementation with ASIC

1.3.2 Comparison of FPGA and ASIC

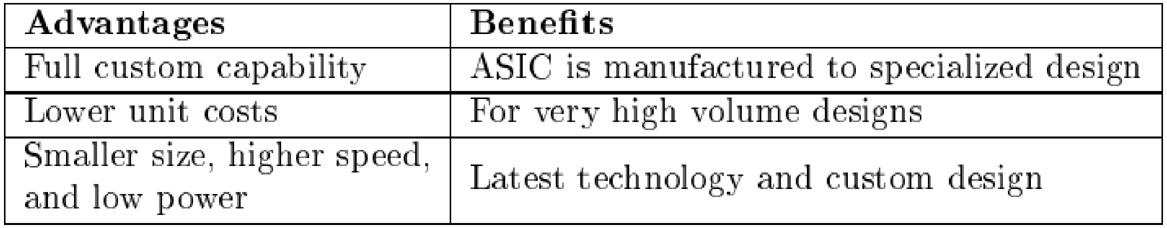
- Advantages with the FPGA
  - Faster to market with simpler design cycle
  - Field reprogramability
- Advantages with the ASIC
  - Resource cost (size, power, etc.) and unit cost are low
  - High performance
  - Downsides: time-consuming design cycle; high cost, high complexity, and big risk for taping out an ASIC

TABLE 1.2
Advantages of FPGA Implementation



Advantages	**Benefits**
Faster time-to-market	No layout, masks, and other manufacturing steps
Simpler design cycle	EDA tools can handle the placement, routing, and timing analysis
Reprogrammability	A new bitstream can be uploaded frequently

TABLE 1.3
Advantages of ASIC Implementation



Advantages	**Benefits**
Full custom capability	ASIC is manufactured to specialized design
Lower unit costs	For very high volume designs
Smaller size, higher speed, and low power	Latest technology and custom design

1.3.3 ASIC Development: MPW vs. Fullset

- A. MPW vs. Fullset
  - The ASIC design process is renowned for its extensive development cycle and the considerable expenses tied to chip manufacturing. Hence, it's commonly advised to initiate with an **MPW** iteration of ASIC fabrication before transitioning to a **Fullset**.
    - All semiconductor foundries, including notable ones like TSMC and Globalfoundries, extend the MPW service. This approach, marked by its cost efficiency, permits multiple IC designs from diverse clients to coexist on a single wafer for production.
    - Conversely, a Fullset constitutes the comprehensive collection of design files and specifications indispensable for the production of a single IC. Once a design has undergone testing and validation via an MPW fabrication, a Fullset becomes the go-to choice for extensive commercial manufacturing.

1.3.3 ASIC Development: MPW vs. Fullset

- B. ASIC Fabrication Cost:

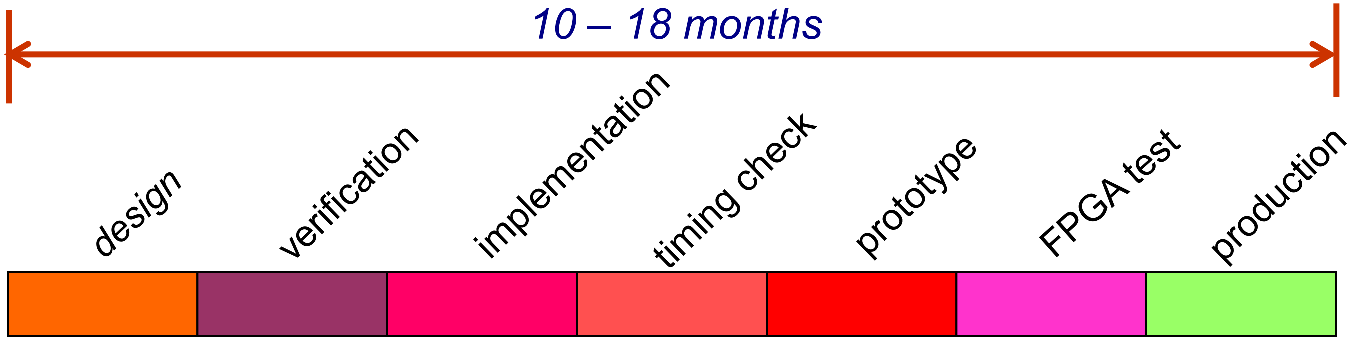
  Project: PLM3K HDTV System-on-Chip in 2008

  Year: 2008

  Specification: 10 Million Transistors, 90nm TSMC technology

  - MPW: $150,000
  - Full-Set: $3 M

- Design Cycle: 14 months
- Head count (70)
  - Designer (10)
  - Verification (30)
  - Backend (10)
  - HW/SW integration (20)



