Basic DIGITAL Electronics

ELEC1303

Course Description

Instructors

- **DR Ali Kharrazu**
  - Lecturer
  - Ali.kharrazi@uwa.edu.au
  - Consultation Hours:

Timetable

Lectures

- Wednesday 2PM- 4:35AM Room: 08-0409
  (26th, April;10th,17th, 24th, 31th, May)
- Thursday 8AM-10:45PM Room: 08-0507
  (27th, April;4th, 11th, 18th, 25th, May;1st, June)
- Friday 2PM- 4:35AM Room: 27-0101
  (5th May)

Unit Description

- This unit introduces the fundamental concepts underpinning the design and analysis of modern digital systems. Topics covered include:
  - Design process abstractions and realisation technologies;
  - Introduction to hardware description language;
  - Boolean Algebra, combinational logic, minimisation, NAND/NOR implementations, multi-level gate networks, standard combinational components;
  - Sequential logic, latches, flip-flops, registers, counters, finite-state machines; analysis, synthesis and optimization of clocked sequential circuits; timing diagrams, hazards, maximum frequency of operation, metastability;
  - Microprocessor based systems, instruction sets, and assembly programming.

Learning outcomes

- Explain digital encoding and processing of information and discuss the associated benefits and limitations of digital design process abstraction.
- Simulate, analyse, optimize, and construct combinational and sequential logic circuits;
- Select the appropriate modern realization technologies and synthesis techniques for the implementation of a given digital system;
- Describe the architecture and operation of a microprocessor based system;
- Analyse, explain and write small assembly programs;
- Work effectively as part of a team and communicate results in a concise technical report.

Philosophy

- How much you will learn will depend on your motivations and interest.
- If you don't understand any point don't hesitate to ask and do not leave it until it's too late.
- Open door policy is provided by the instructor. Don't hesitate to ask questions and to consult with the instructor during the provided consultation hours

Assessment for this Unit

Class Tests	**30%**
Final Exam	**70%**

- 3 tests on weeks 34, 38, and 41. Please check timetable.
- To pass this unit, you have to achieve at least 50% overall score for this unit.
- Up to 5% bonus marks awarded for active participation in the Lectures. Attendance will also be recorded.

Supplementary Assessment

Supplementary Assessment

(1) Students may qualify for a supplementary assessment (including examination) on one of the following grounds:

(a) **Academic grounds** (by offer only) in which students must have:

i. a good academic standing overall;

ii. passed more than half the units in the teaching period, where relevant / appropriate;

iii. has no finding of academic integrity misconduct against them in the unit;

iv. submitted all assessment items for the unit; and

v. obtained a mark in the range of 45 – 49 in the unit overall and/or in a failed component within the unit; **OR**

(b) Concession grounds (by application by student) whereby it is the last remaining unit to graduate and has a mark in the range of 45-49 in the unit overall and/or in a failed component within the unit.

(2) All students must attempt the supplementary assessment on the scheduled date and no further supplementary assessment is available.

(3) Students undertaking a supplementary assessment will need to achieve a pass mark to be awarded a final mark and grade no higher than 50% and Pass respectively. It will be recorded as PS (Passed Supplementary) in the Academic Transcript and the 50% mark will contribute to the calculation of the WAM.

(4) Students who fail the supplementary assessment will receive the grade associated with their original mark, which will be N+.

(5) Units seeking to opt out from providing supplementary assessment must seek approval from the Pro Vice-Chancellor (Academic). Units approved exemption from providing supplementary assessment must be clearly identified as such in the Handbook and Unit Outlines.

(6) Where a student qualifies for a supplementary exam / assessment they will not be able to request for a review of the supplementary mark. Any failed assessment paper must be checked to ensure that there are no errors in marking process and procedures.

Textbook and references

- No textbook required for this unit

There are numerous publication for self stufy

- Lecture notes are sufficient. Many additional resources will be posted on LMS as required.
- If you want to read more about the topic, check the library for books on digital logic.

A Comprehensive Guide to Digital Electronics and Computer System Architecture *Mark Balch McGRAW-HILL*

Introduction

So, Why Study Digital Systems?

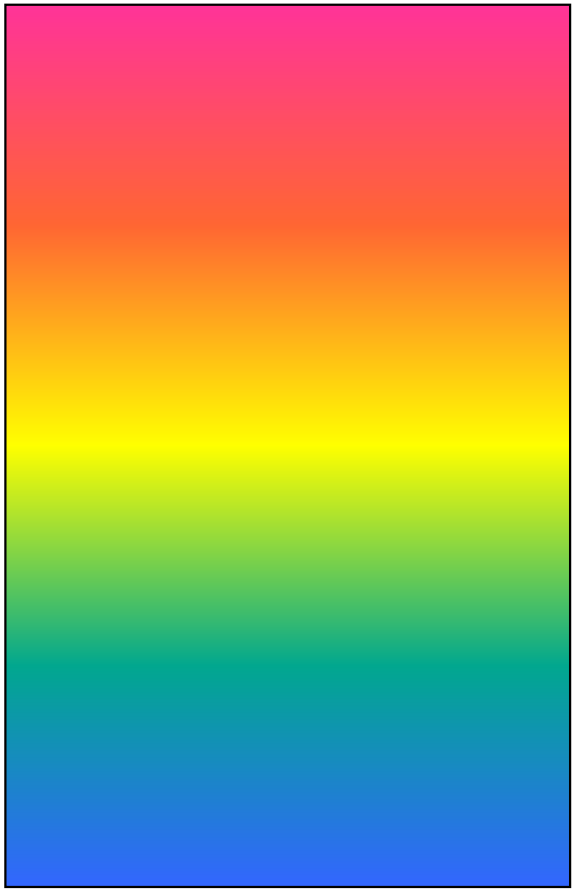
- We live in an ever increasingly "**Digital World**" with more and more electronic devices becoming digital:
  - Computers
  - Data communication
  - Cell Phones
  - Cameras, TVs, etc.

- It is essential to have a sound understanding of the fundamental principles and techniques underlying digital system design
- This unit acts as a foundation for more advanced work on digital systems engineering

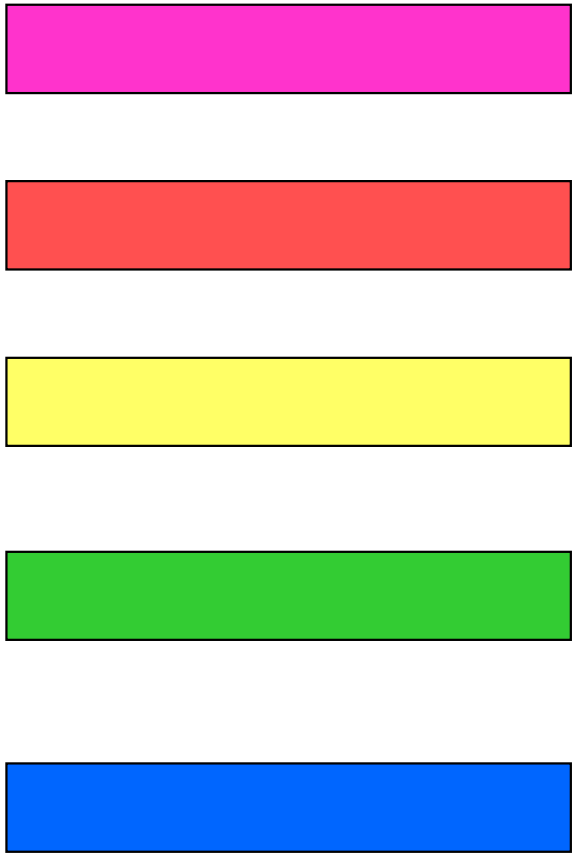
What is a digital system?

"A digital system is one that accepts as input digital information representing numbers, symbols, or physical quantities, processes this input information in some specific manner, and produces a digital output."

Analog versus Digital



Analog signal can take a **continuous range of values** with no discontinuity points



Digital signal can only take a **fixed set** of discrete values

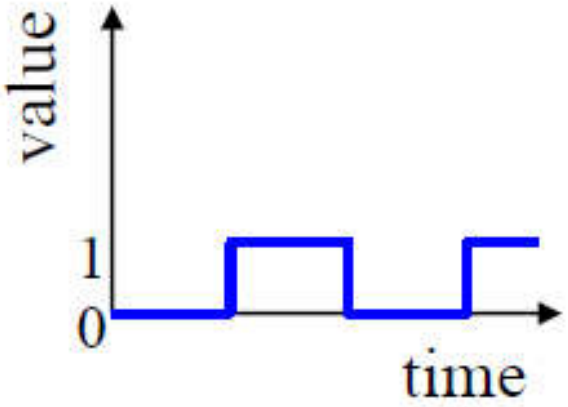
Digital versus analog systems

- In **digital** systems information is represented by signals (voltages, current or charge) which are assumed to take on values that correspond to one of a **finite number of discrete** information **values**.

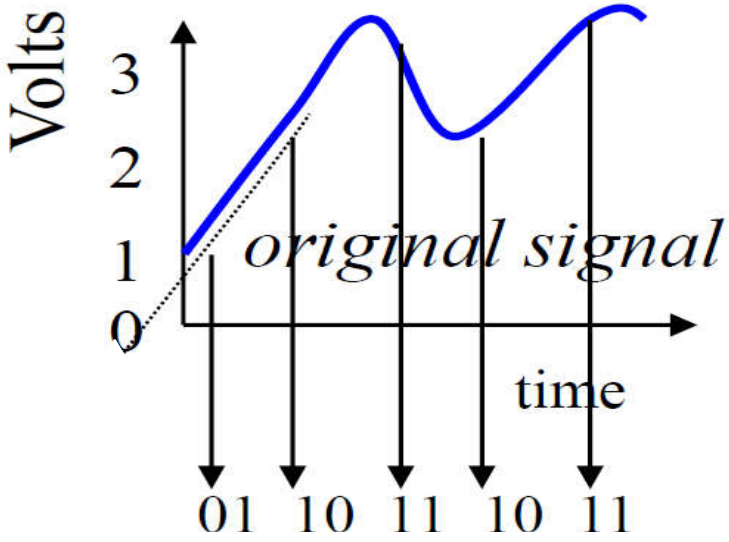
- In contrast to **analog** systems in which a **continuous range of signal values** correspond to a continuous range of information values (infinite number of possible values).

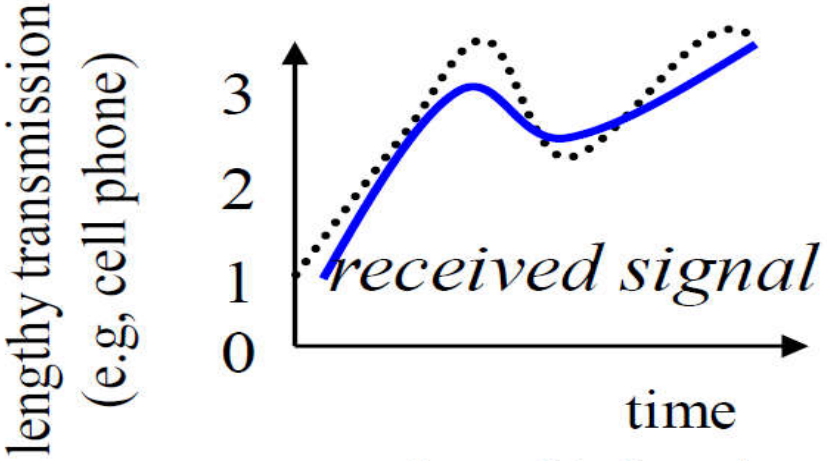
Digital Signals are Binary

- Each binary digit (called a bit) is either 1 or 0
- Bits have no inherent meaning, they can represent Unsigned/signed integers, Fractions, Characters, Images, sound, etc.
- Binary is popular because:
  - Transistors, the basic digital electric components, operate as switches using only two voltages (high/low)
  - Storing/transmitting one of two values is easier than three or more (e.g., loud beep or quiet beep, reflection or no reflection)

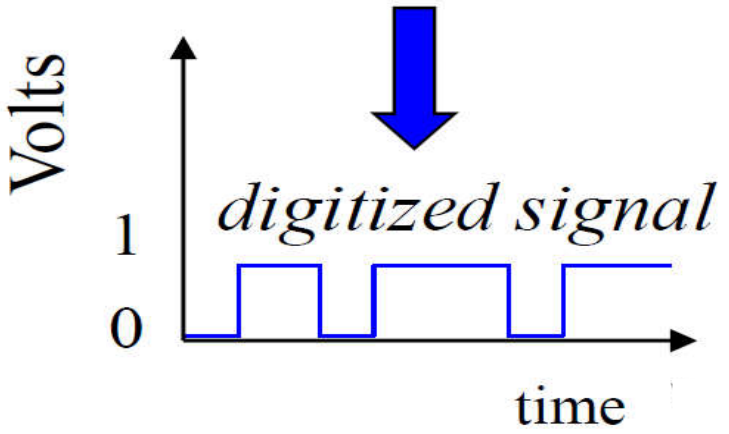


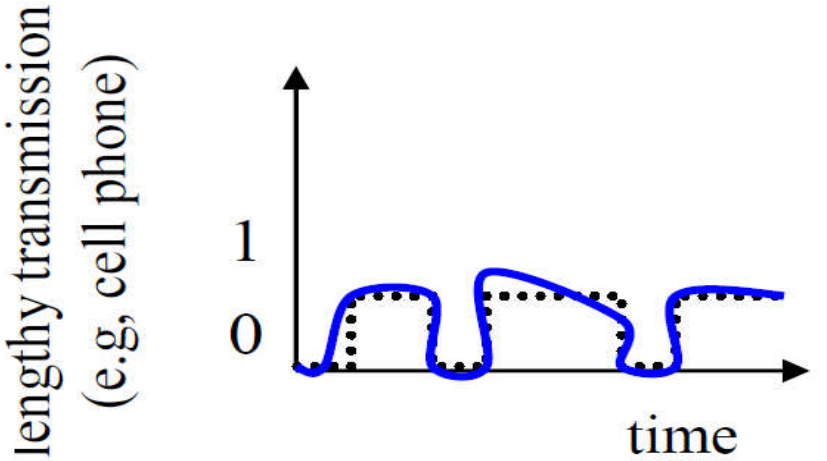
Better Noise immunity





How fix -- higher, lower, ?

