COM-601.3

Digital Logic and Computer Organisation

Basanta Joshi, PhD

Course Objectives

- Main objective of this course is to provide the student with the fundamental knowledge about the basic building logical blocks of digital computer systems.

- This course also aims at familiarizing the student to design aspect of the logic elements of a computer system.

Syllabus

1. Digital Computers and Information(4 hrs)

Digital Computers, Information Representation, Computer Structure, Generic Computer, Number Systems, Arithmetic Operations, Conversion of Base, Decimal Codes (BCD), Alphanumeric Codes, Parity Bit.

2. Combinatorial Logic Circuits(8 hrs)

Binary logic and logic gates, Boolean Algebra, Standard Forms, (Min Term and Max Term, Sum of Product and Product of Sum Forms), Map Simplification (1,2,3,4 variable k-map, Do-not-Care Condition), Universal Gates (NAND, NOR, XOR), Integrated Circuits (Levels of Integration, Digital Logic Families, Positive and Negative Logic, Transmission Gates)

3. Combinatorial Logic Design(8 hrs)

Combinatorial Circuits, Design Issues, Analysis Procedure (Derivation of Boolean Function, Derivation of Truth table, Logic Simulation), Design Procedure (Code Converters), Decoders, Encoders, Multiplexers, Binary Adders, Binary Subtractor, Binary Multipliers, Decimal Arithmetic.

Syllabus

4. Sequential Circuits(8 hrs)

Definition, Latches (SR and D), Flip Flops (Master Slave, Edge Triggered), Sequential Circuit Analysis, (Input Equations, State Table, State Diagram, Analysis with JK Flip Flop), Sequential Circuit Design (Design Procedure, Finding State Diagram and State Tables,) Design with D Flip Flops, Design with JK Flip Flops, Flip Flop Excitation Table, Registers (Shift, Serial), Counters (Ripple, Synchronous Binary, BCD)

5. Memory and Programmable Logic Devices(8 hrs)

Random Access Memory, (Write and Read Operations, Timing Diagram, Properties of Memory, RAM IC's,) Array of RAM IC's, Read Only Memory, Programmable Logic Array, Programmable Logic array Devices, Memory Hierarchy, Locality of Reference, Cache Memory, Virtual Memory.

6. Instruction Set Architecture(8 hrs)

Concept, Operation Cycle, Register Set, Operand Addressing (Zero, One, Two, Three Address Instructions), Addressing Modes (Implied, Immediate, Register-Register, Direct, Indirect, Relative, Indexed), Instruction set Architecture, Data Transfer Instruction, Stack Instruction, Data Manipulation Instructions, Logical and Bit Manipulation Instruction, Program Control Instructions, Interrupts.

Syllabus

7. Central Processing Unit(8 hrs)

Data Paths and Operations, Register Transfer Operations, Microoperations, Bus Based Transfer The ALU, The CISC Computer (Instruction Set architecture, Data Path Organization, Microprogrammed Control Organization, Micro-program Structure, Micro Routines).

8. The Input Out Put and Communications(8 hrs)

Computer I/O, Peripherals (Key Board, Hard Disk, Graphics Display, I/O transfer Rates,), I/O interfaces(I/O Bus and interface units, Strobing, Handshaking,), Serial Communication(Asynchronous Transmission, Synchronous Transmission), Parity Interrupt (Daisy Chain), DMA (Controller and Transfer), I/O processors.

Evaluation

	Theory	Practical	Total
Sessional	50	-	50
Final	50	-	50
Total	100	-	100

- Attendance + Class performance: 20%
- Assignments: 20%
- Assesments:60%

References

Text Book :

1.Mano, M. M., Kime, R. C., Logic and Computer Design Fundamentals, 2E, Pearson Education Asia, ISBN81-7808-334-5

Reference Books :