10 – 18 months

design

verification

implementation

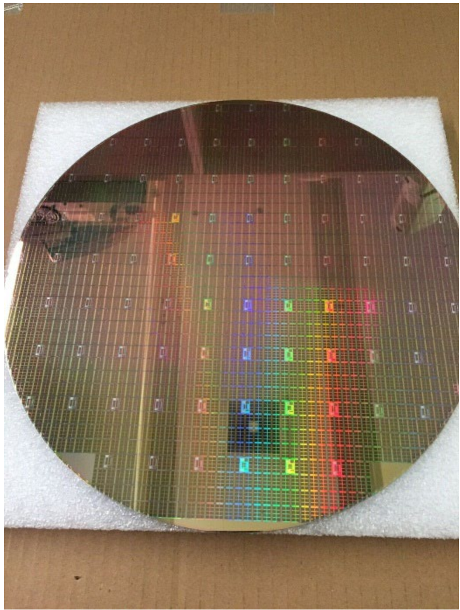
timing check

prototype

FPGA test

production

Silicon wafer 12” 300mm copper pattern reclaim



1.3.4 FPGA Development and Applications

- FPGA development serves two primary applications in the industry.
  - Firstly, FPGA verification and prototyping play a crucial role *within the ASIC design flow*, facilitating risk mitigation associated with ASIC fabrication and contributing to time and cost reduction.
  - Secondly, FPGA designs find popularity in various specific applications where *production volume is low*. These applications span across aerospace and defense, automotive, data centers, medical devices, video and image processing, telecommunication, data communication, high-performance computing, industrial sectors, and more.

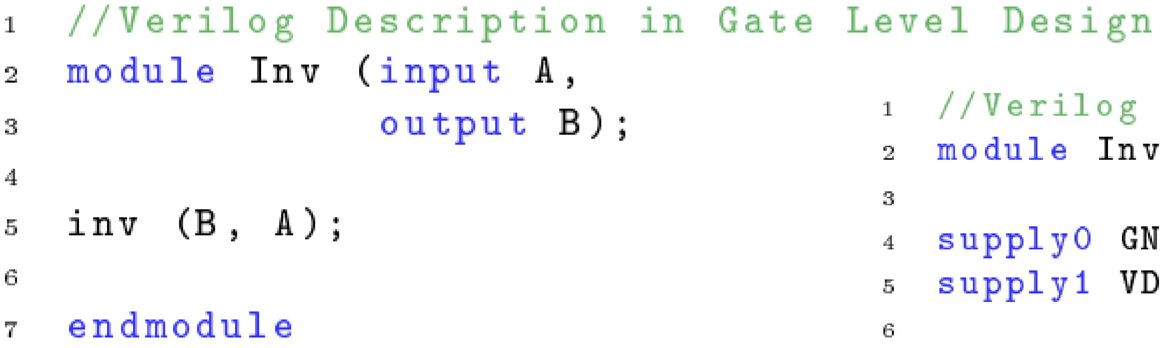
  - Recent advancements in FPGA development have witnessed the rise of *software-hardware co-design* platforms integrating FPGAs and CPUs, such as the Xilinx Zynq FPGA and Intel HARPv2. These platforms serve as demonstrations and accelerators for specific algorithm processing tasks, catering to evolving computing demands.

Outline

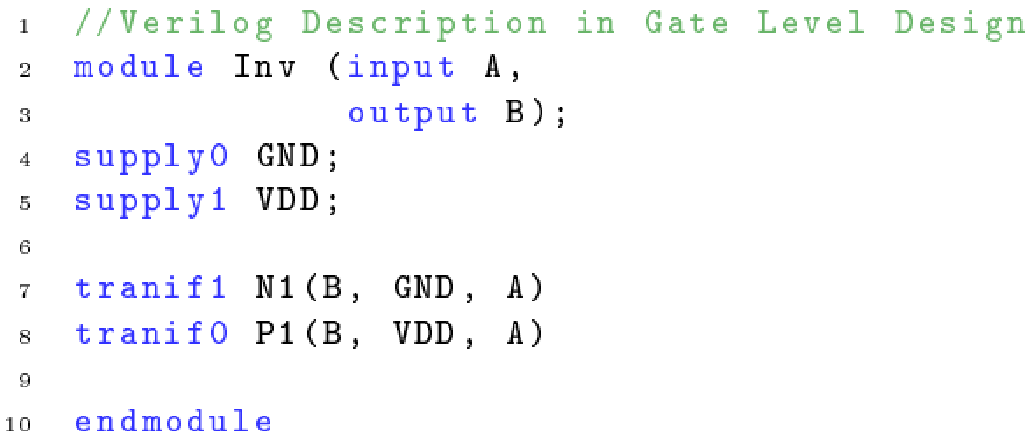
- 1.1 History and Road Map of IC Industry
- 1.2 Introduction to IC Industry
- 1.3 FPGA and ASIC Design
- 1.4 Fundamentals of Verilog HDL Design
- 1.5 Attributes of Verilog HDL Design

1.4.1 Multi-Level Description with Verilog HDL

- A. Multi-Level Designs
  - Specification Level
    - Includes the definitions of IOs, timings, functional registers, and specific design block diagrams
  - RT Level
    - RT-level description is in higher abstract level which can be synthesized into gate-level netlist with synthesis tools like Synopsys Design Compiler and FPGA tool AMD/Xilinx Vivado
  - Gate Level



```
//Verilog Description in Gate Level Design
module Inv (input A,
            output B);

inv (B, A);

endmodule
```