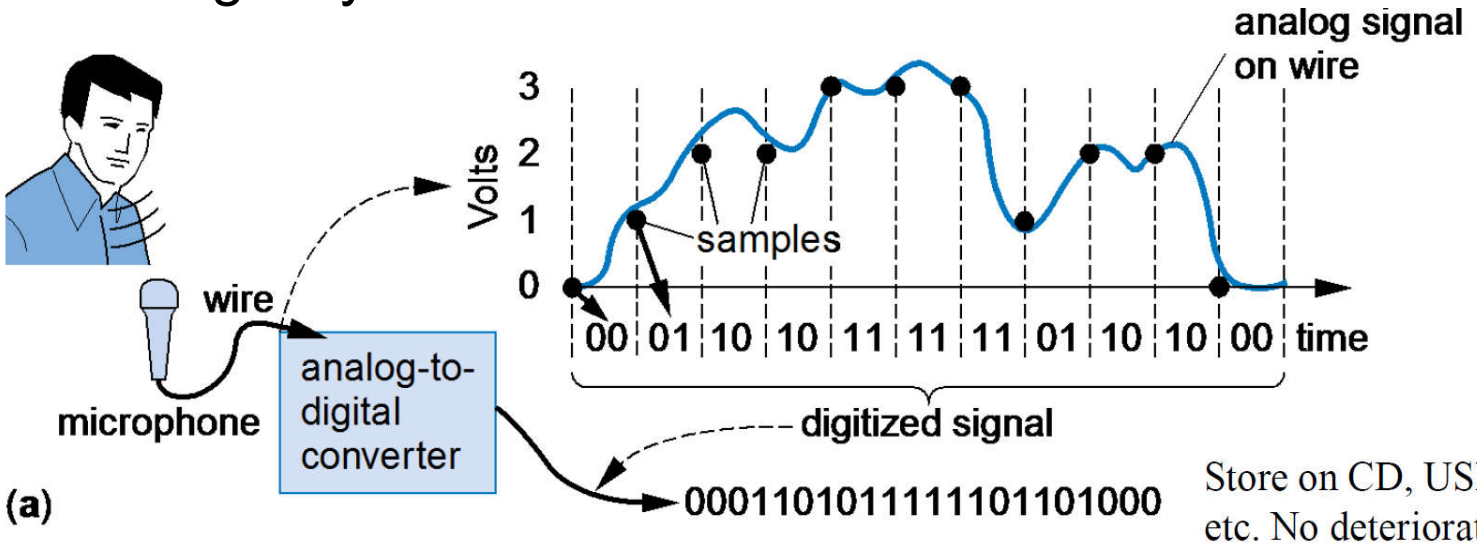




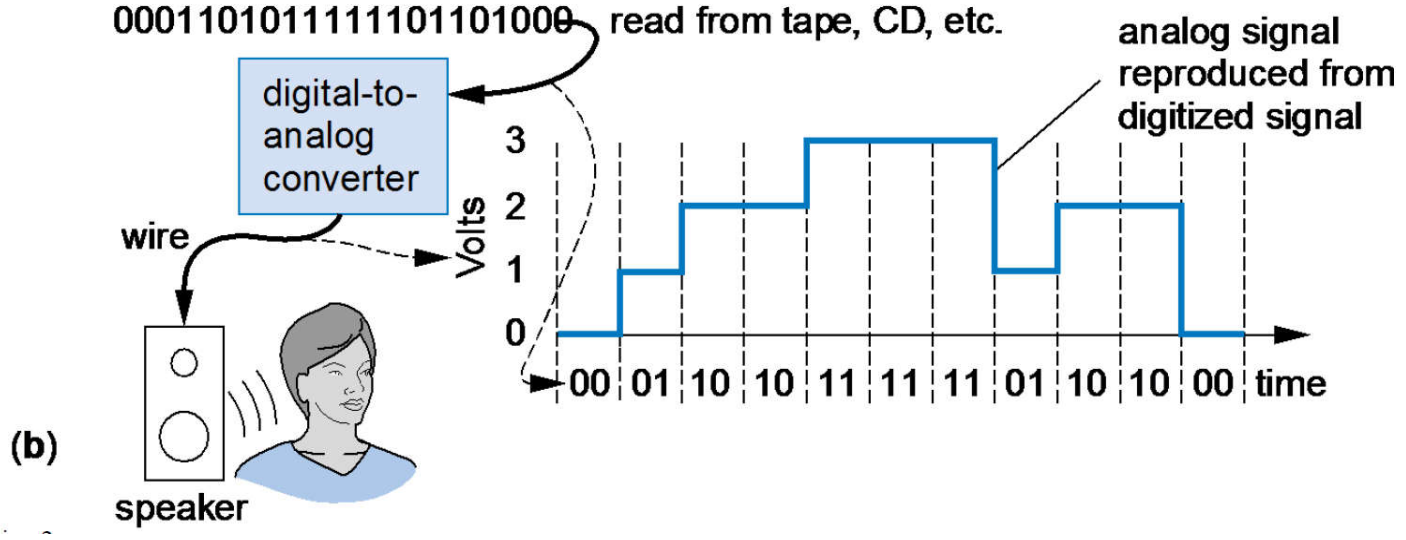
Can fix—distinguish 0s/1s, restore

Digitization of Analog Signals

- The "real" or "physical" world is analog but analog signals can be converted to digital by taking measurements or "samples" of the continuously varying signal at regular intervals. Digitized output coded using only 2 values: 0 & 1



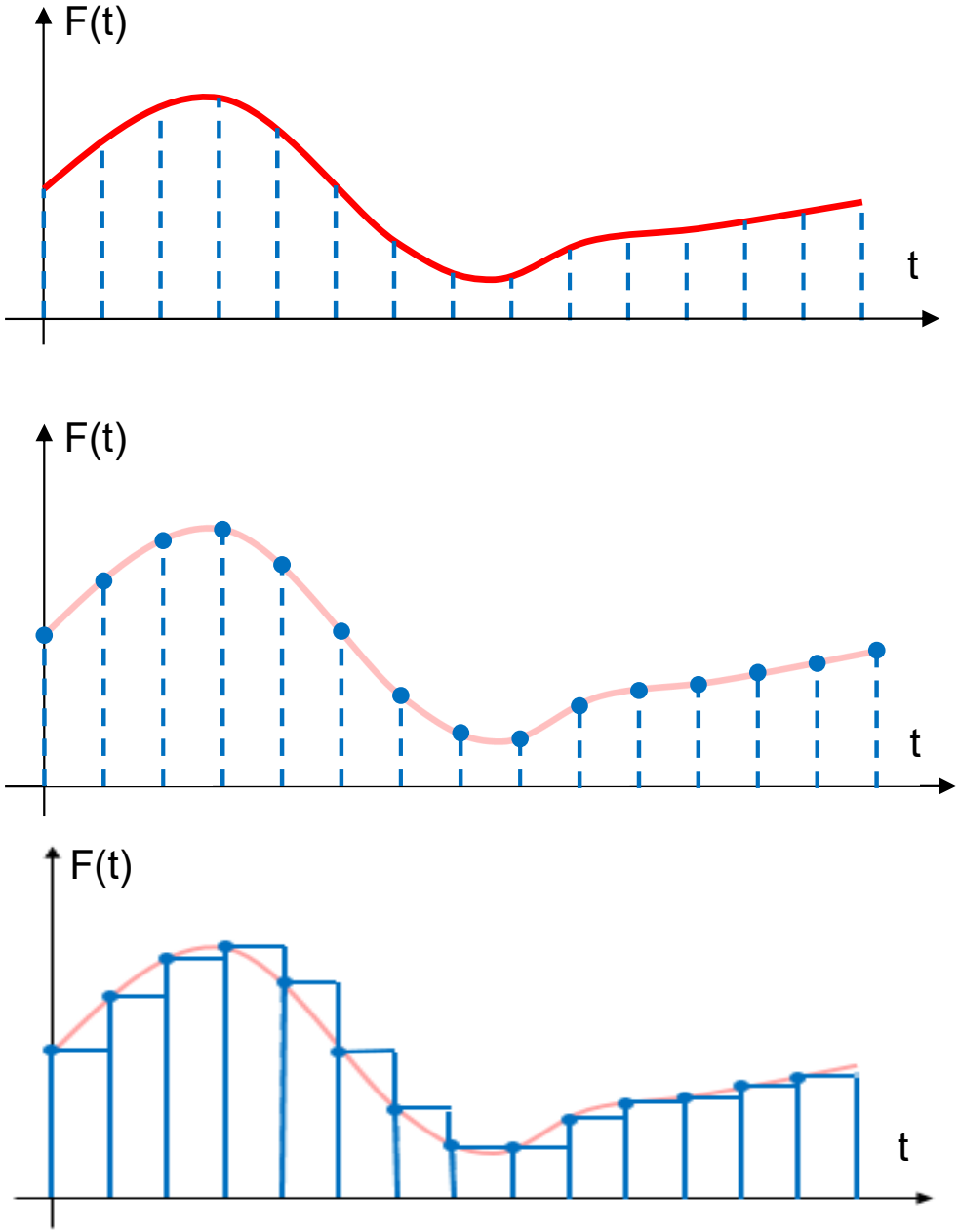
Store on CD, USB drive, etc. No deterioration.



Higher accuracy by adding more bits

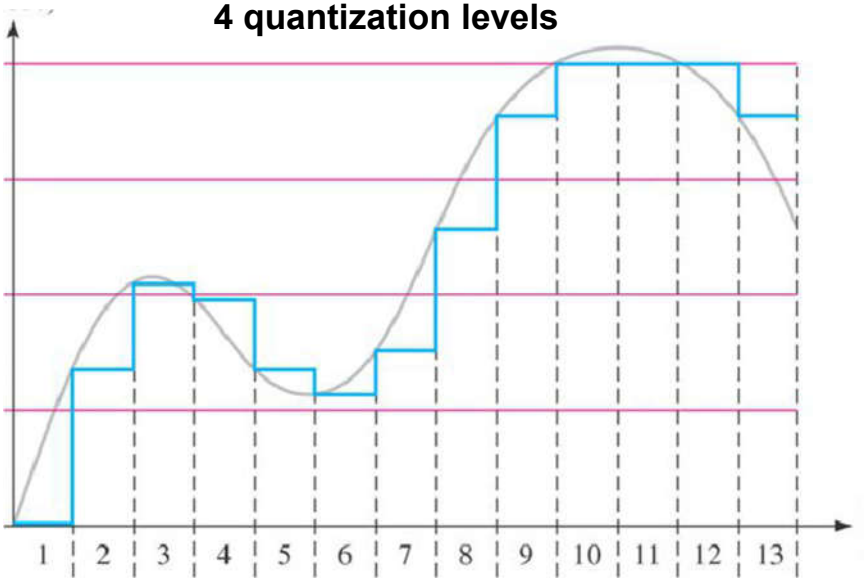
Analog to Digital Conversion – Sample & Hold

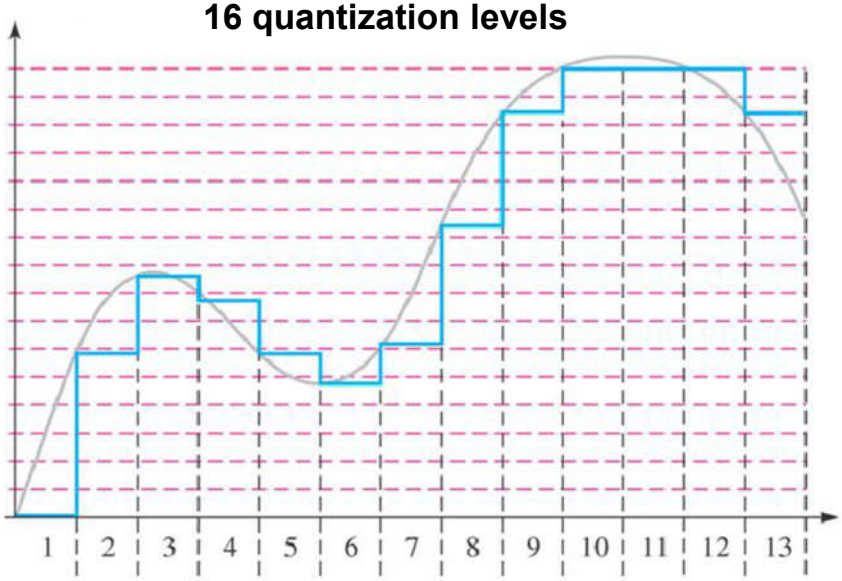
- A sample is a single measurement or read-out of the analog signal performed at a given time.
- Samples tare taken (most often) at regular intervals at a given sampling frequency. The digital signal is only defined at the sampling times.
- Sampling converts the continuous time scale into discrete time samples
- No loss of information if sampling frequency is at least twice the maximum frequency of the signal you are sampling: **Nyquist–Shannon sampling theorem**.
- **Sample and hold**: maintain sampled value for a length of time, i.e. until start of next sample.