1. Book by M. Morris Mano/ PHI

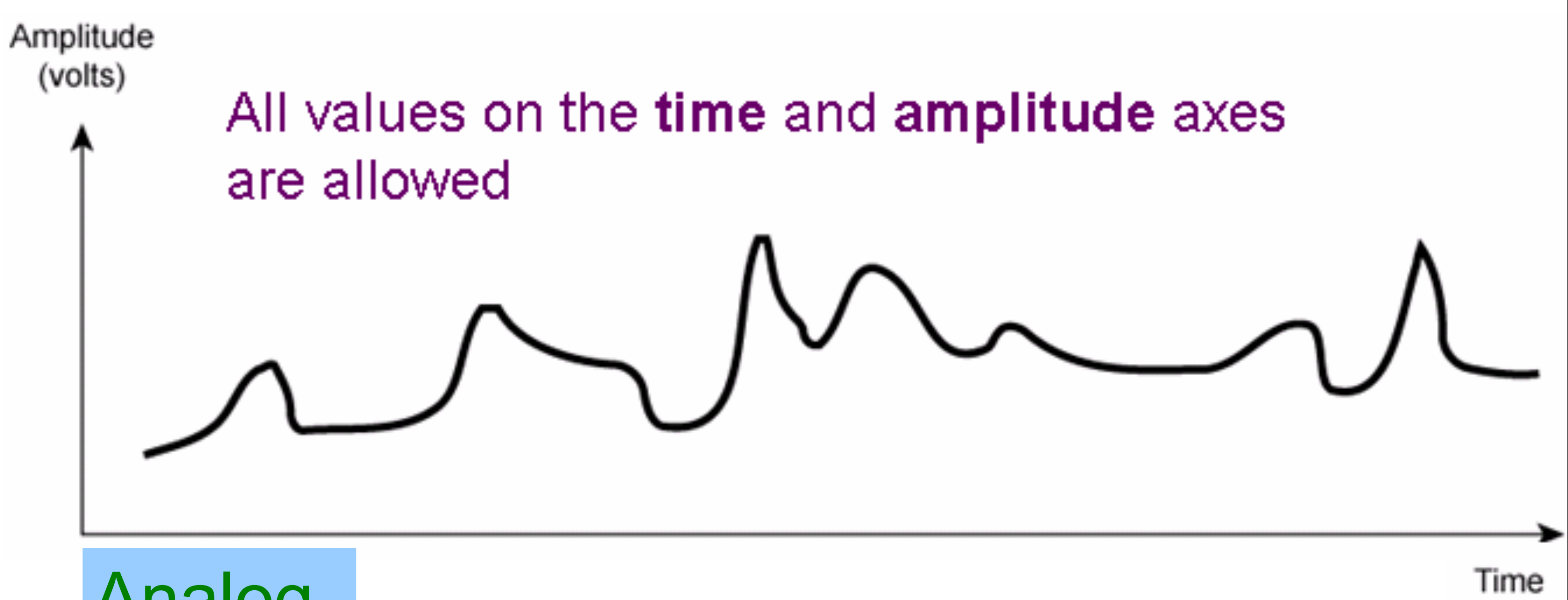
2. Book by Hamachar/Zakie/ McGraw Hill

Chapter 1
Digital Computers and Information

Basanta Joshi, PhD

1. Introduction to Digital Systems
Continuous Vs Discrete (Analog Vs Digital)

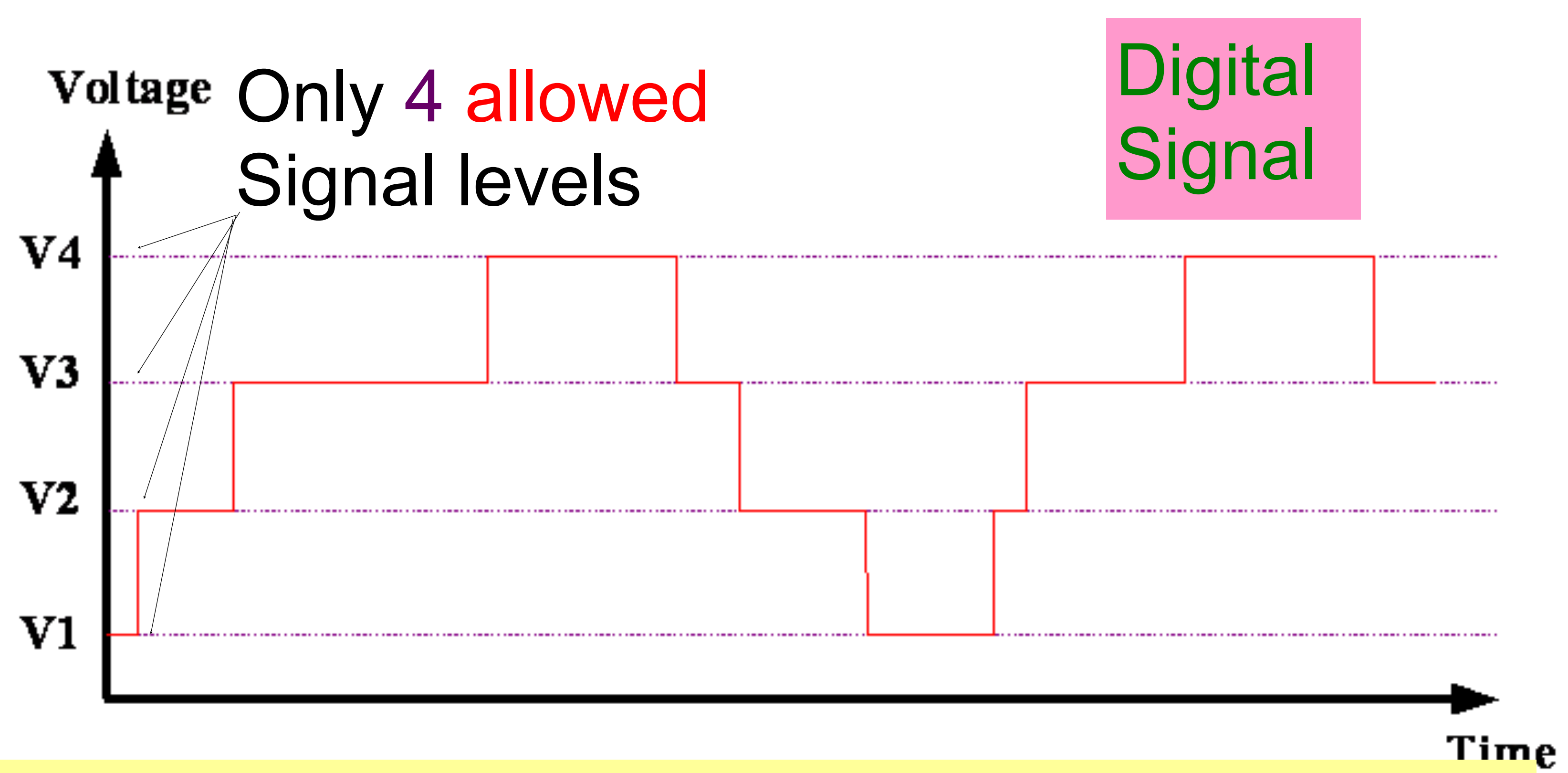
- We live in a predominantly Analog world
- Analog means Continuous (both in time and amplitude)
- Analog information exhibit smooth, gradual changes over time and assume a continuous (infinite) range of amplitudes
- **Examples:**
  - Earth's movement
  - Body temperature
  - Our speech