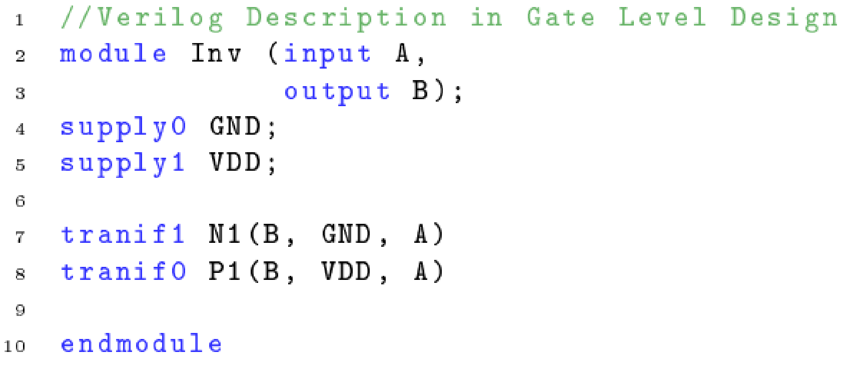


```
//Verilog Description in Gate Level Design
module Inv (input A,
            output B);
supply0 GND;
supply1 VDD;

tranif1 N1(B, GND, A)
tranif0 P1(B, VDD, A)

endmodule
```

  - Final Physical Level

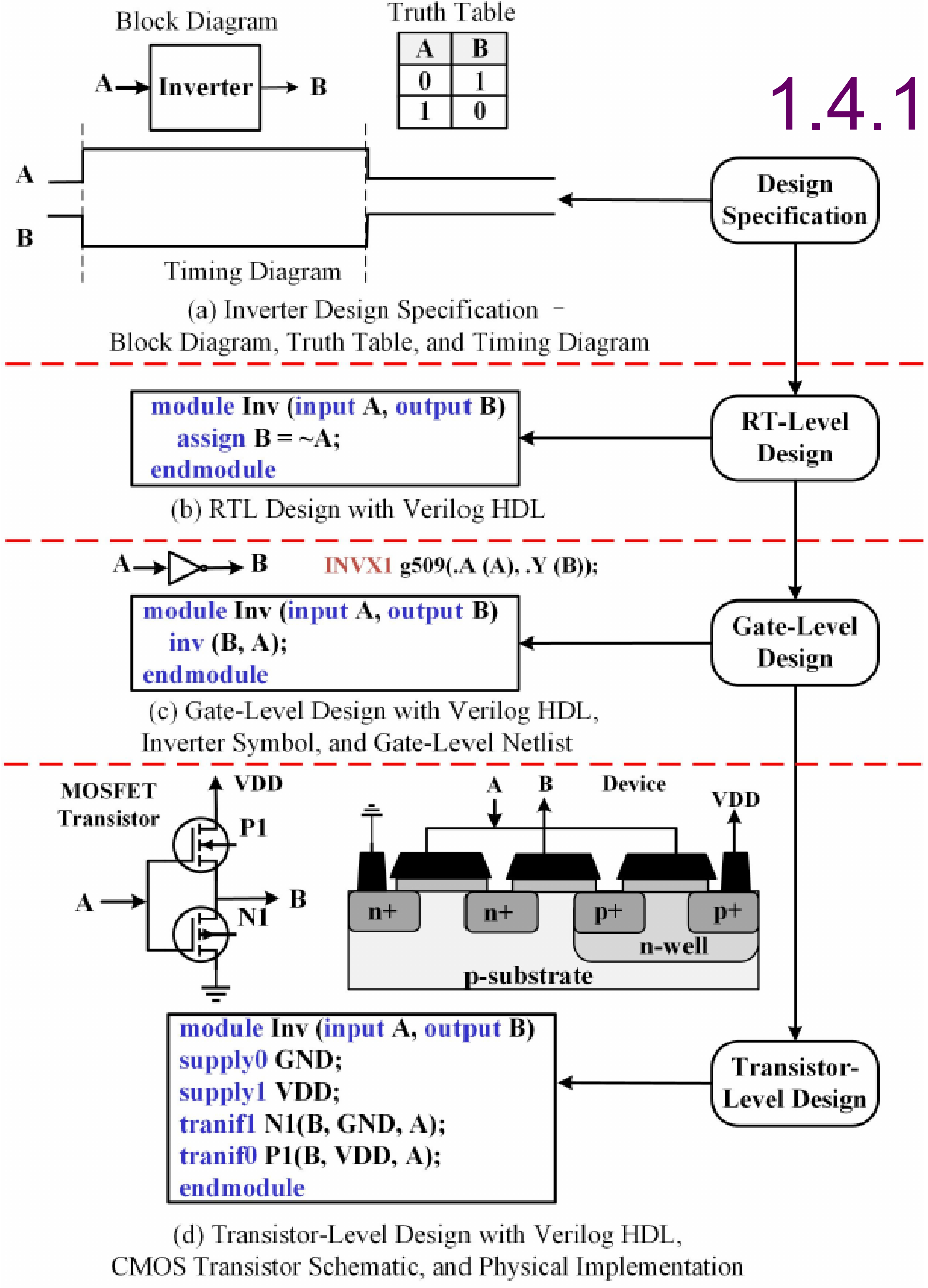
1.4.1 Multi-level HDL Description

- CMOS, PMOS and NMOS Transistors
  - CMOS technology incorporates both P-type (PMOS) and N-type (NMOS) metal-oxide-semiconductor field-effect transistors (MOSFETs) on a single chip, enabling the creation of complementary logic circuits.
    - PMOS transistor: when the gate terminal is binary zero, it creates a conducting channel between the source and the drain (channel ON status); conversely, when the gate terminal is binary one, the channel remains open (channel OFF status).
    - NMOS transistor: when the gate terminal is binary one, it establishes a conducting channel between the drain and the source (channel ON status); conversely, when the gate terminal is binary zero, the channel remains open (channel OFF status).



```
//Verilog Description in Gate Level Design
module Inv (input A,
            output B);
supply0 GND;
supply1 VDD;

tranif1 N1(B, GND, A)
tranif0 P1(B, VDD, A)

endmodule
```