Analog to Digital Conversion – Amplitude Quantization

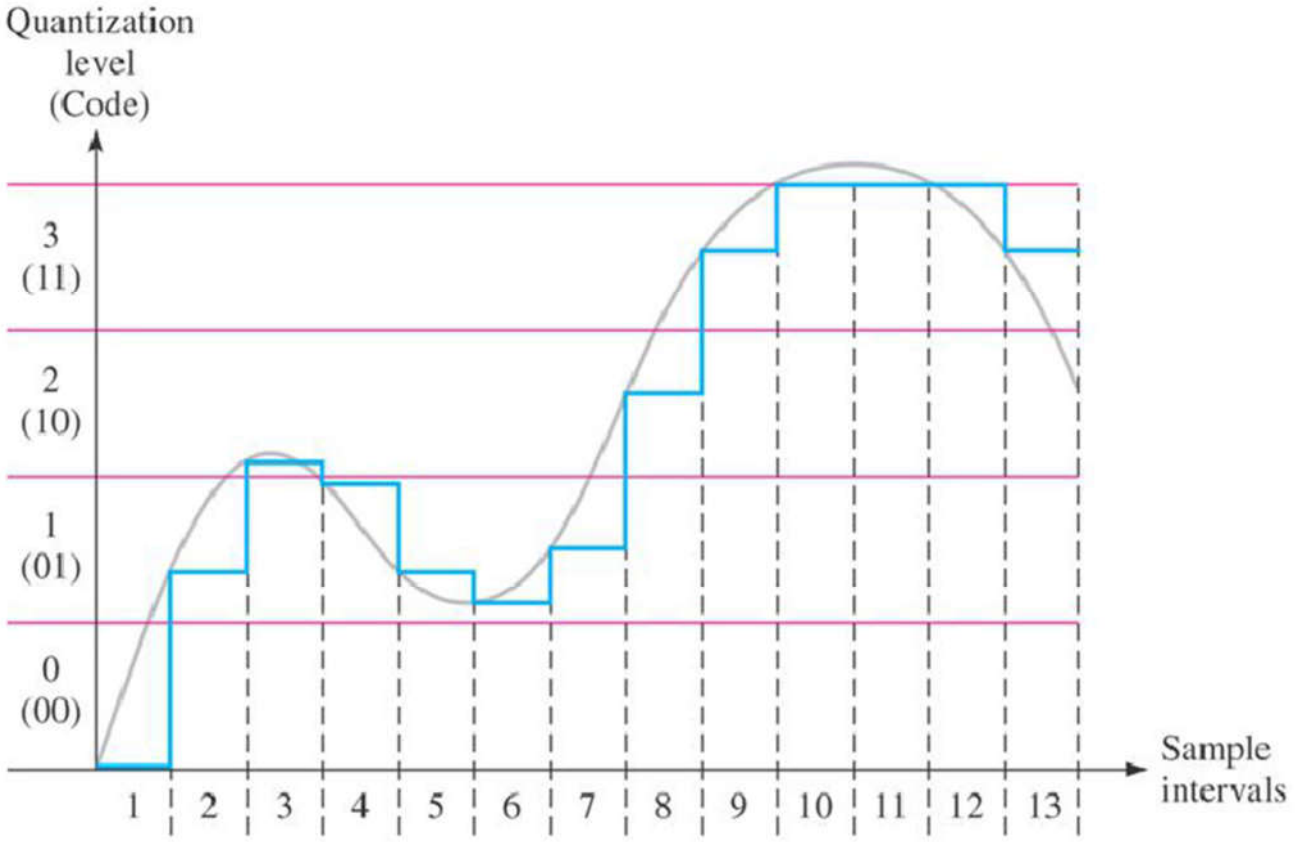
- Quantization converts the continuous amplitude scale to a discrete (finite) set of numbers.
- The amplitude scale is divided into a set of finite numbers (e.g. 256 values: 0 – 255).
- The amplitude of each signal is given one of the finite number of discrete levels, which are sometimes higher and sometimes lower than the original signal.
- This adds **quantization noise**, which is the difference between the quantized representation and the original analog signal.
- Quantization noise leads to an **irreversible corruption** of the signal.
- The more quantization levels the more closely the original signal is reproduced





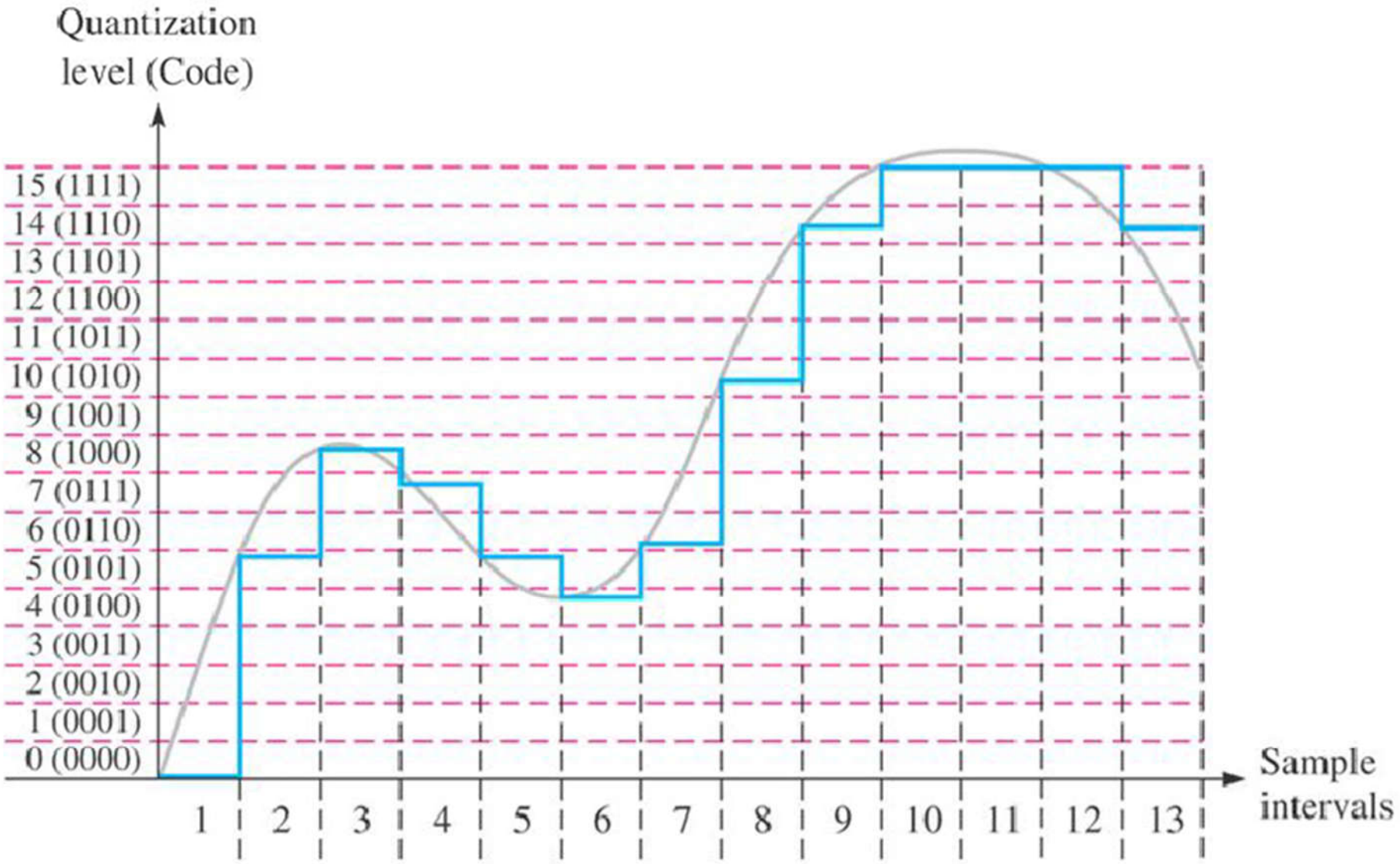
Analog to Digital Conversion – Encoding

- Assigning a unique binary code to each quantization level.
- A bit is a single value, either 1 or 0.
- N-bit code can represent 2^N quantization levels.
  - 2 bits required for 4= 2^2 quantization levels



Analog to Digital Conversion – Encoding

- 4 bits required for 16= 2^4 quantization levels



Bits can represent anything !!!

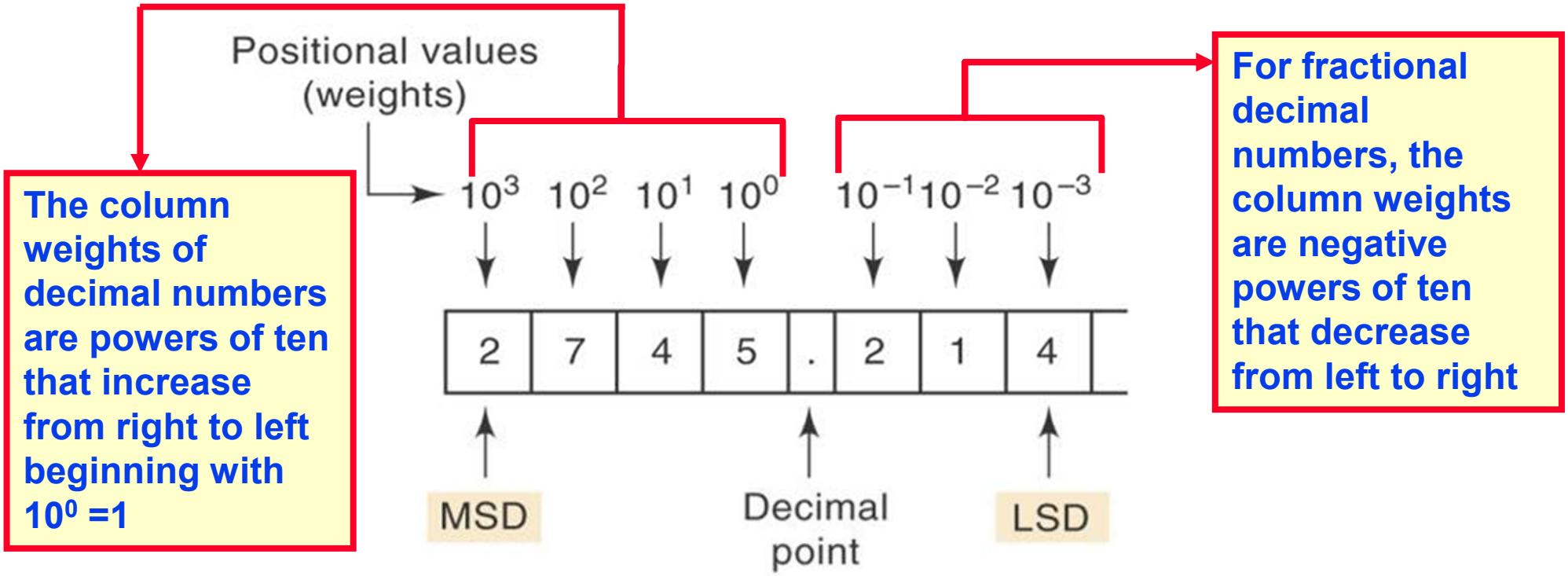
- Characters?
  - 26 letters → 5 bits (25 = 32)
  - upper/lower case + punctuation → 7 bits (“ASCII”)
  - standard code to cover all the world’s languages → 8,16,32 bits (“Unicode”)
- Logical values?
  - 0 → False,
  - 1 → True
- Colors ? **Red (00)** **Green (01)** **Blue (11)**
- Locations, addresses, commands?
- **MEMORIZE:** N bits ⇔ at most 2^N things

Digital Number Systems

- For digital systems, the binary number system is used.
- Number systems differ in the amount of symbols they use:
  - Decimal – 10 symbols (base 10)
  - Binary – 2 symbols (base 2)
  - Hexadecimal – 16 symbols (base 16)
    - used to represent large binary numbers by grouping bits 4 at a time starting from the right.
- The position of each digit in a weighted number system is assigned a weight based on the base (also called radix) of the system.

Decimal Number System

- The Decimal (base 10) System
  - 10 symbols: 0, 1, 2, 3, 4, 5, 6 , 7, 8, 9.
    - Each number is a *digit* (from Latin for *finger*).

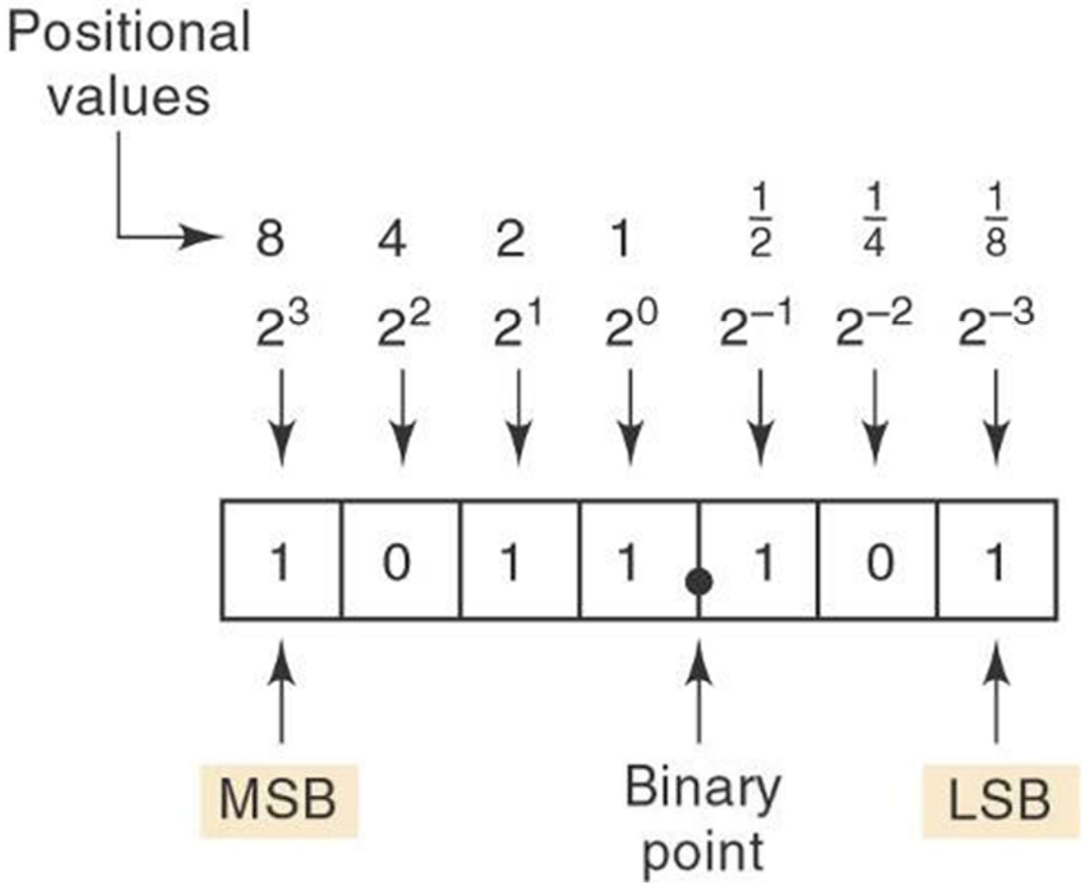


Most significant digit (MSD) & least significant digit (LSD).

Positional value may be stated as a digit multiplied by a power of 10.

Binary Number System

- The Binary (base 2) System
  - 2 symbols: 0,1
    - Lends itself to electronic circuit design since only two different voltage levels are required.



Positional value may be stated as a digit multiplied by a power of 2.