Analog Signal

Digital Information

- Digital ⇒ Discrete, Not continuous
- Digital information assume a limited (finite) set of "*Discrete*" values, not a continuous range of values
- Values change abruptly (not smoothly) by "*Jumping*" between values
- **Examples:**
  - The Alphabet
  - Position of a switch
  - DNA sequences (TAGC)
  - Energy levels of electrons in atoms



If we have only two signal levels ⇒ Binary signal
… So binary is a special case of digital.

Analog Vs Digital

- It is a lot ***easier*** to **design** digital systems than analog ones, Why?
    - Much simpler to deal with a limited set of values as inputs and outputs for the circuits
    - Greater tolerance to drift, noise low error rates

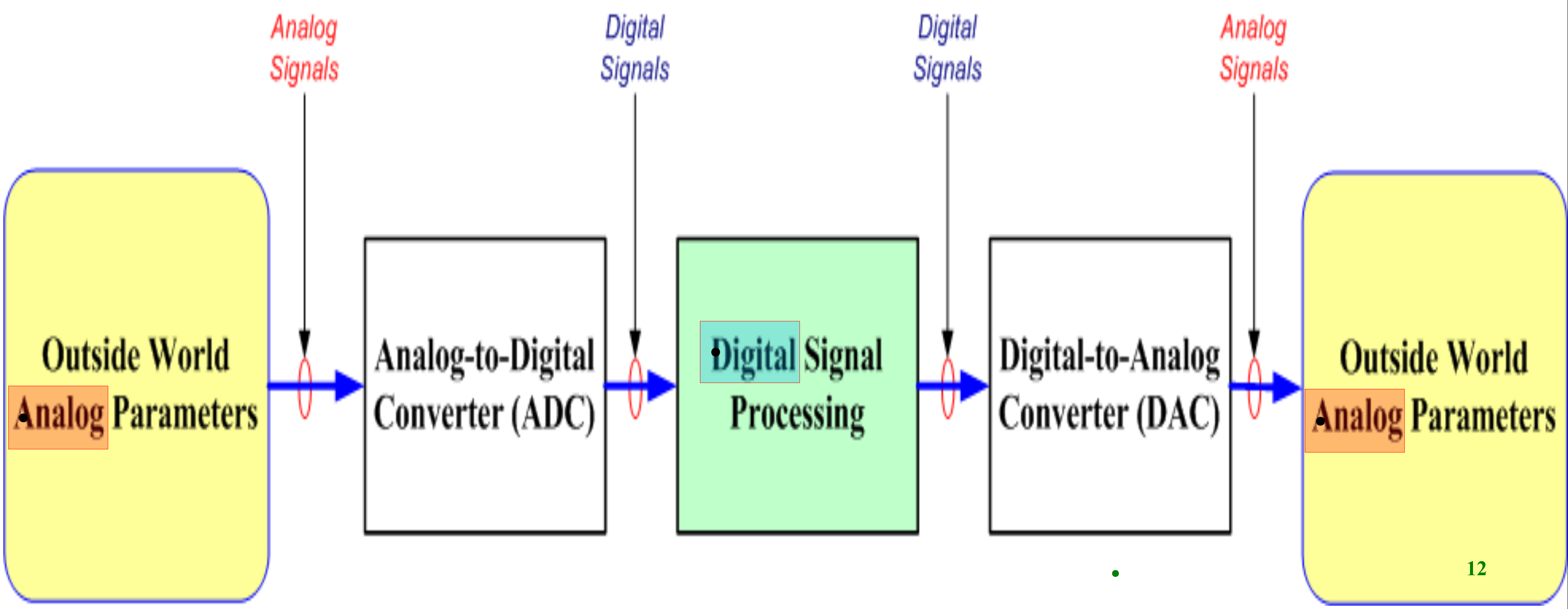
Also

- Digital circuits are more available, simpler and cheaper (VLSI)
- It is also more advantageous to store and communicate information digitally

- Dilemma here: Our natural world is mainly analog… but it is easier to process it digitally!

Digital Processing in the Real World

- Analog-to-digital-converters (ADC) are used to digitize raw analog inputs.
- Digital-to-analog-converters (DAC) are used to regenerate analog signals from their digitized form