1.4.1 Multi-level HDL Description



A	B
0	1
1	0

(a) Inverter Design Specification – Block Diagram, Truth Table, and Timing Diagram

```
module Inv (input A, output B)
   assign B = ~A;
endmodule
```

(b) RTL Design with Verilog HDL

```
INVX1 g509(.A (A), .Y (B));
```

```
module Inv (input A, output B)
   inv (B, A);
endmodule
```

(c) Gate-Level Design with Verilog HDL, Inverter Symbol, and Gate-Level Netlist

```
module Inv (input A, output B)
supply0 GND;
supply1 VDD;
tranif1 N1(B, GND, A);
tranif0 P1(B, VDD, A);
endmodule
```

(d) Transistor-Level Design with Verilog HDL, CMOS Transistor Schematic, and Physical Implementation

FIGURE 1.3
Multi-Level Verilog HDL Descriptions

- Design Specification of a Single-bit Inverter
  - Including the block diagram showing the IOs and the design block, the truth table, and the timing diagram.
- Verilog HDL in RT-level
- Synthesis Netlist
  - Logic cells
- CMOS Schematic and Fabrication
  - PMOS and NMOS

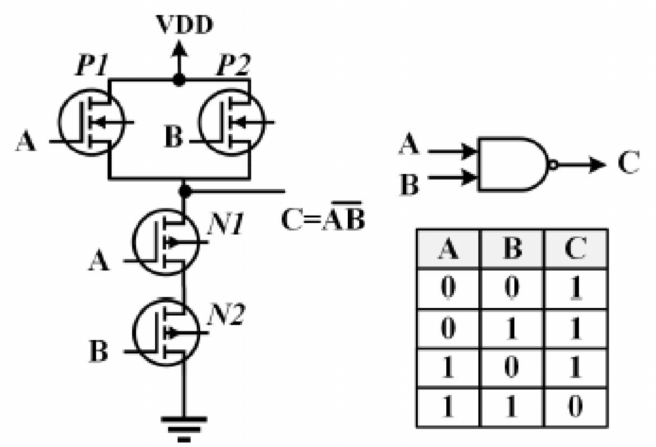
1.4.2 Fundamentals of Combinational Circuits

- Combinational circuit:
  - Its outputs only depend on the current inputs
  - E.g.: Inverter, NAND, NOR, Tri-state buffer, Multiplexer, etc.
- Sequential circuit
  - Its outputs depend on not only the current inputs but also the status of the previous inputs
  - Latches and Flip-flops and Registers
    - Both the latch and flip-flop contain a clock/enable input EN/CLK, a data input D, and an output Q
      - A latch is a level-trigger device
      - A flip-flop is usually a rising edge-trigger component
    - In this lecture, a register is multi-bit flip-flops

1.4.2 Fundamentals of Combinational Logic

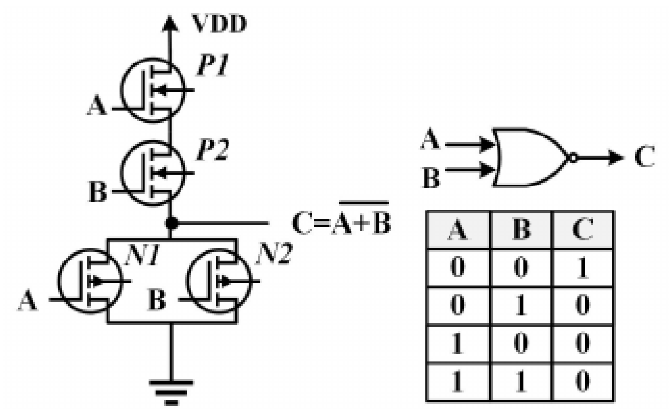
- Construction of a NAND Gate
  - PMOS: source, gate, drain
  - NMOS: drive, gate, source

- Construction of a NOR Gate
  - PMOS: source, gate, drain
  - NMOS: drive, gate, source