Thinking binary

- Powers of 2
  - Useful to memorize up to about 2^{12}

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$
$2^9 = 512$
$2^{10} = 1{,}024$
$2^{11} = 2{,}048$

$2^{12} = 4{,}096$
$2^{13} = 8{,}192$
$2^{14} = 16{,}384$
$2^{15} = 32{,}768$
$2^{16} = 65{,}536$

$2^{20} = 1{,}048{,}576$

$2^{24} = 16{,}777{,}216$

$2^{30} = 1{,}073{,}741{,}824$
$2^{31} = 2{,}147{,}483{,}648$
$2^{32} = 4{,}294{,}967{,}296$

$2^{10} \approx 1{,}000$ ("k")

$2^{20} \approx 1{,}000{,}000$ ("M")

Binary Numbers

- A binary counting sequence for numbers from 0 to 15 is shown.
- Notice the patterns of 0's and 1's in each column.
- The decimal equivalent of a binary number can be determined by adding the column values of all of the bits that are 1 and discarding all of the bits that are 0.
- Convert the binary number 100101.01 to decimal.
  - Start by writing the column weights; then add the weights that correspond to each 1 in the number:

2^5	2^4	2^3	2^2	2^1	2^0 .	2^{-1}	2^{-2}
32	16	8	4	2	1 .	½	¼
1	0	0	1	0	1.	0	1

32 +4 +1 +¼ = 37¼

Decimal Number	Binary Number
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

Hexadecimal Number System

- Hexadecimal uses sixteen characters to represent numbers: the numbers 0 through 9 and the alphabetic characters A through F.
- Large binary number can easily be converted to hexadecimal by grouping bits 4 at a time and writing the equivalent hexadecimal character.
- Express **1001 0110 0000 1110**$_2$ in hexadecimal:
- Group the binary number by 4-bits starting from the right. Thus, **960E**

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Hexadecimal Number System

- Hexadecimal is a weighted number system. The column weights are powers of 16, which increase from right to left.

Column weights	16^3	16^2	16^1	16^0.
	4096	256	16	1

- Express **1A2F**$_{16}$ in decimal:
  - Start by writing the column weights:

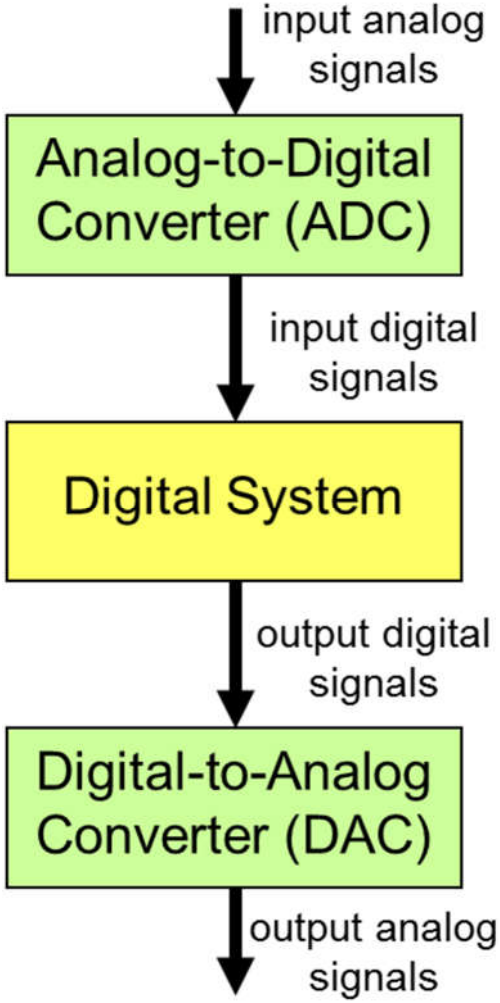
4096	256	16	1
1	A	2	F_{16}

1(4096) + 10(256) +2(16) +15(1) = 6703_{10}

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

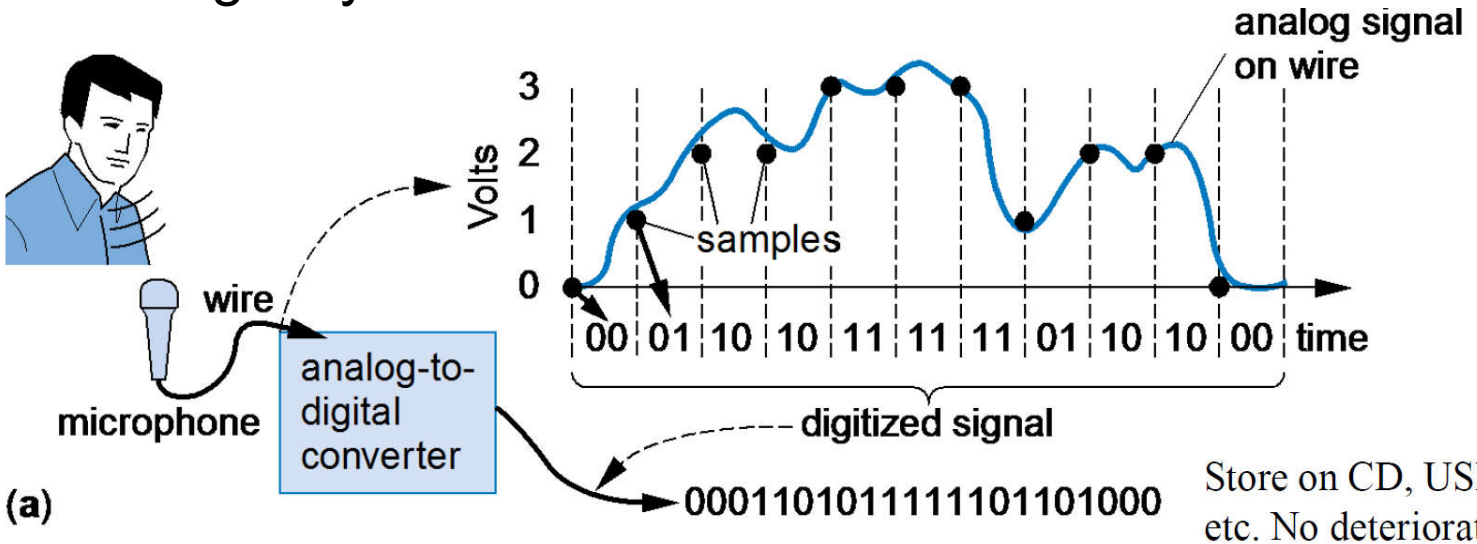
ADC and DAC Converters

- Analog-to-Digital Converter (ADC)
  - Produces digitized version of analog signals
  - Analog input => Digital output
- Digital-to-Analog Converter (DAC)
  - Regenerate analog signal from digital form
  - Digital input => Analog output
- Our focus is on digital systems only
  - Both input and output to a digital system are digital signals

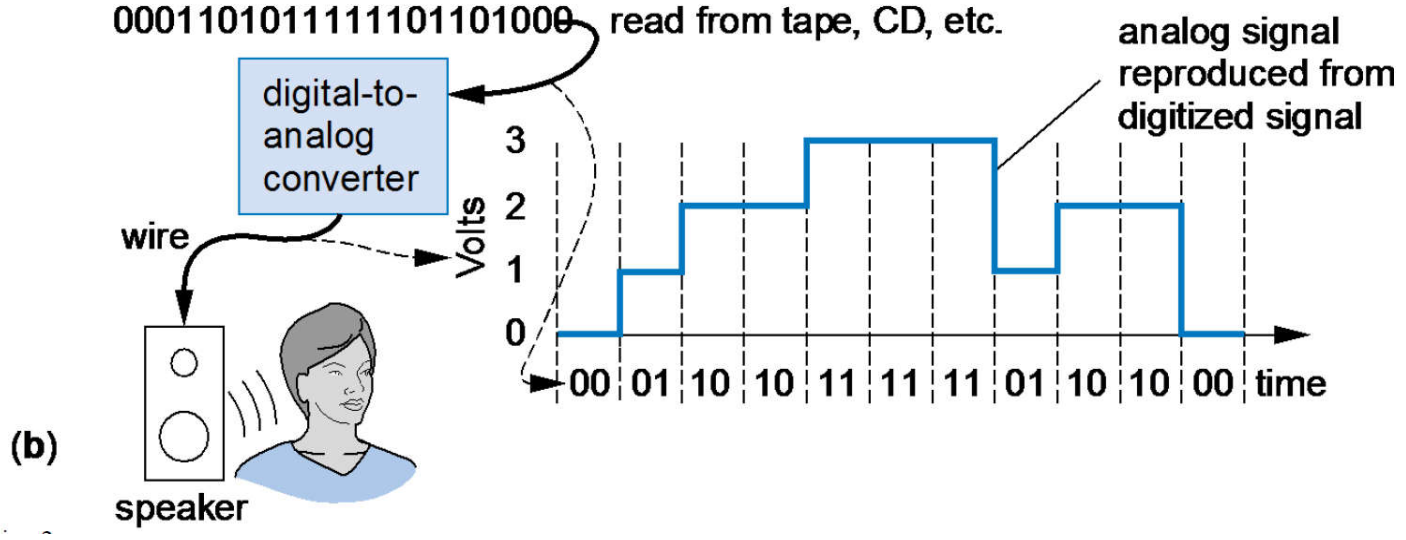


Digitization of Analog Signals

- The "real" or "physical" world is analog but analog signals can be converted to digital by taking measurements or "samples" of the continuously varying signal at regular intervals. Digitized output coded using only 2 values: 0 & 1



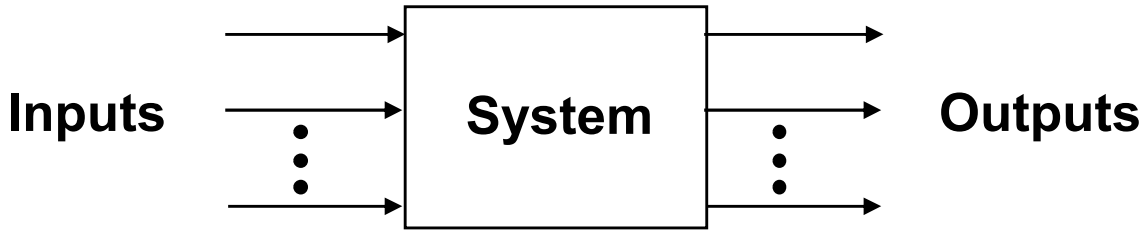
Store on CD, USB drive, etc. No deterioration.



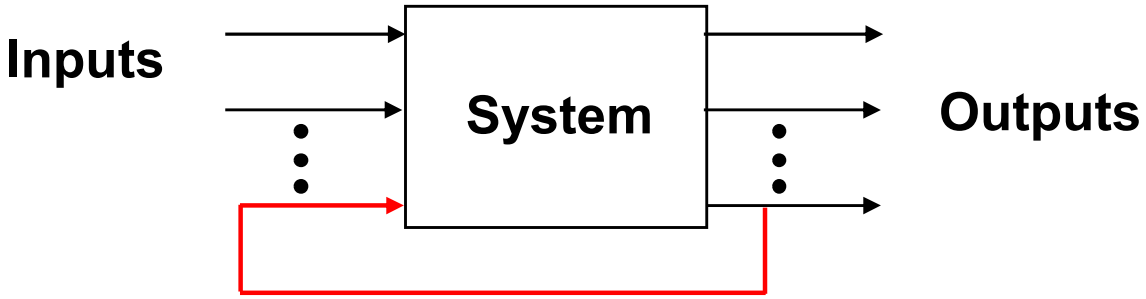
Higher accuracy by adding more bits

2 types of Digital systems

- Combinational systems are memoryless
  - The outputs depend only on the present inputs



- Sequential systems have memory
  - The outputs depend on the present inputs and on the previous inputs



Microprocessor

- Most sophisticated digital system available today
- General-purpose and programmable, it manipulates data based on instructions in the form of a program.
- Found in a wide range of products (not just computers!)
- Understanding microprocessors will enable you to design a wide range of digital hardware.

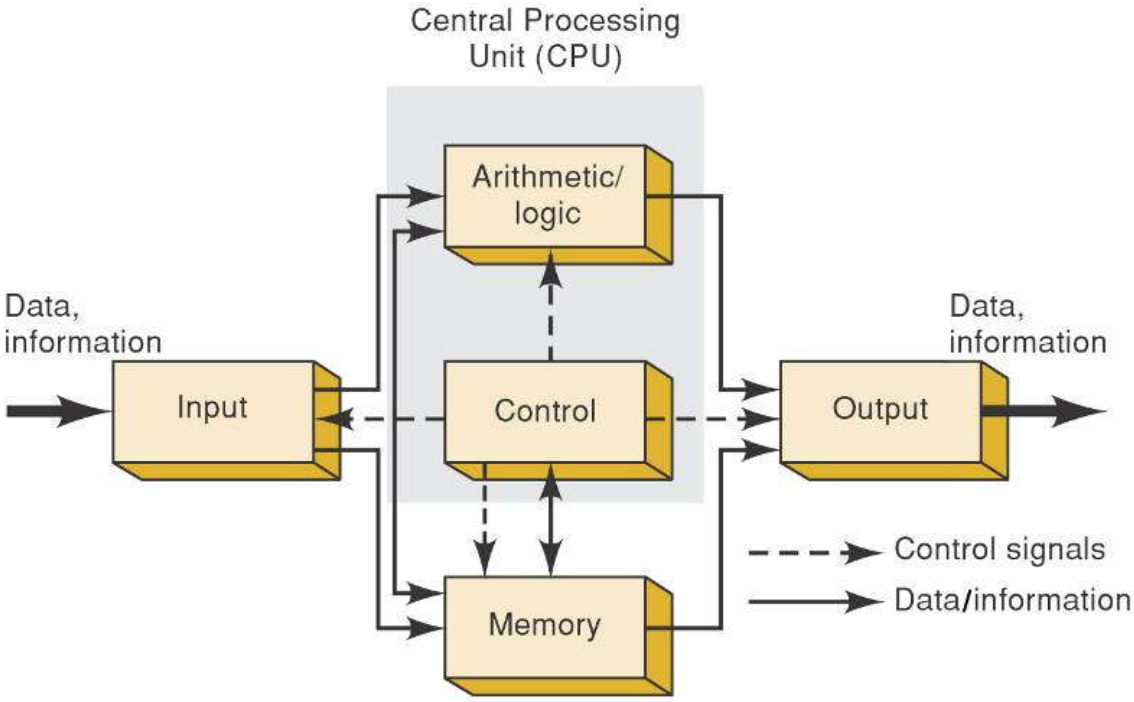






Digital Computers

- Major parts of a computer:
  - **Input unit**—Processes instructions and data into the memory.
  - **Memory unit—**Stores data and instructions.
  - **Control unit**—Interprets instructions and sends appropriate signals to other units as instructed.
  - **Arithmetic/logic unit**—arithmetic calculations and logical decisions are performed.
  - **Output unit**—presents information from the memory to the operator or process.