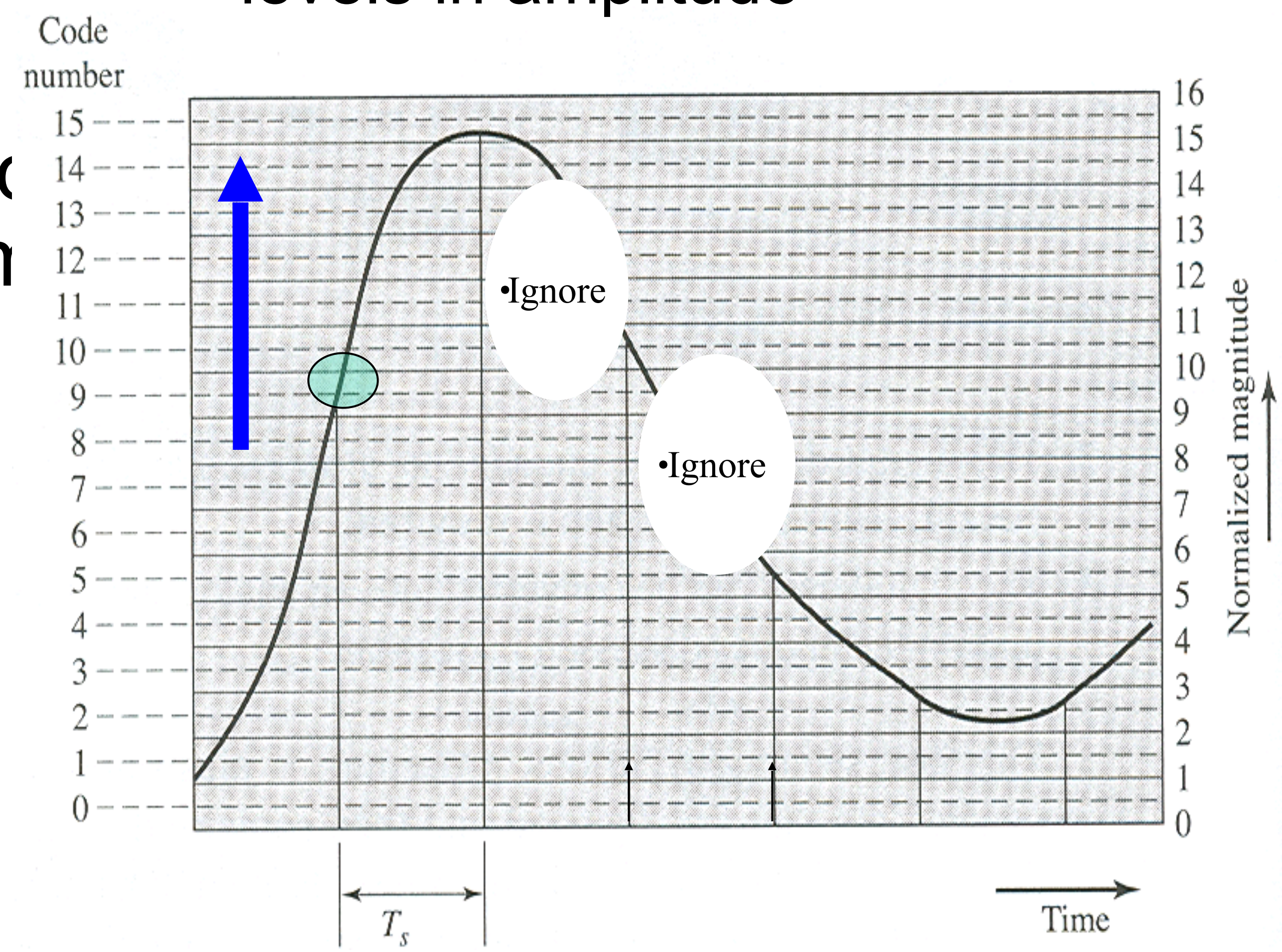
Digitization of Analog Signals

- Nowadays most processing, computing, and communication are done digitally
- **Solution:** Convert analog signals to digital parameters and process them digitally
- This requires two steps:

1. Sampling in time (impossible to handle the ∞ number of values existing on the time axis!). Ignore signal between samples