A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

FIGURE 1.4
NAND Gate Implementation at the Transistor Level

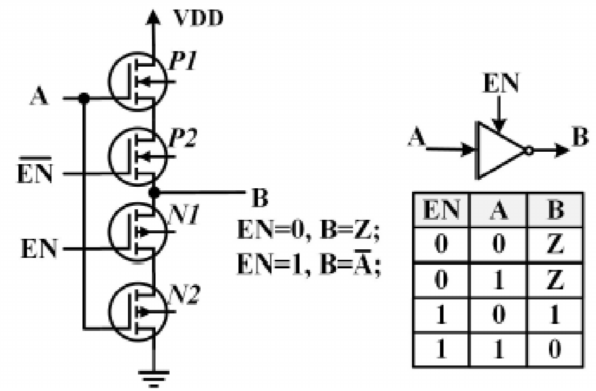


A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

FIGURE 1.5
NOR Gate Implementation at the Transistor Level

1.4.2 Fundamentals of Combinational Logic

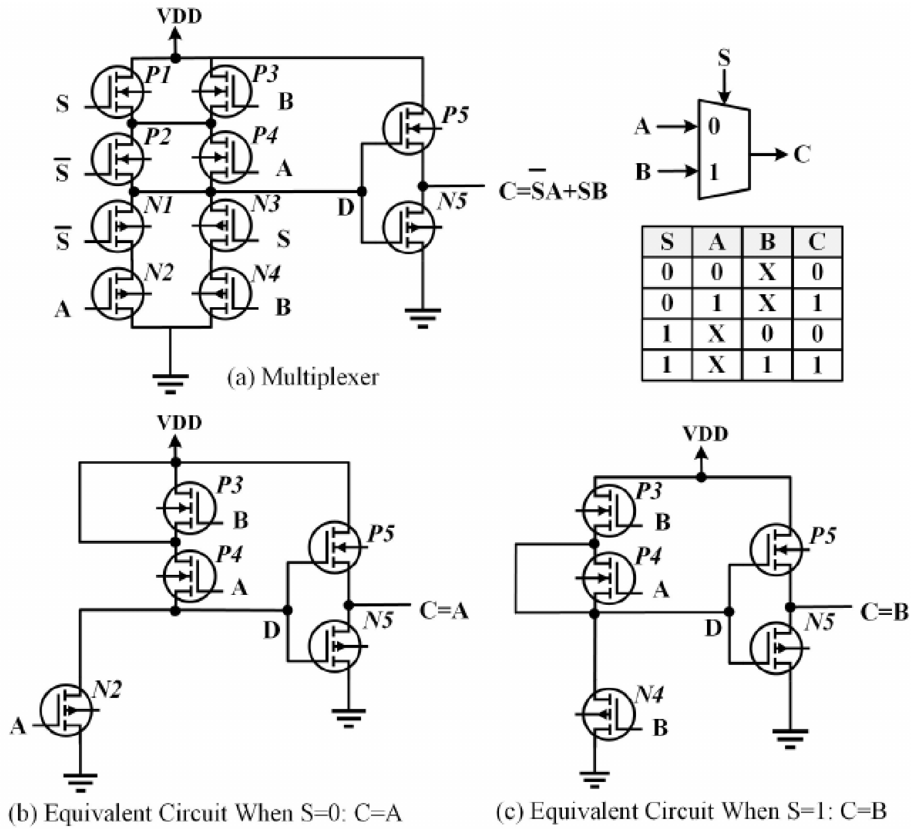
- Construction of a Tri-state Inverter



EN	A	B
0	0	Z
0	1	Z
1	0	1
1	1	0

FIGURE 1.6
Tri-State Inverter Implementation at the Transistor Level

- Construction of a 2-to-1 Multiplexer



S	A	B	C
0	0	X	0
0	1	X	1
1	X	0	0
1	X	1	1

(a) Multiplexer

(b) Equivalent Circuit When S=0: C=A

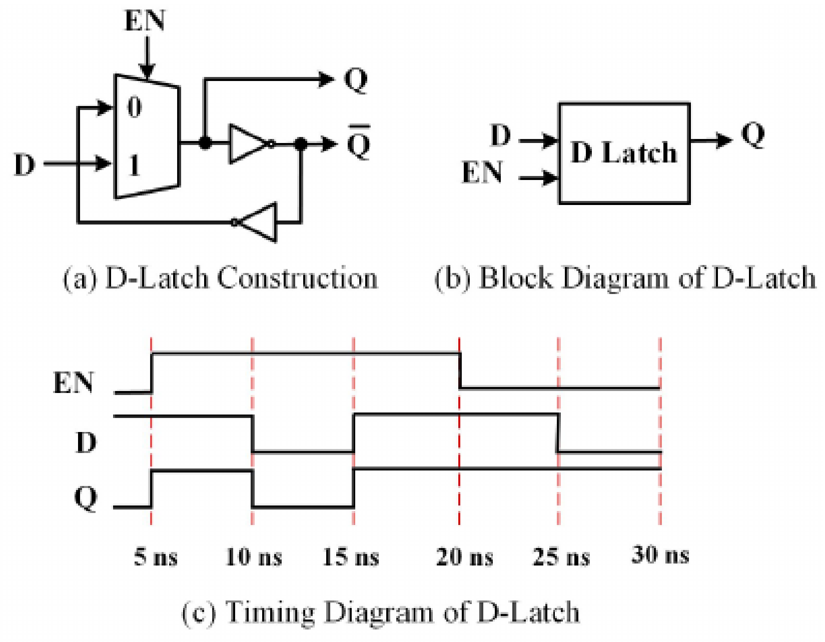
(c) Equivalent Circuit When S=1: C=B

FIGURE 1.7
2-to-1 Multiplexer Implementation at the Transistor Level

1.4.3 Fundamentals of Sequential Logic

- Construction of a D Latch
  - Level sensitive

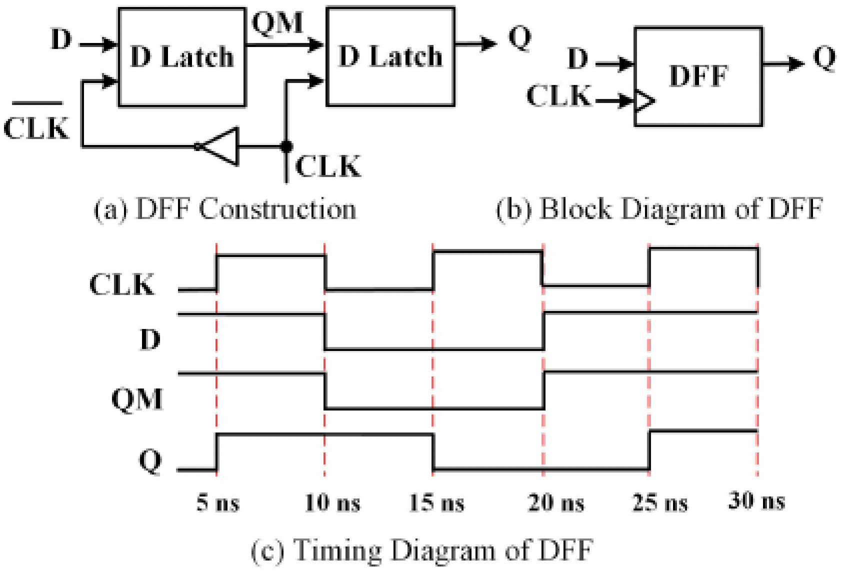
- Construction of a D Flip-Flop
  - Clock edge sensitive
  - Multi-bit flip-flop is a register



(a) D-Latch Construction (b) Block Diagram of D-Latch

(c) Timing Diagram of D-Latch

FIGURE 1.8
D-Latch Implementation



(a) DFF Construction (b) Block Diagram of DFF

(c) Timing Diagram of DFF

FIGURE 1.9
DFF Implementation

Outline

- 1.1 History and Road Map of IC Industry
- 1.2 Introduction to IC Industry
- 1.3 FPGA and ASIC Design
- 1.4 Fundamentals of Verilog HDL Design
- 1.5 Attributes of Verilog HDL Design

1.5.1 Key Attributes of Hardware Description

- A. Hardware Construction: Block Diagram
  - Software coding
    - Software programming involves executing specific tasks and operations on hardware but is not directly part of the hardware.
  - Hardware design
    - **Linkage between the code and the specific hardware:**
      - Will be implemented in actual electronic circuits
      - Directly impacts hardware performance and resource utilization.
    - *Block diagram*