The Modest Switch

- Digital systems are built using an extremely simple component:

  A controllable *switch*

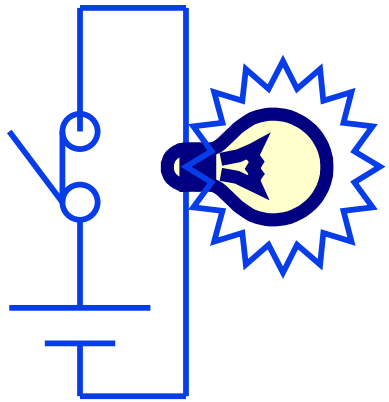
- The usual Electrical switch we use every day

  The electric switch we use turns current on and off

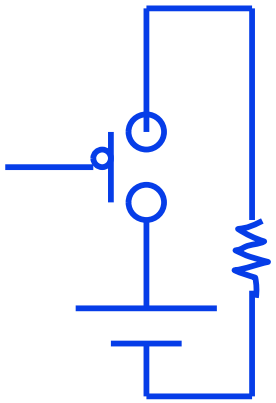
  But we need to turn it on and off by hand

  The result of turning the switch on?

  - The “top end” in the figure becomes
  - raised to a high voltage
  - Which makes the current flow through the bulb

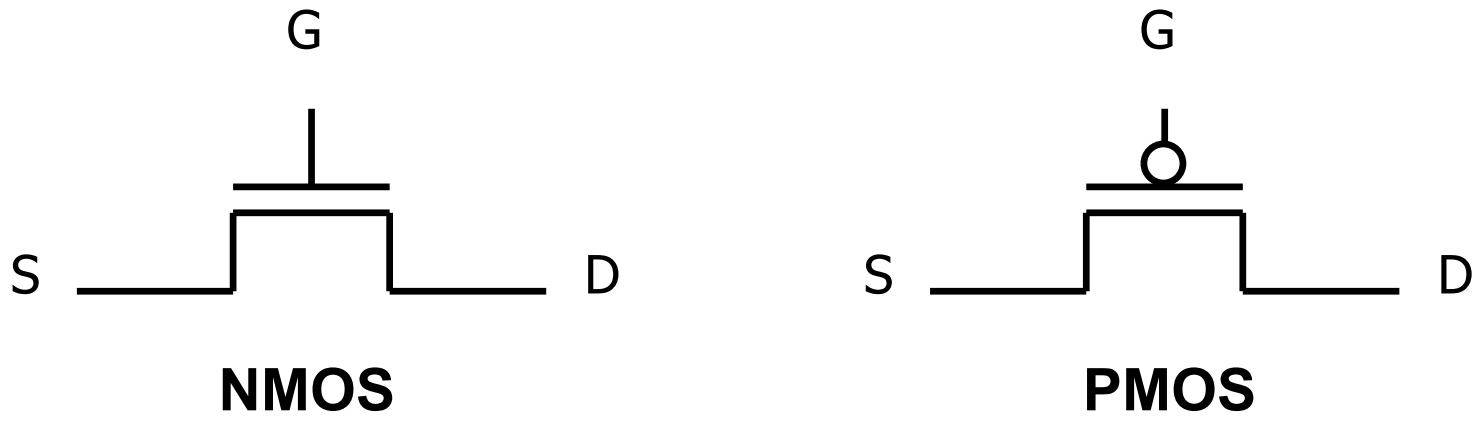


- The Controllable Switch
  - No hands
  - Voltage controls if the switch is on or off
  - High voltage at input: switch on
    - Otherwise it is off



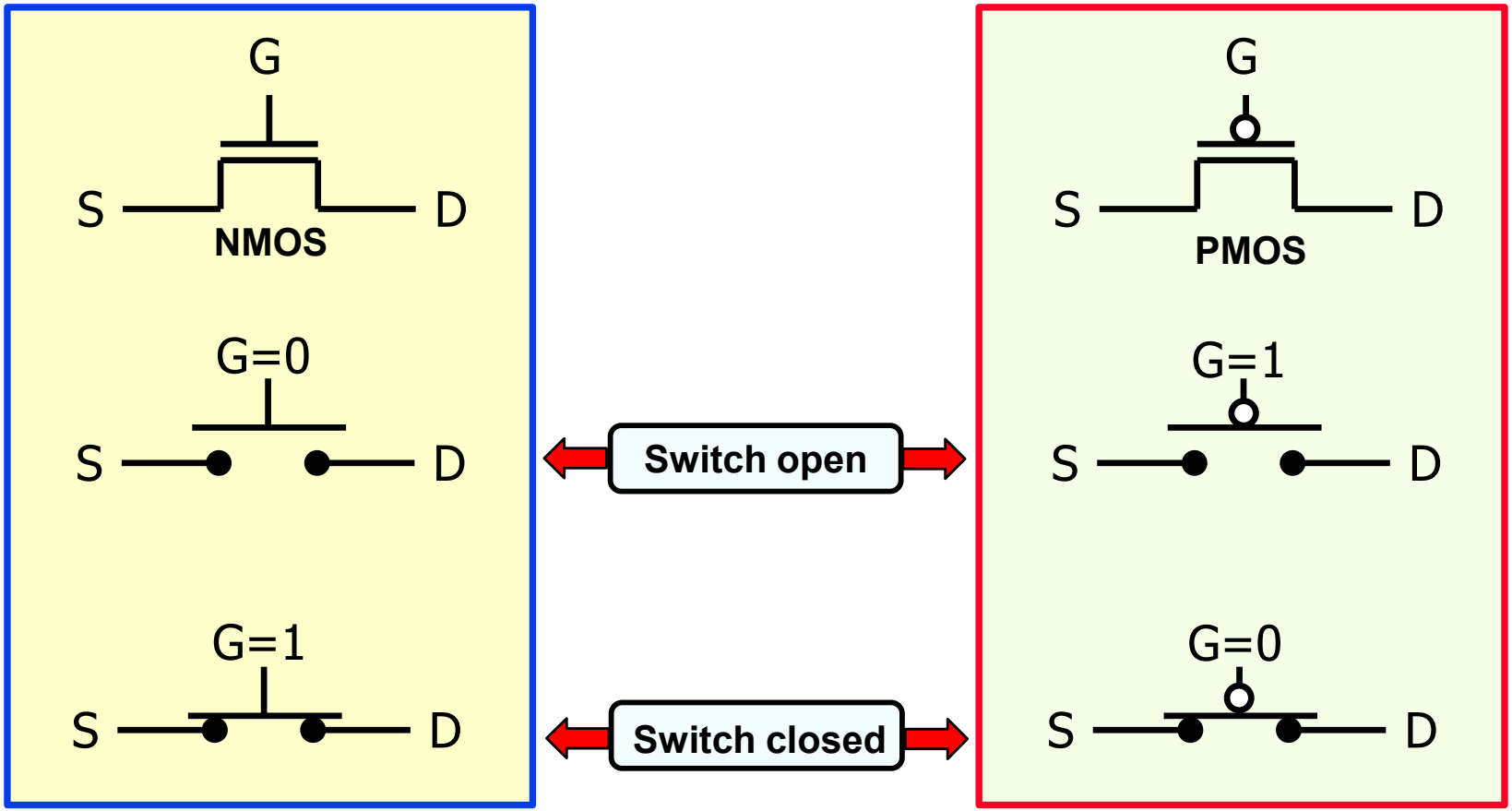
Transistor as switches

- Modern digital systems are designed in CMOS technology
  - MOS stands for Metal-Oxide on Semiconductor
  - C stands for complementary because there are both normally-open and normally-closed switches
- MOS transistors act as voltage-controlled switches
- MOS transistors have three terminals: drain, gate, and source.

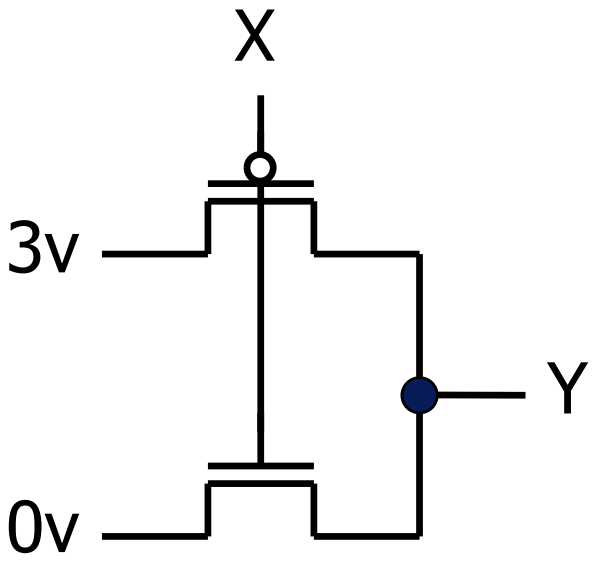


MOS transistors

- PMOS/NMOS transistors act as complementary switches:
  - **NMOS:** if the voltage on the gate terminal is "**1**" (sufficiently high voltage) then a conducting path will be established between the drain and source terminals.
  - **PMOS:** if the voltage on the gate terminal is "**0**" (sufficiently low voltage) then a conducting path will be established between the drain and source terminals.



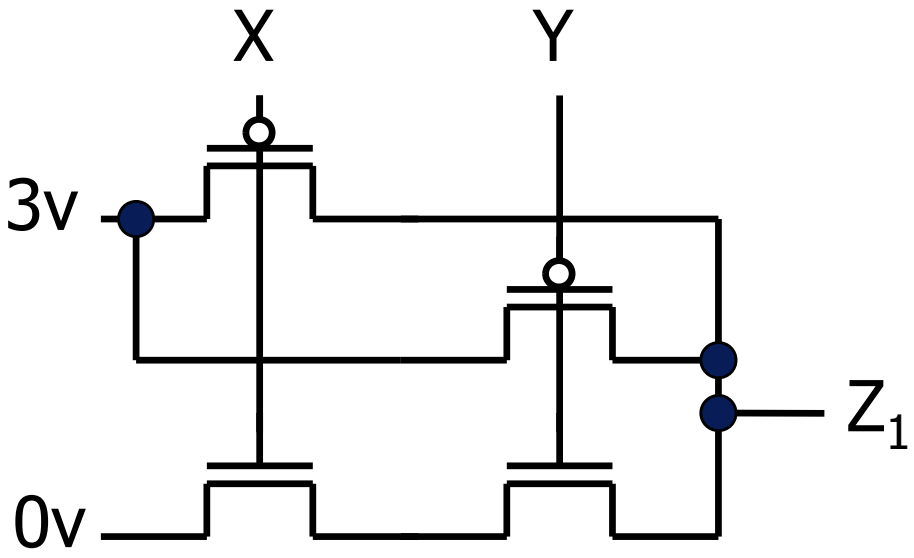
MOS networks

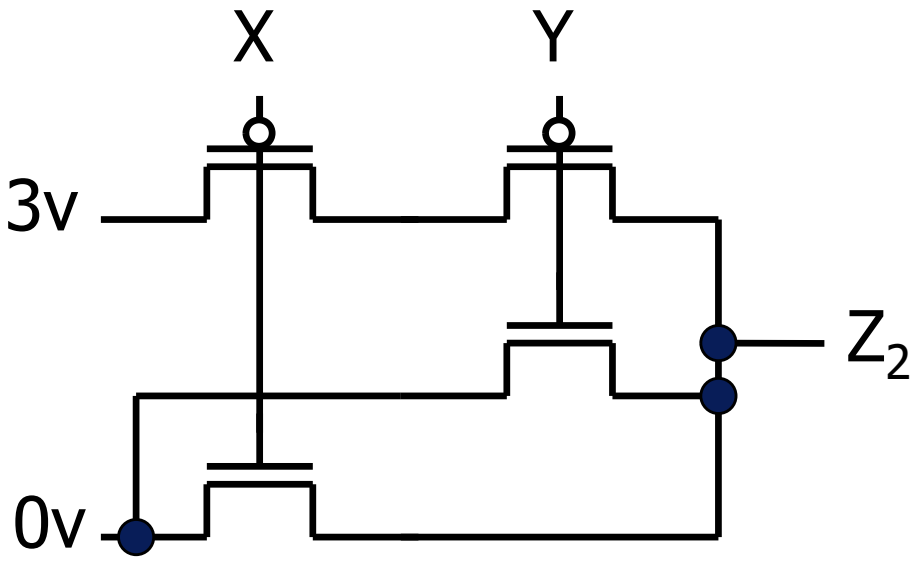


what is the relationship between x and y?

x	y
0 volts	
3 volts	

Two input networks

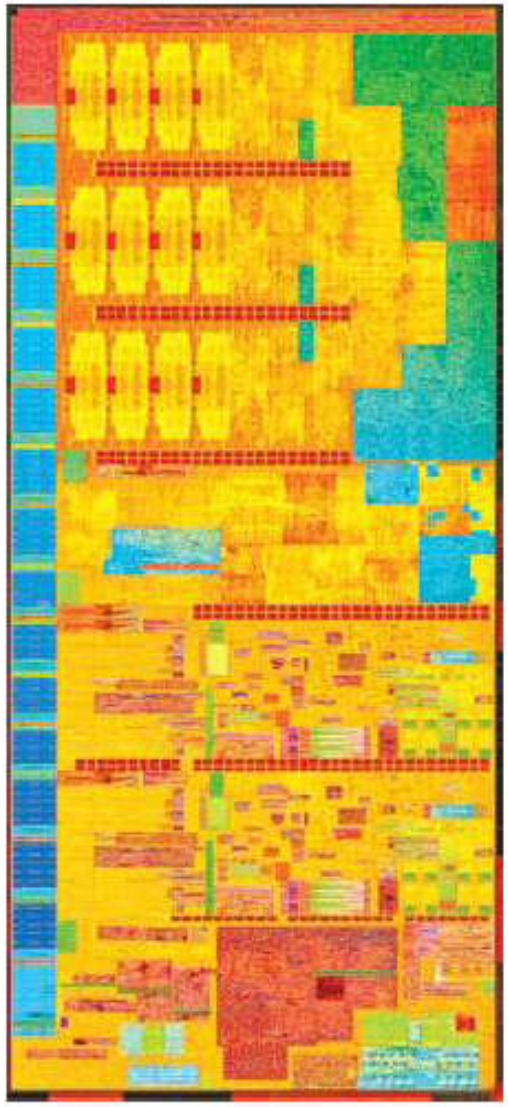




what is the relationship between x, y and z?