2. Quantization in amplitude (impossible to handle the ∞ number of values existing on the amplitude axis!). Approximate sample value to the nearest quantization level

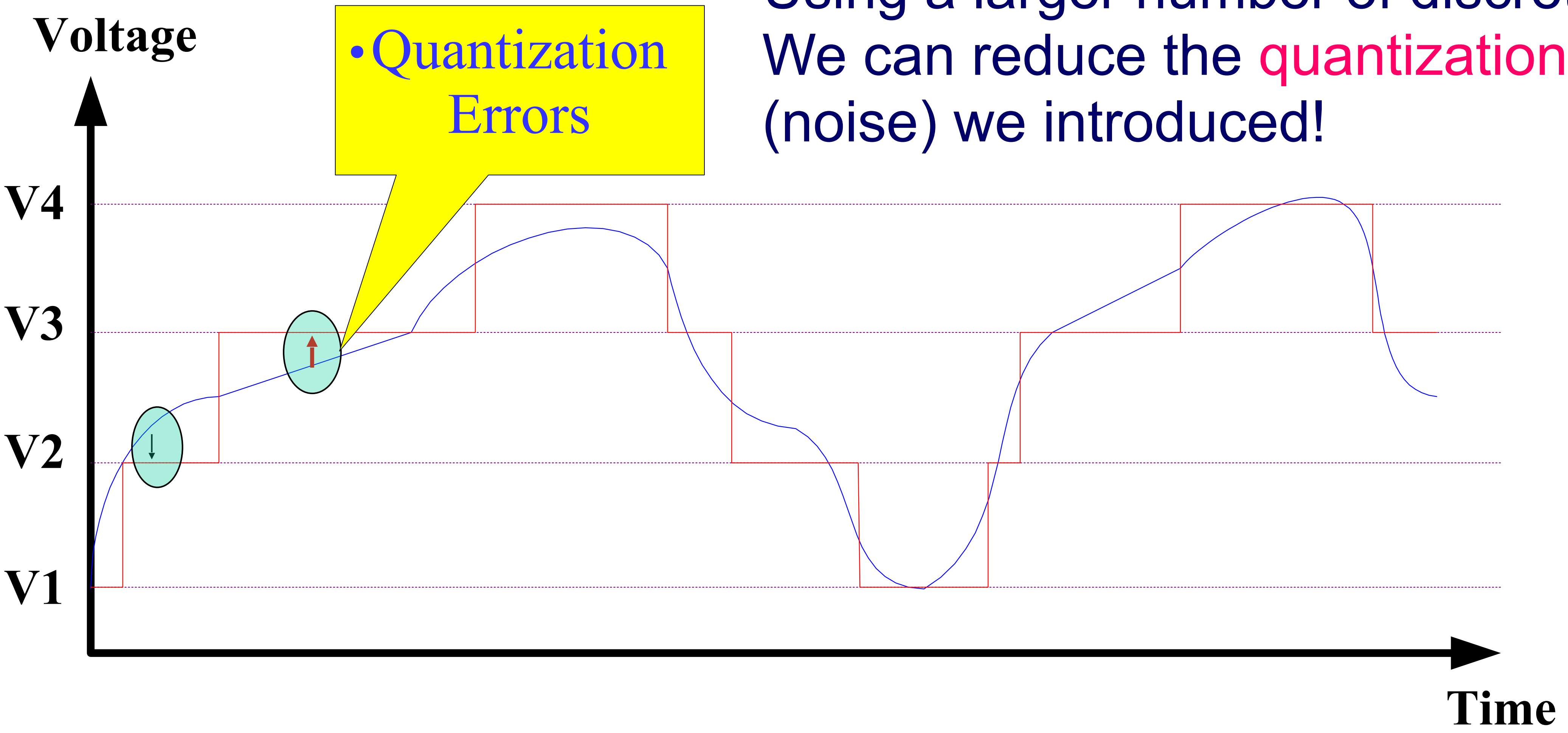
2. Quantization to discrete levels in amplitude



1. Sampling at discrete points in time

Example:
Amplitude Quantization: 4 discrete levels

Using a larger number of discrete levels We can reduce the quantization errors (noise) we introduced!