      - Prior to HDL coding, designers should outline a clear *block diagram* to visualize the hardware components and their interconnection.
      - It serves as a blueprint for the hardware design and ensures that the Verilog code accurately represents the intended hardware behavior, data paths, and interfacing.

1.5.1 Key Attributes of Hardware Description

- B. Hardware Behaviors: Timing Diagram
  - Timing controls play a vital role in synchronizing and coordinating the behavior of hardware components within the entire system.
    - Clock-cycle-related counters and timers
    - Finite state machines to transition operations between different states
    - Bus protocols:
      - Ready-valid bus protocols for data transactions
      - Enable-finish handshaking for data processing
      - Request-grant mechanisms for bus arbitration
  - Proper timing management is essential in digital systems to prevent issues
    - Address/data collision
    - Race conditions
    - Timing violations

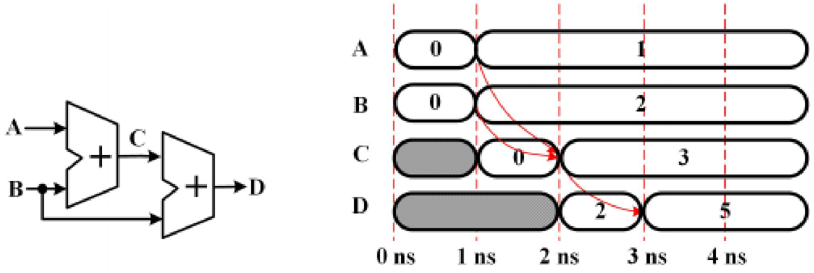
1.5.2 Concurrency

- C

```
int A = 1;
int B = 2;
//Add A and B and assign the result to C
int C = A + B;
//Add the updated C and B and assign the result to D
int D = C + B;
```

- Verilog HDL

```
module Binary_Adder (input  [31:0] A, B,
                     output [31:0] C, D);
// 32-bit binary adder with inputs A and B, output C
assign C = A + B;
// 32-bit binary adder with inputs C and B, output D
assign D = C + B;
endmodule
```



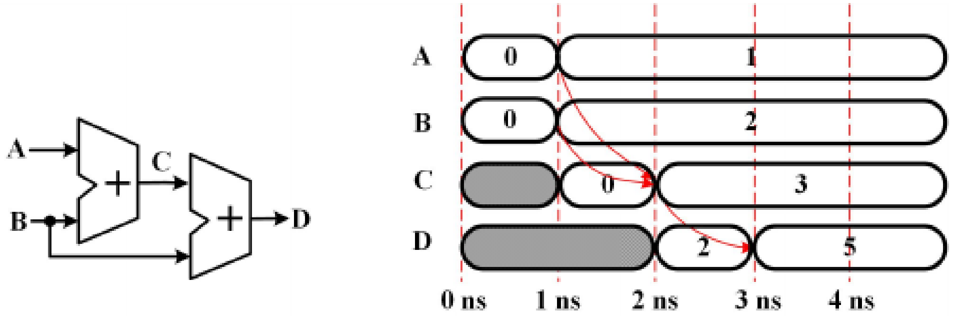
(a) Block Diagram of Two Adders

(b) Timing Diagram of Two Adders

FIGURE 1.10
An Example of Hardware Concurrency: Two Cascading Adders

1.5.2 Concurrency

- C
- Verilog HDL
  - Initial A=0, B=0. Assume that the adders delay is 1 ns. Both C and D are unknown.
  - After 1 ns, A=1 and B=2; C=0 based on A=1 and B=0.
  - At 2 ns, C=3 based on A=1 and B=2. D=2 based on B=2 and C=0.
  - At 3 ns. D=5 based on B=2 and C=3.