x	y	z1	z2
0 volts	0 volts		
0 volts	3 volts		
3 volts	0 volts		
3 volts	3 volts		

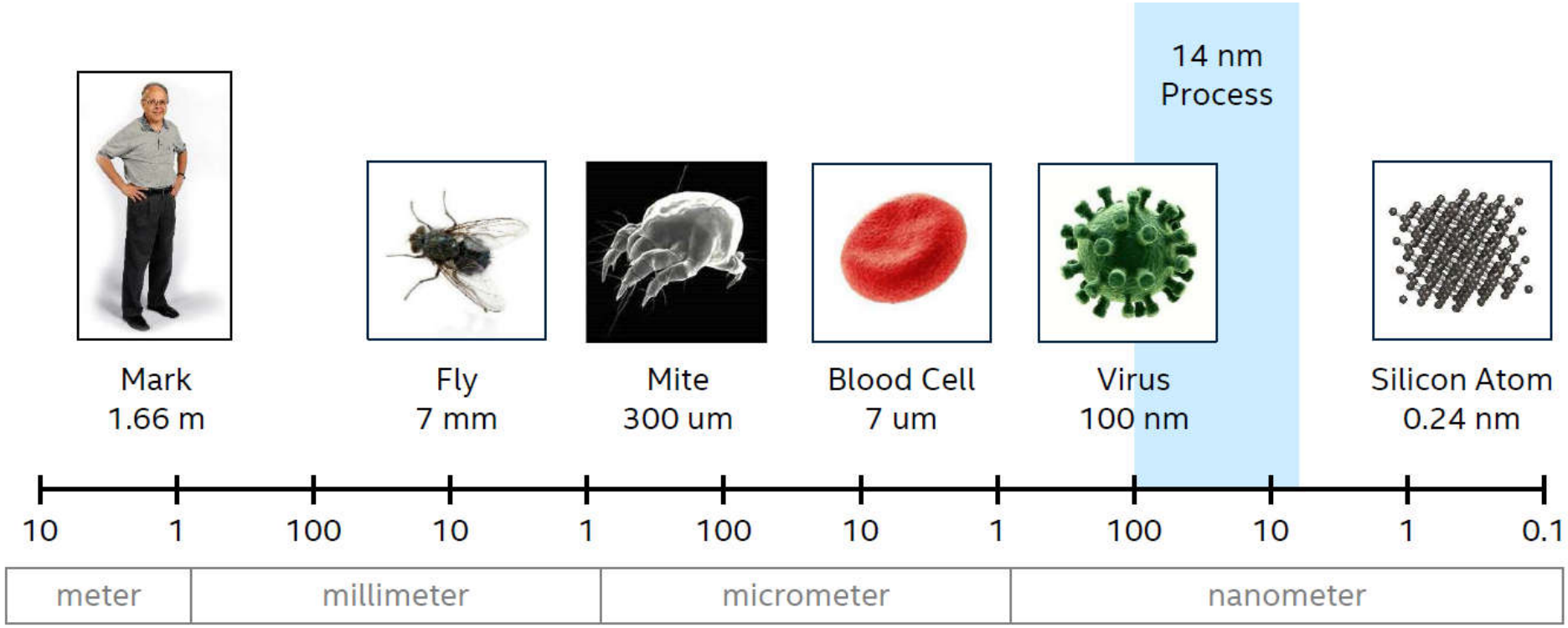
14 nm Intel® Core™ M Processor



1.3 billion transistors

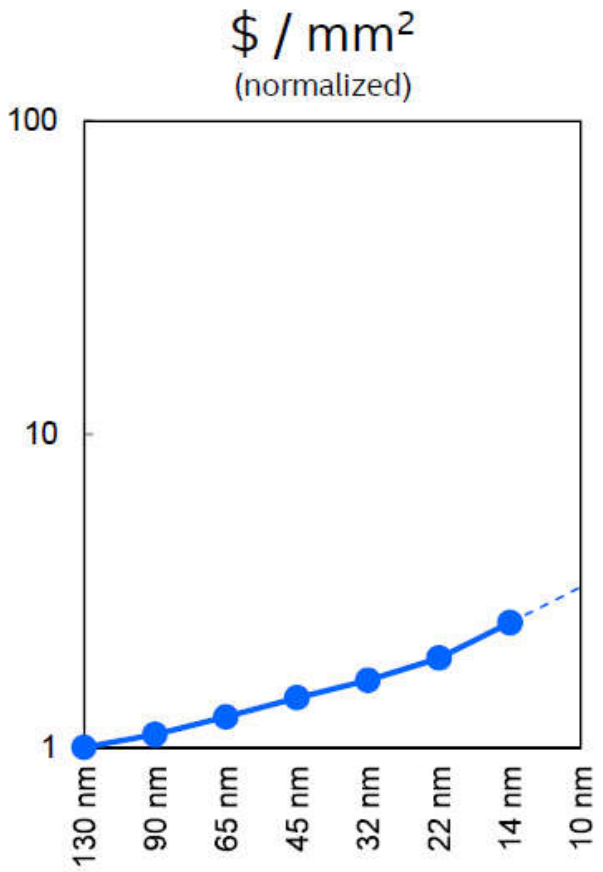
82 mm^2 die size

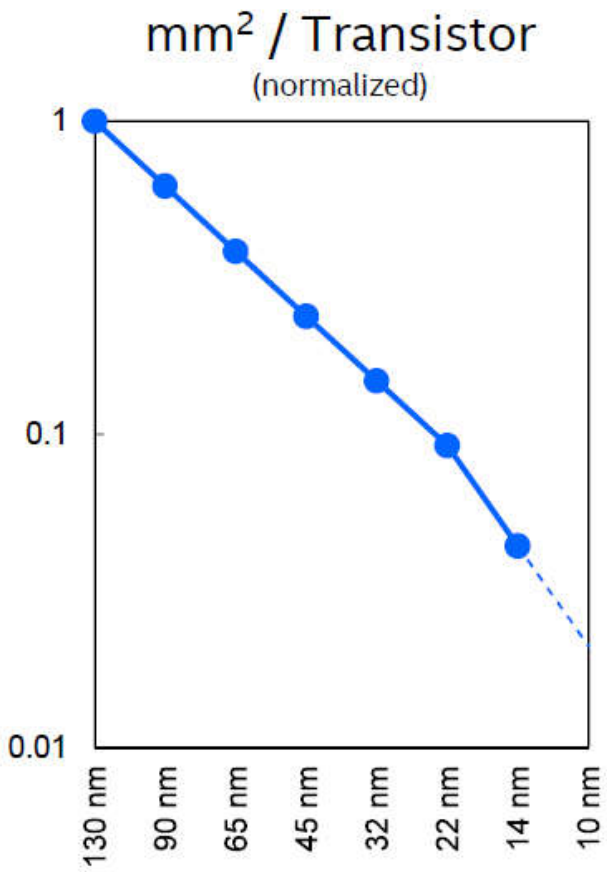
How small is 14nm?

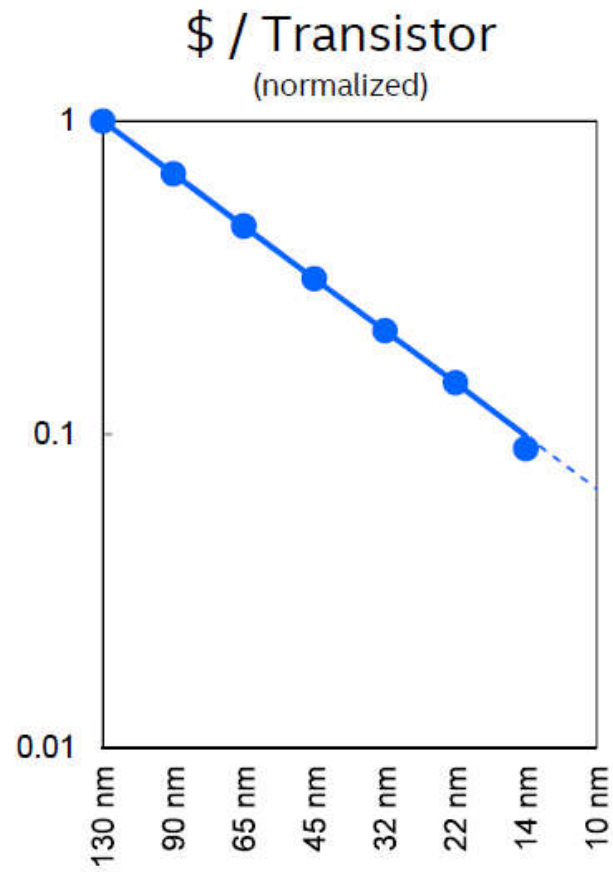


Getting closer to physical limits !

Cost per transistor continues to go down!

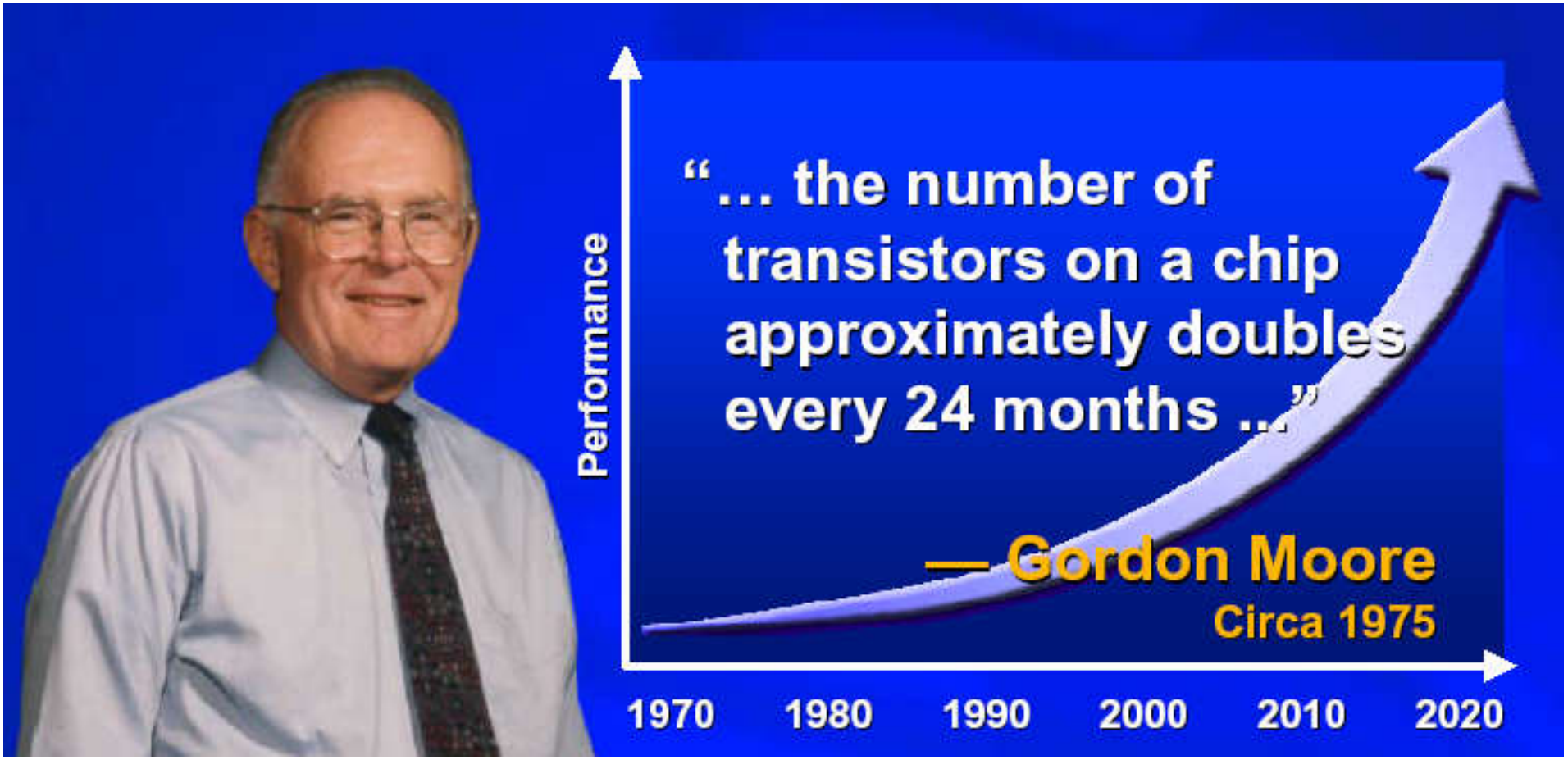




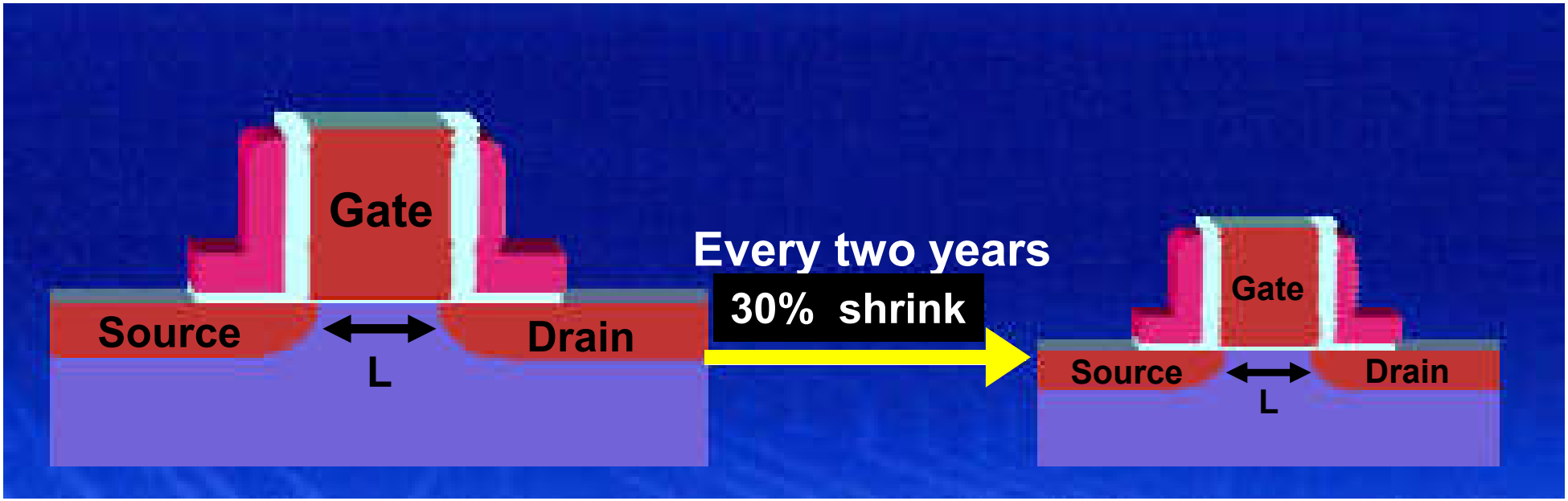


Moore's law continues!