- Analog Signal levels are mapped to the <u>nearest</u> value among the set of discrete voltages $\in$ {V1, V2, V3, V4} <u>allowed</u> for the digital signal

Information Representation

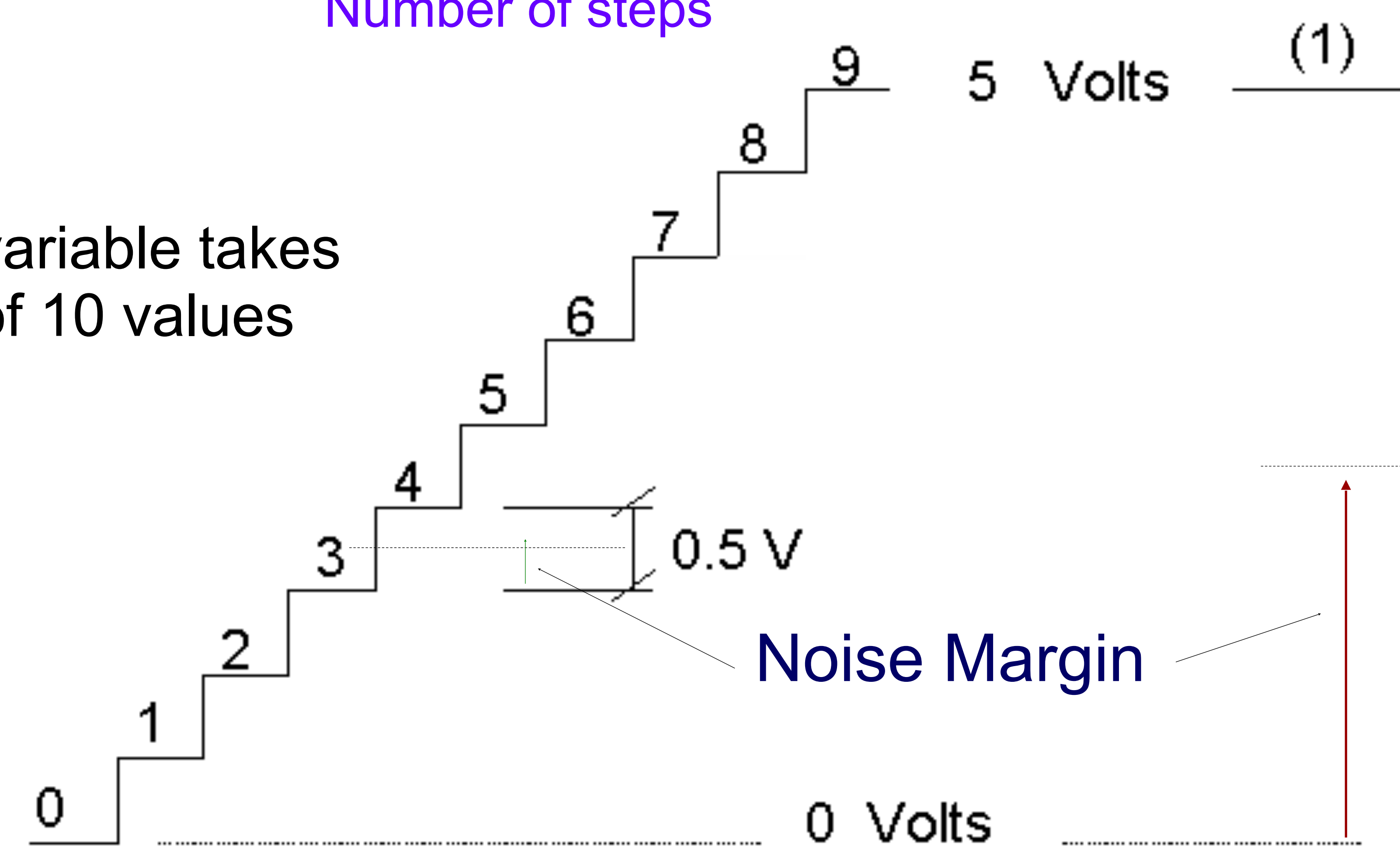
How to represent 10 discrete levels (0 to 9)?