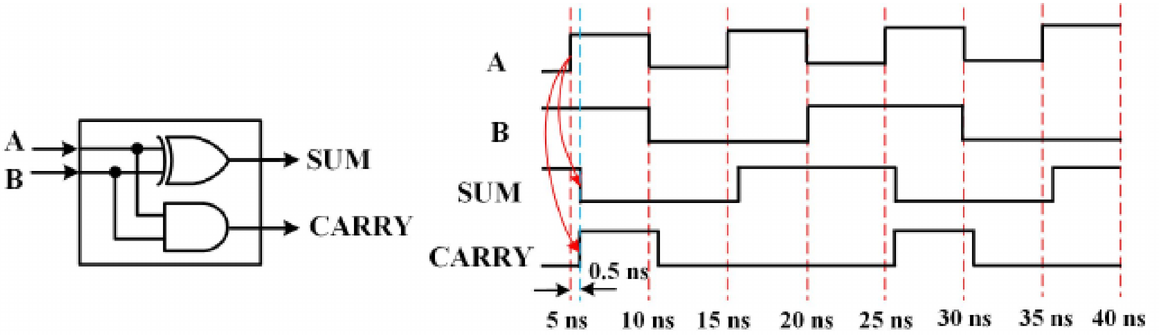
(a) Block Diagram of Two Adders

(b) Timing Diagram of Two Adders

FIGURE 1.10
An Example of Hardware Concurrency: Two Cascading Adders

1.5.3 Timing Diagram

- Timing Diagram
  - A valuable tool for analyzing signal transitions over time, particularly in sequential circuits experiencing clock cycle delays, is the timing diagram.
- A. Data Transitions within Combinational Circuit
  - Assume that gate delays are 0.5ns
  - 5ns, A from 0 to 1. Then 5.5ns, SUM 1 to 0 CARRY 0 to 1
  - 10ns, A 1 to 0, B 1 to 0. Then 10.5 ns, CARRY 1 to 0

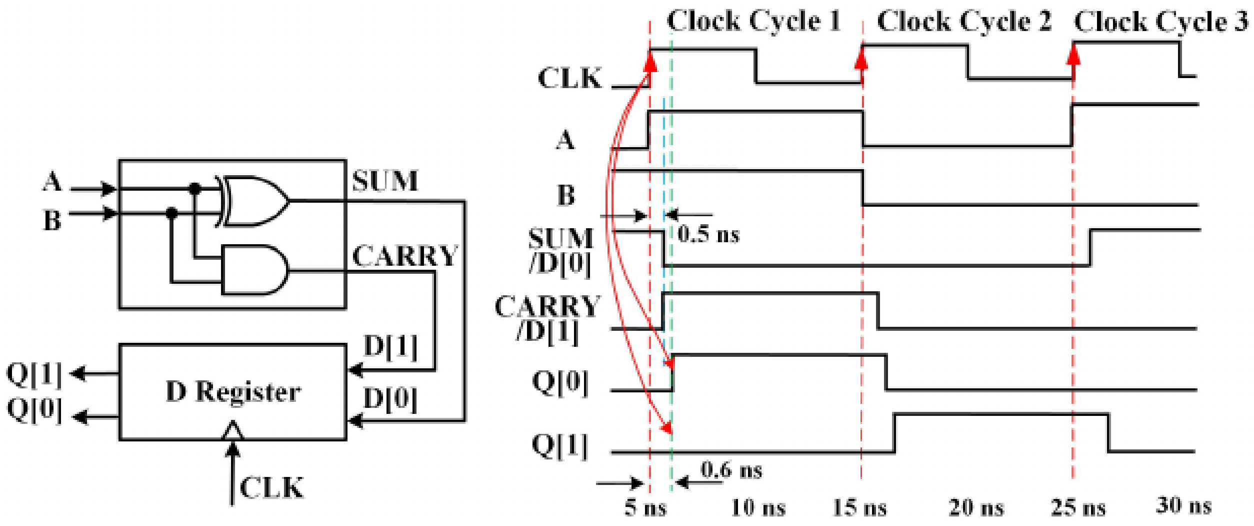


(a) Block Diagram of Single-Bit Half Adder (b) Timing Diagram of Single-Bit Half Adder

FIGURE 1.11
Timing Diagram of Combinational Circuit: Single-Bit Half Adder

1.5.3 Timing Diagram

- B. Data Transitions within Sequential Circuit:
  - Assume clk-q: 0.6ns, gate delay: 0.5ns
  - First rising edge of the clk (5ns), Q[0] 0 to 1 after 0.6ns
  - Second rising edge of the clock (15ns), Q[0] 1 to 0 after 0.6ns
  - Third rising edge of the clock (25ns), Q[0] sustains 0 after 0.6ns



(a) Block Diagram of Sequential Circuit - Single-Bit Half Adder and Register

(b) Timing Diagram of Sequential Circuit - Single-Bit Half Adder and Register

FIGURE 1.12
Timing Diagram of Sequential Circuit: Single-Bit Half Adder and Register

1.5.3 Timing Diagram

- C. Register-Transfer Level Design:
  - Represents data flows and computational logic between registers.
  - “register-in and register-out" design pattern:
    - Combinational logic is responsible for data processing and computation
    - Registers serve as memory elements storing data within the sequential circuit and synchronize the entire design system over clock cycles.