- In 1975, Gordon Moore: co-founder of Intel, predicted that:

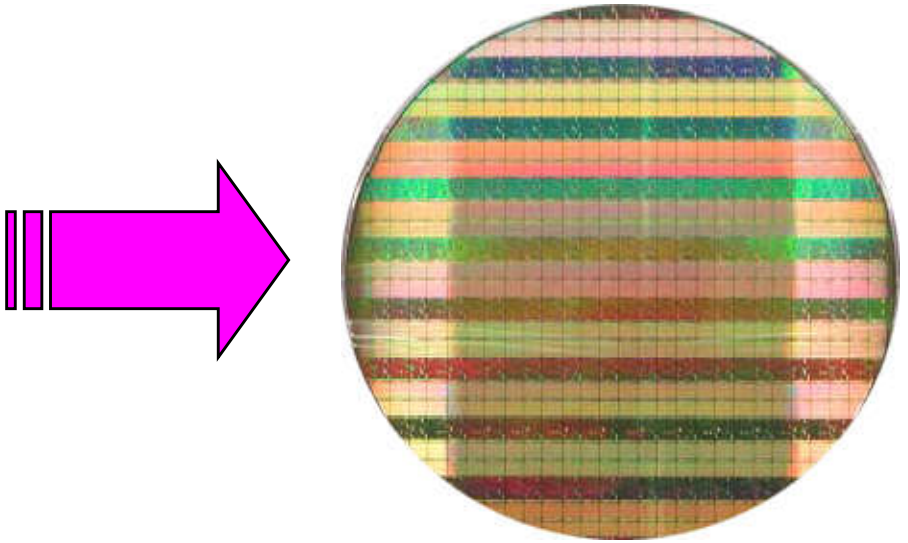


Moore's Law: Smaller-Cheaper-Better model



Why do we scale MOSFET?

- **Reduce cost by putting more components on a single wafer**
- **Improve speed α 1/L**

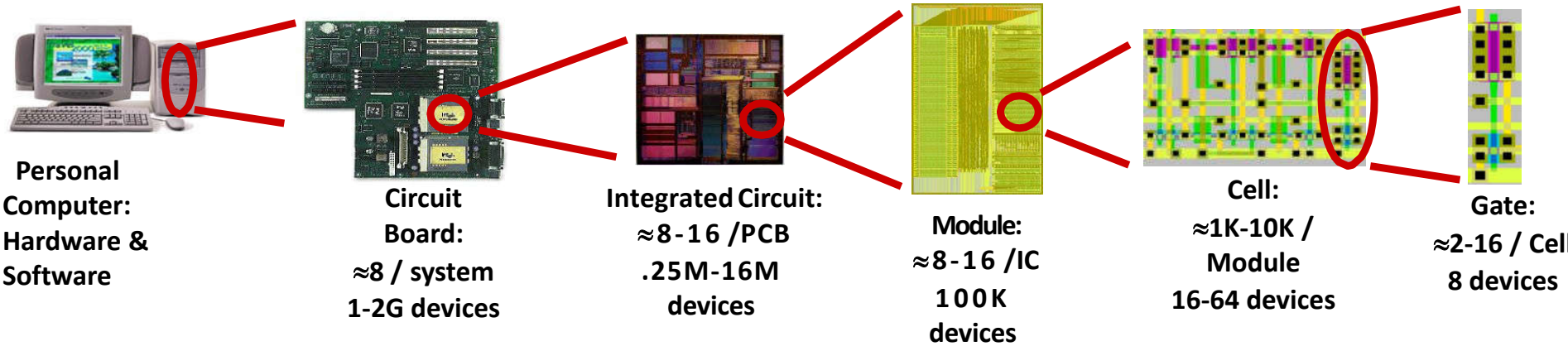


What is Digital system Design?

- What is design?
  - given a specification of a problem, come up with a way of solving it choosing appropriately from a collection of available components
  - while meeting some criteria for size, cost, power, beauty, elegance, etc.
- What is digital system design?
  - determining the collection of digital logic components to perform a specified control and/or data manipulation and/or communication function and the interconnections between them
  - which logic components to choose? – there are many implementation technologies (e.g., off-the-shelf fixed-function components, programmable devices, transistors on a chip, etc.)
  - the design may need to be optimized and/or transformed to meet design constraints

The Key to System Design

- A system will exhibit a specified behaviour if all of its components obey their specified behaviors.
- How is this achieved?
  - **Contracts!**
- Every system component will have clear obligations and responsibilities. If these are maintained, one should expect the system to behave as planned.



Dealing with design complexity

- How does one design large and complex digital systems (e.g. microprocessors) comprising perhaps billions of components?
  - Abstraction and Hierarchy/modularity
  - Design Automation tools
  - Hardware Description languages

Abstraction

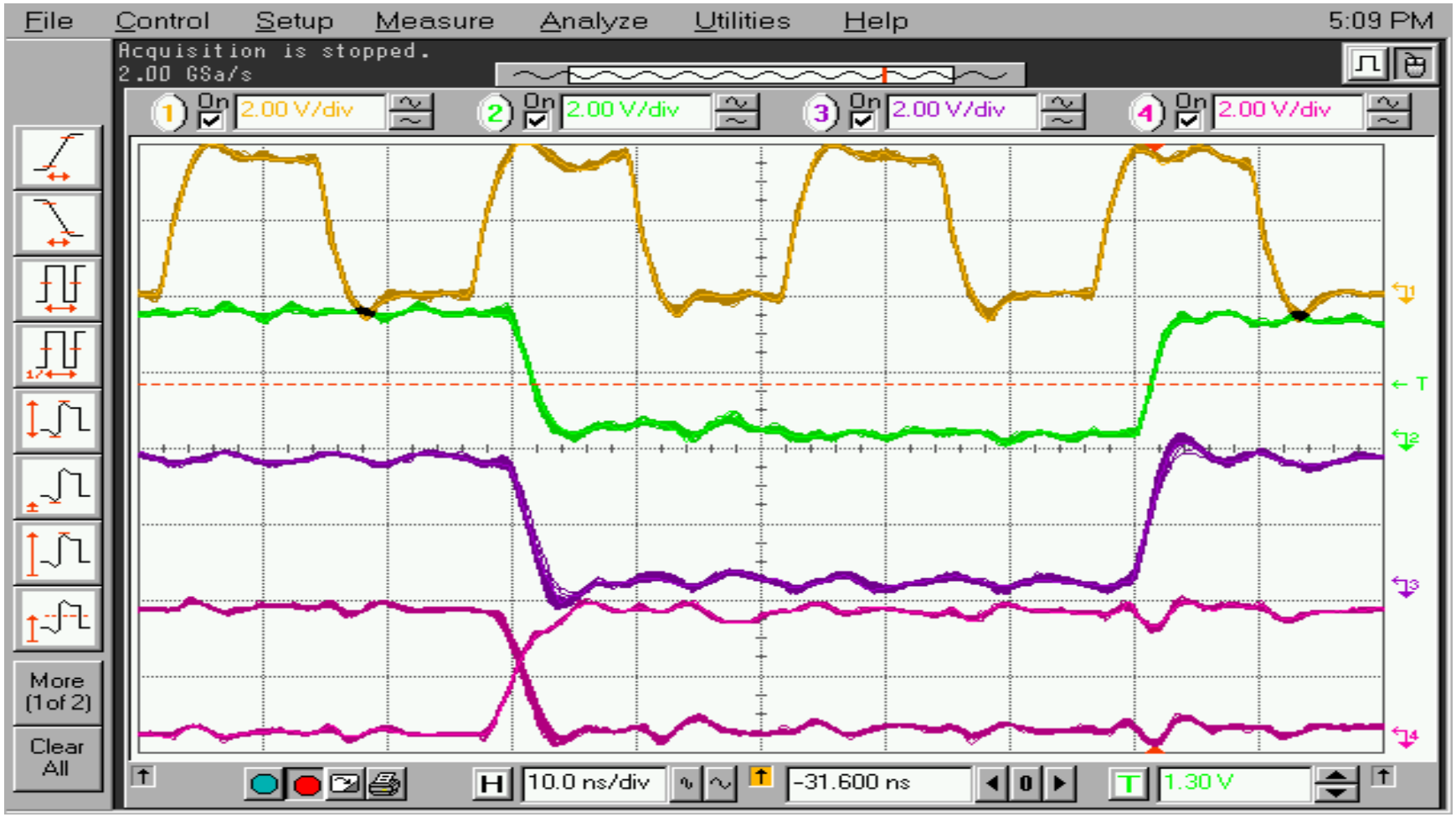
- In digital system design, we rely on different levels of **abstractions** (e.g. digital abstraction, Boolean Algebra) to manage the design process, that is, we use **models** with different levels of detail and/or from **different perspectives**.

- Abstraction can thus provide a **simplified model of a system**. Abstractions may be formed by :
  - **reducing the information** content of a concept or an observable phenomenon, to **retain only information which is relevant for a particular purpose/perspective.**

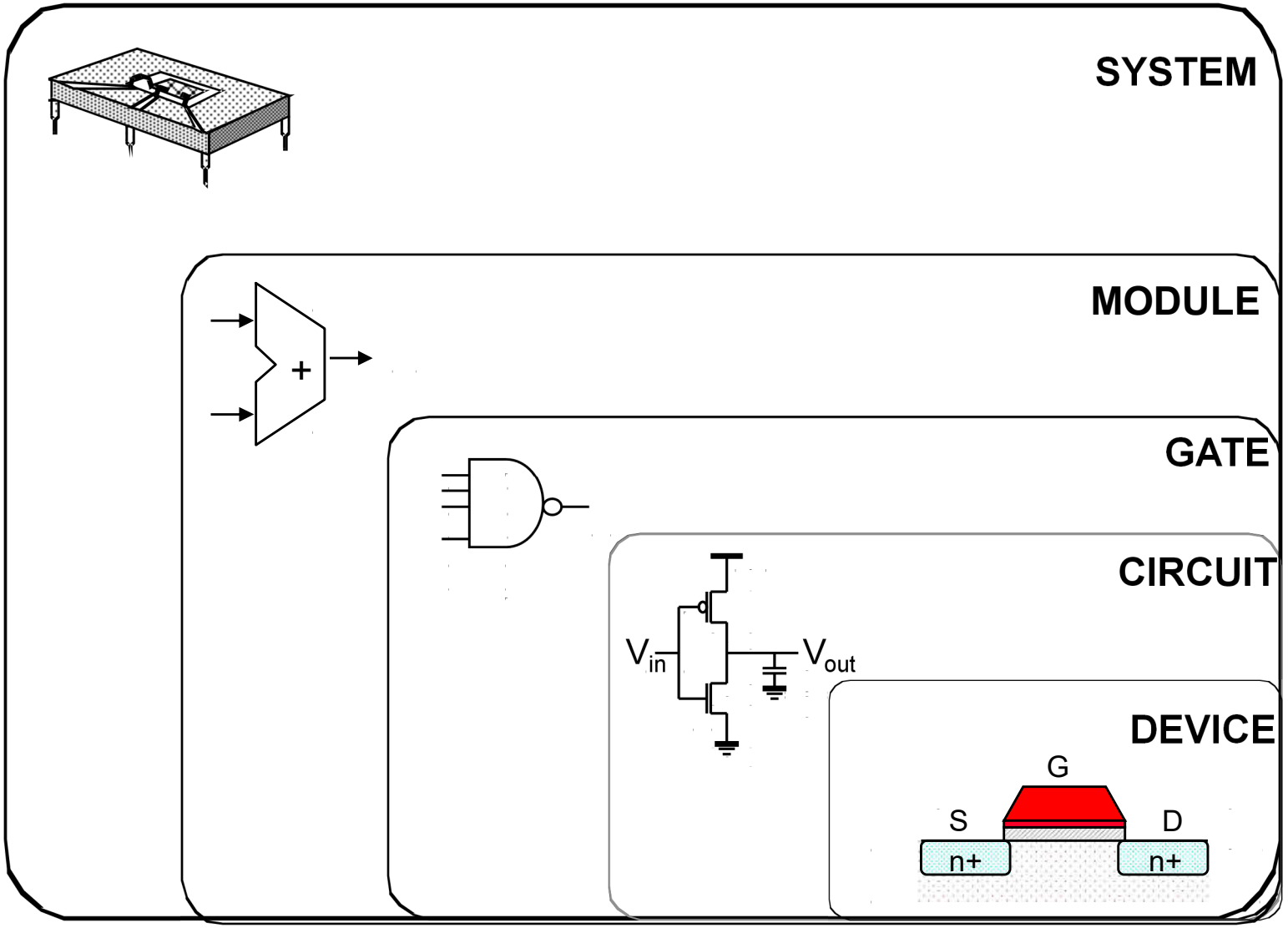
- One should always keep in mind **what assumptions** make it possible to have **useful abstractions.**

- One should ensure that the **appropriate assumptions are valid** and that the system is designed with **acceptable failure rates and failure types**.

Digital Abstraction – Are these good digital signals?