Assume a 0 to 5 V range to represent the discrete quantization levels

	Direct 10-level Representation	Using Binary (2-level) Representation
Our circuits deal with:	Ten Signal levels	Two Signal levels (ON/OFF) Simpler, reliable Circuits
Noise Margin:	$(5/9)/2 \approx 0.25$ V	$(5/1)/2 = 2.5$ V Larger (better)

Number of steps

1 variable takes
1 of 10 values



Use n variables, each takes
1 of 2 values {0,1}

n binary digits (bits)

e.g. with n = 4 bits
6 is represented as 0110

Binary Values:
Abstract and Physical Representations

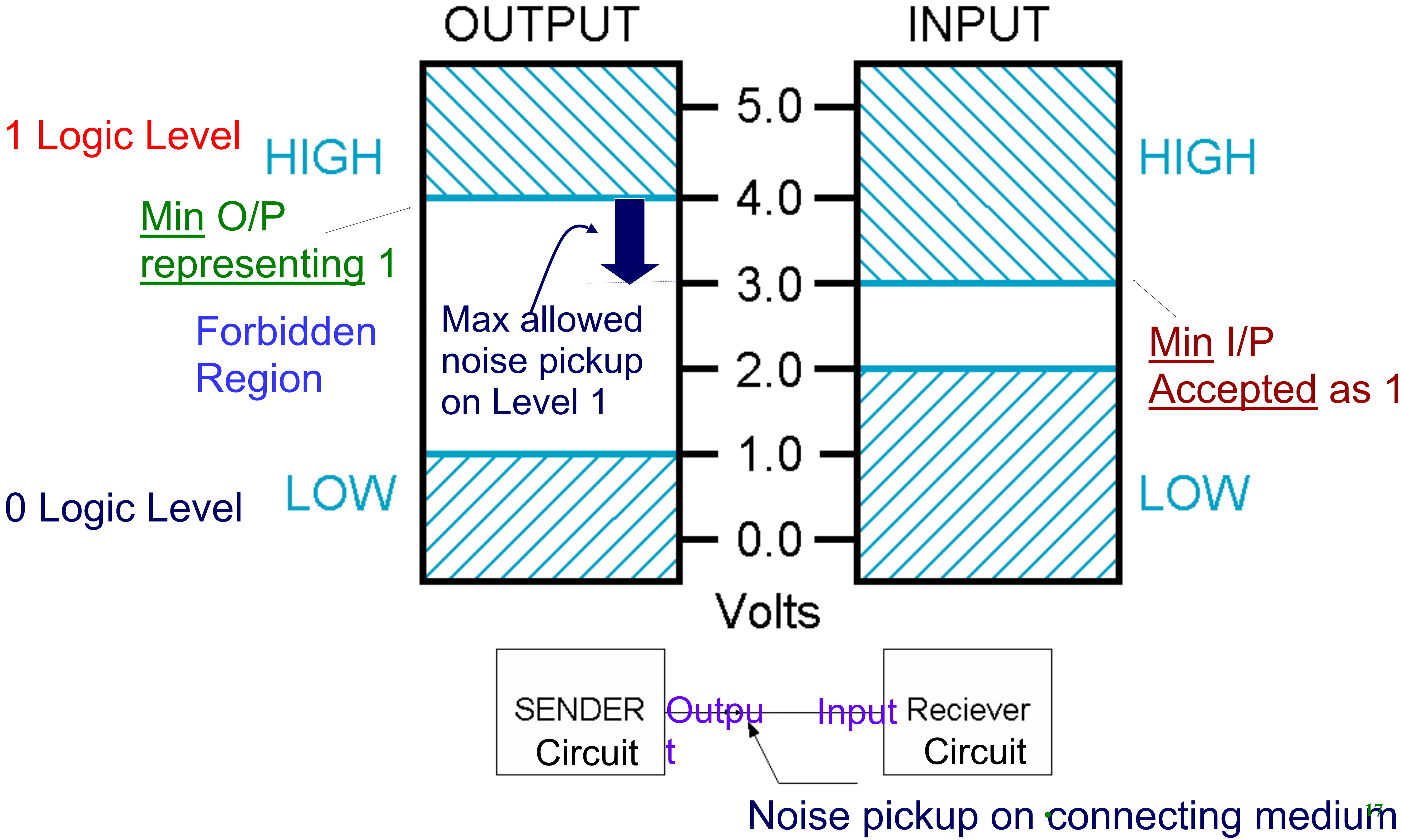
- Abstract Representations

  Digit Values: 0, 1
  Logic Levels: True, False
  Signal Levels: Low, High
  States: ON, OFF

Physical Representations

- **In an IC (e.g. a microprocessor): Voltage or Current**
- **On a Hard Disk: Magnetic Field Direction**
- **On a CD: Surface pits/Light interference**
- **In a Dynamic RAM: Electrical Charge**