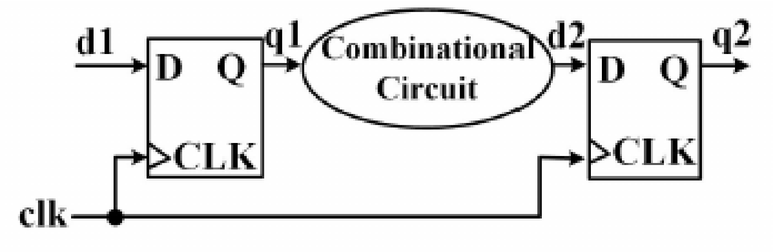


FIGURE 1.13

Register-Transfer Level Design with Verilog HDL

1.5.4 Signal Value

- Verilog HDL supports four values: 0, 1, X, and Z
- High Z: shows a high impedance circuit, or an open circuit
  - High-Z is mainly used to the description of the tri-state buffer.
  - Tri-state buffer can be mainly employed to the design of the bi-direction IOs.

TABLE 1.4
Verilog Signal Values

Value	Interpretation
X	Unknown
0	0
1	1
Z	High Impedance

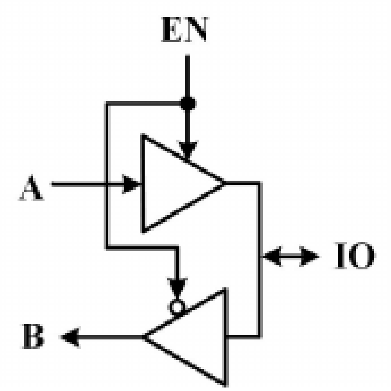


FIGURE 1.14
Bi-Directional IO

1.5.4 Signal Values

- X: Shows the status of unknown logic value, or could be either 0 or 1.
  - If the enables signals are all 0, the IO bus will drive the output O1, O2, until On;
  - If the enable signals are all 1, the I1, I2, until In circuits will multi-drive the IO bus (not allowed!);
  - The bus circuit is usually used to the IO distribution of the ASIC design.

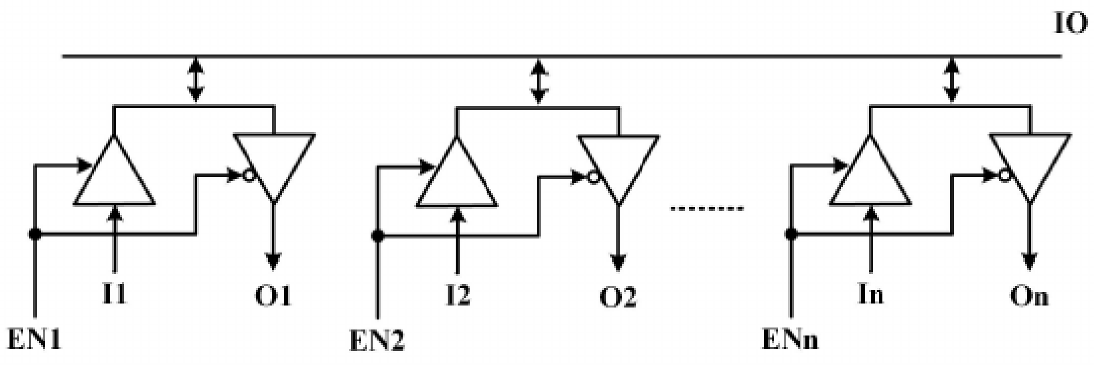
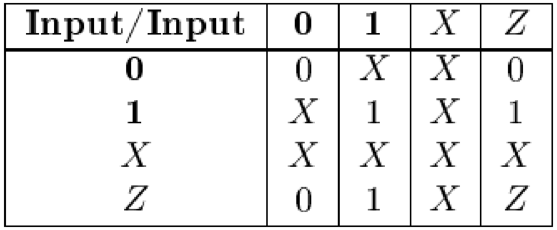


FIGURE 1.15
Multi-Drive Bus and Bi-Directional IO

1.5.4 Signal Values – Multi-drive Bus

- Design rules about X:
  - X is not allowed by ASIC Design
    - Though being supported by the Verilog Standard, it is not recommended when describing or designing a synthesizable ASIC.
    - The unknown status may create the risks when taping out the chip.
  - Specific value of all signals
    - All the signals and IOs of an ASIC must be specific values of either 0 or 1, or high-Z when describing tri-state buffers.
    - When designing a testbench, all the inputs of the design-under-test should be initialized either 0 or 1.
  - Multi-drive is not allowed
    - Although multi-drive status is not allowed by describing an ASIC, the results are supported by the Verilog Standard as shown in Table 1.5.

TABLE 1.5
Verilog Multi-Drive Results



Input/Input	**0**	**1**	X	Z
0	0	X	X	0
1	X	1	X	1
X	X	X	X	X
Z	0	1	X	Z