Multiple levels of abstractions



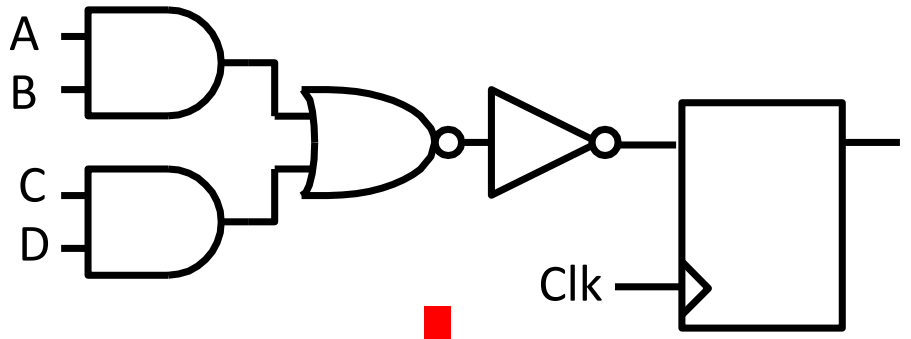
Hierarchical/modular Design

- Top-down design strategies
  - Refine Specification successively
  - Decompose each component into small components
  - Lowest-level primitive components
- Bottom-up design strategies
  - Build-up from primitive components
  - Combined to form more complex components
  - Risk wrong interpretation of specifications
- Mixed strategies
  - Mostly top-down, but also bits of bottom-up
  - Reality: need to know both top level and bottom level constraints

Hardware Description Language

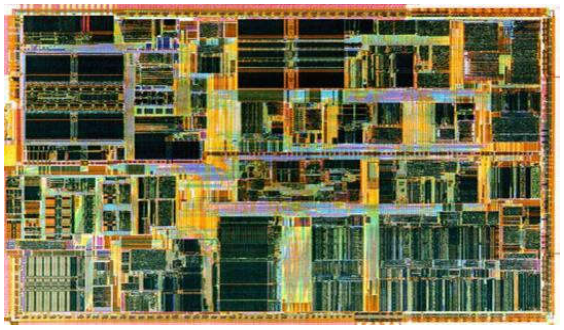
- Conventional Approach:
  - Schematic Entry → good for fairly small designs

    (Draw K-maps, optimize the Boolean logic, draw the schematic)



(10 gates)

  - Possible for large designs?
    - **NO!**



(10 millions gates)

Why use a Hardware Description Language?

- Schematic entry not feasible for large designs:
  - Time consuming to draw the schematic for millions of gates
  - Prone to mistakes
  - Difficult design entry and sharing
  - Different design entry tools to learn
  - Tools not compatible (hard to convert the design from one to another)
  - Not easy to modify
- Solution:
  - Describe the design in text using a **Hardware Description language or HDL**
  - Just describe the design "**behavior**" not the detailed gate-level logic
  - Gate-level logic is generated automatically by a "**synthesis**" tool

Computer-Aided Design (CAD) Tools

A CAD package include the following tools:

- Design entry: the process of entering a description of the circuit being designed through a truth table, wave editor, schematic entry or hardware description language.

- Synthesis and optimization: the process of generating a digital circuit from stated functional behavior using components from targeted technology. Not only will the CAD tool produce a logic circuit but it can also optimize it in terms of speed, size and /or power.

- Simulation: To verify that the circuit functions as expected:

  In a ***functional simulation,*** the user specifies input vectors and the CAD tool generates the outputs (timing diagram). User verifies generated outputs against expected outputs.

  Use a ***timing simulator*** to obtain accurate (complete) simulation

Benefits of a Hardware Description Language

- Quick Time-to-Market
  - Allows designers to quickly develop designs requiring thousands of logic gates
  - Provides powerful high-level constructs for describing complex hardware without actually designing the hardware itself
  - Supports modular design methodology and multiple levels of hierarchy
  - Facilitate the re-use of previously designed components
- Allows creation of **device-independent** designs that are portable to multiple vendors.
  - Allows automated mapping of your high-level description to a technology-specific implementation
  - Allows user to easily explore design alternatives and delay decisions on implementation details
  - Allows user to pick any synthesis tool, vendor, or device
- One language for design and simulation

A glimpse of VHDL

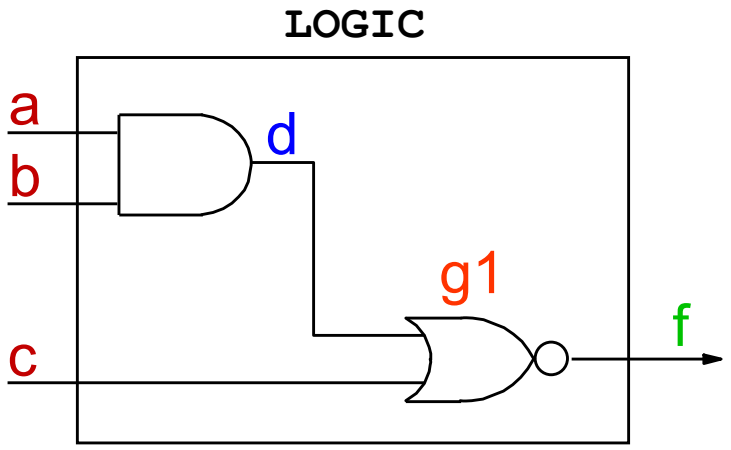
- VHDL acronym for **V**HSIC **H**ardware **D**escription **L**anguage, VHSIC standing for Very High Speed Integrated Circuit
- The ENTITY describes the periphery of the black box (Ports I/O) and the ARCHITECTURE describes the behavior of the entity: they are paired together to form a design.

```
ENTITY logic IS PORT (
     a,b,c: IN STD_LOGIC;
     f:     OUT STD_LOGIC);
END logic;

USE WORK.a_package_name.ALL;
ARCHITECTURE archlogic OF logic IS
     SIGNAL d: STD_LOGIC;
BEGIN
     d <= a AND b;
     g1: nor2 PORT MAP (c, d, f);
END archlogic;
```

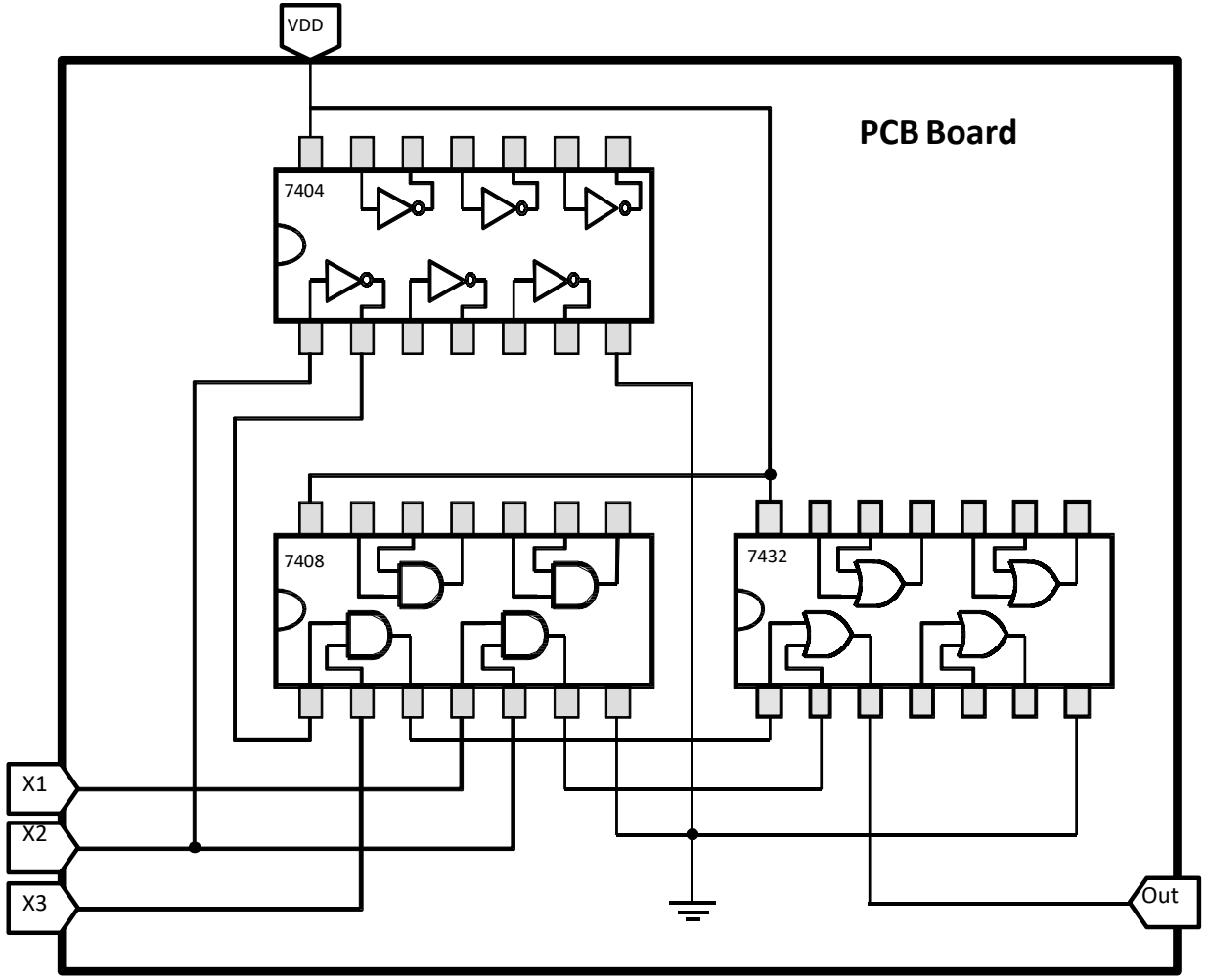
Behavioral

Structural



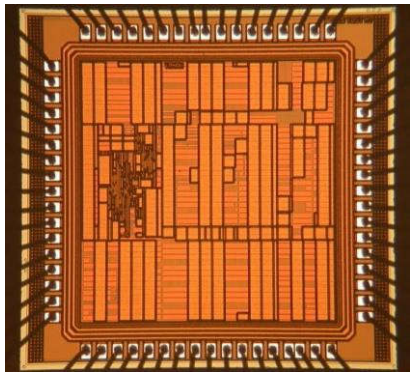
Conventional Realization approach

- Large number of chips (containing basic logic gates) on a single Printed Circuit Board (PCB)



Modern Realization – High-density Single Chip

- A single chip replaces the whole multi-chip design on PCB :
  - Lower overall cost
  - On-chip interconnects are many times faster than off-chip wires
  - Lower area with the same functionality
  - Lower power consumption
  - Lower noise
- Two single-chip solutions: Programmable Logic Devices (PLDs) or Application Specific Integrated Circuits (ASICs)



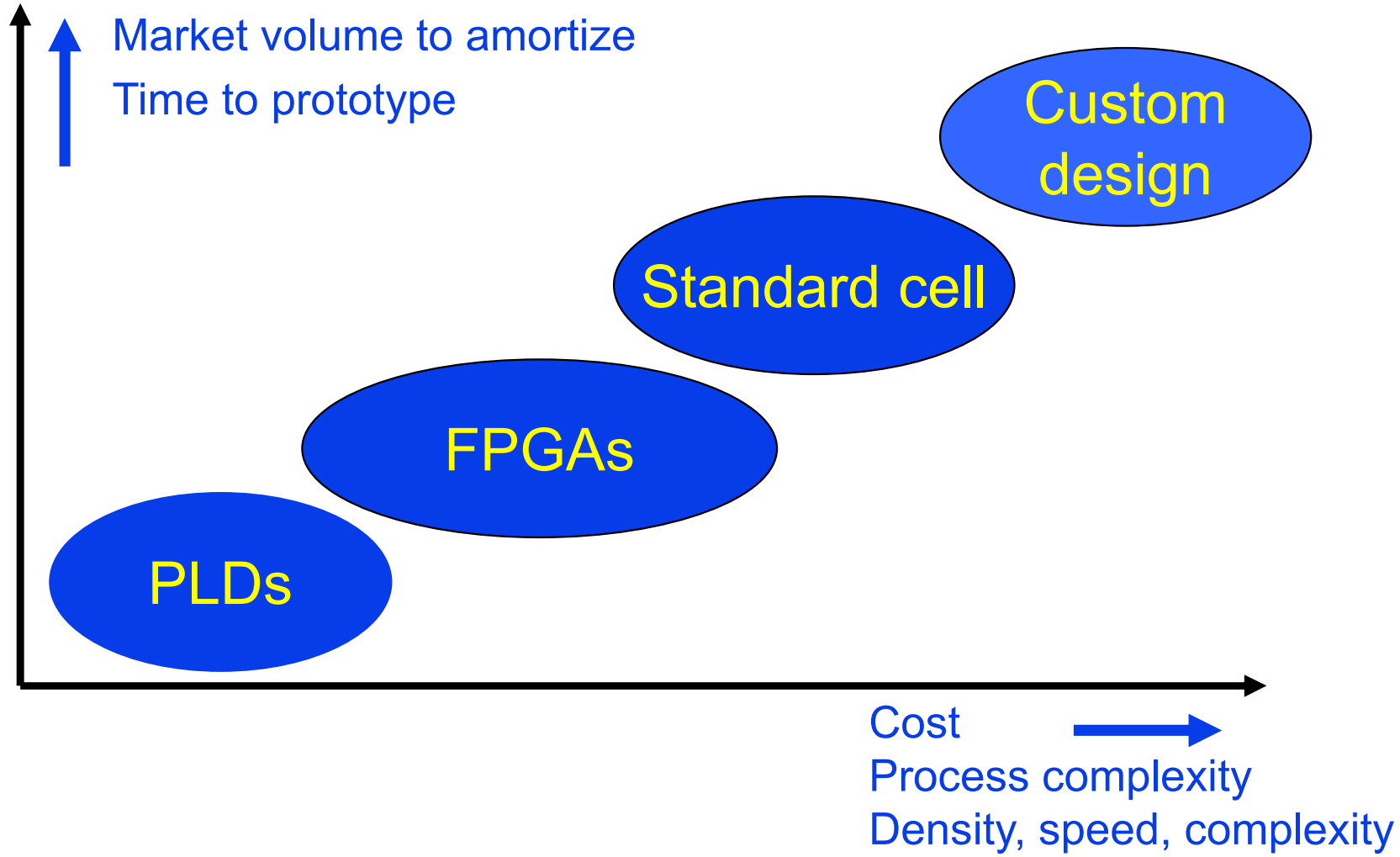
ASIC

PLD

IMPLEMENTATION TECHNOLOGIES

- Custom design
  - Mostly manual design, long design cycle
  - High performance, high volume
  - Microprocessors, analog, IP …
- Standard cell
  - Pre-designed cells, CAD, short design cycle
  - Medium performance ASIC
- FPGA/PLD
  - Pre-fabricated, fast automated design, low cost
  - Prototyping, reconfigurable computing

COMPARISON OF TECHNOLOGIES



Benefits of Digital Circuits

- Tolerance to noise
- Reproducibility of information
- **Flexibility and functionality**: easier to store, transmit and manipulate information
- **Shorter time to market** because **easier to design** with well-developed techniques and automated tools
- Benefits directly from **Moore's law** – The number of transistors on a chip doubles every 24 months – cost/function can decrease by 29%/year for digital circuitry
- Digitization results in **smaller, faster and cheaper circuits**

Acknowledgments

- Credit is acknowledged where credit is due. Please refer to the full list of references.