Signal Voltage Levels: Noise Considerations



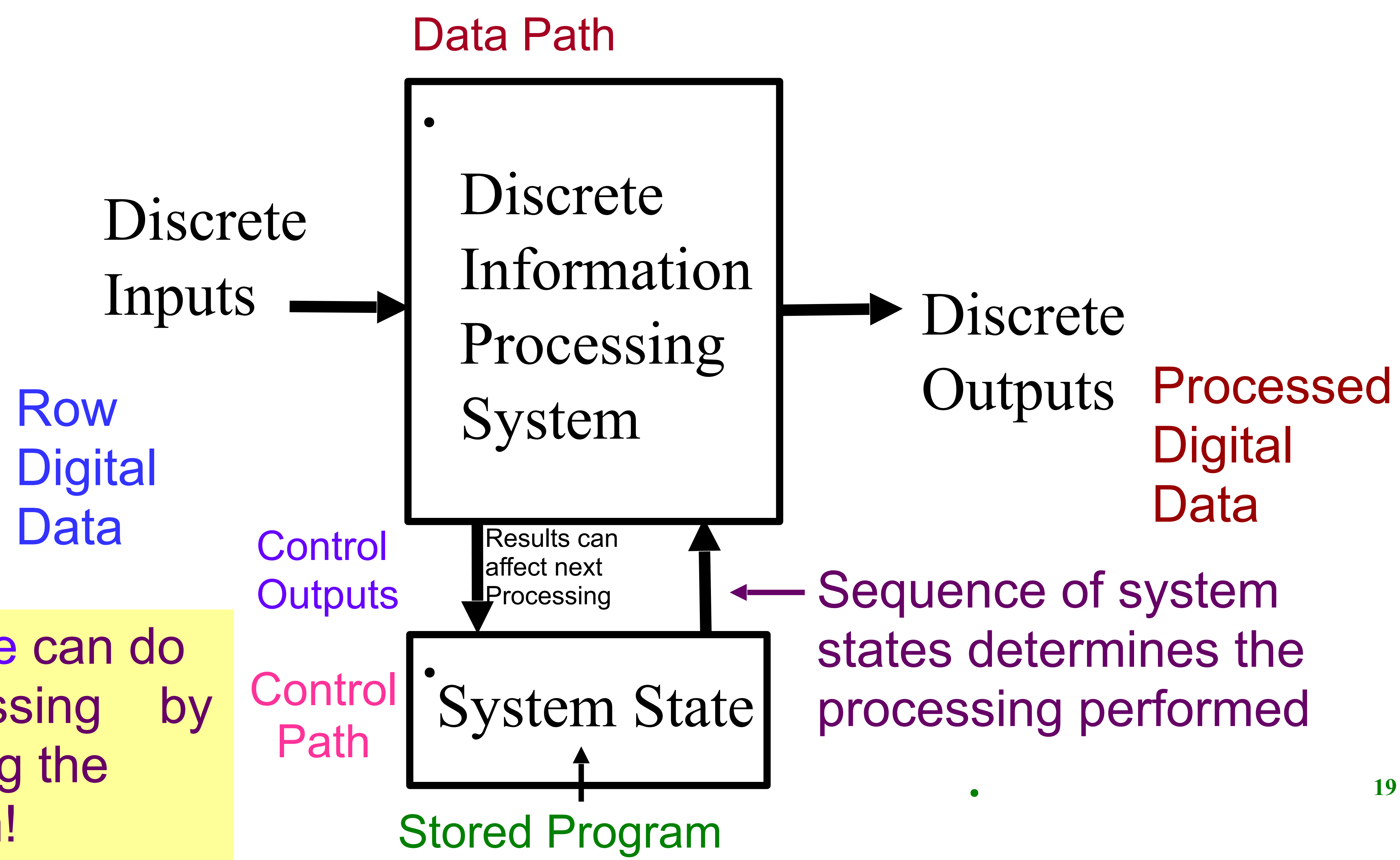
Types of Digital Systems

- No State Present
  - Response determined only by present inputs
    i.e. Output = Function (Inputs only)
  - Combinational Logic Circuits, e.g. adder
- A State Present
  - Response determined by present inputs <u>and</u> the present state of the system
    i.e. Output = Function (Present State, Inputs)
    and State (new) = Function (Present State, Inputs)
  - Sequential Logic Circuits, e.g. Counter

- Two Types of state-based systems:
- State updated at discrete times (e.g. at clock pulses)
  Synchronous Sequential System
- State updated at any arbitrary time
  Asynchronous Sequential System

Digital Processing System

Takes a set of discrete information <u>inputs</u> and discrete internal system information <u>(system state)</u> and generates a set of discrete information <u>outputs</u>.



Same hardware can do different processing by simply changing the stored program!

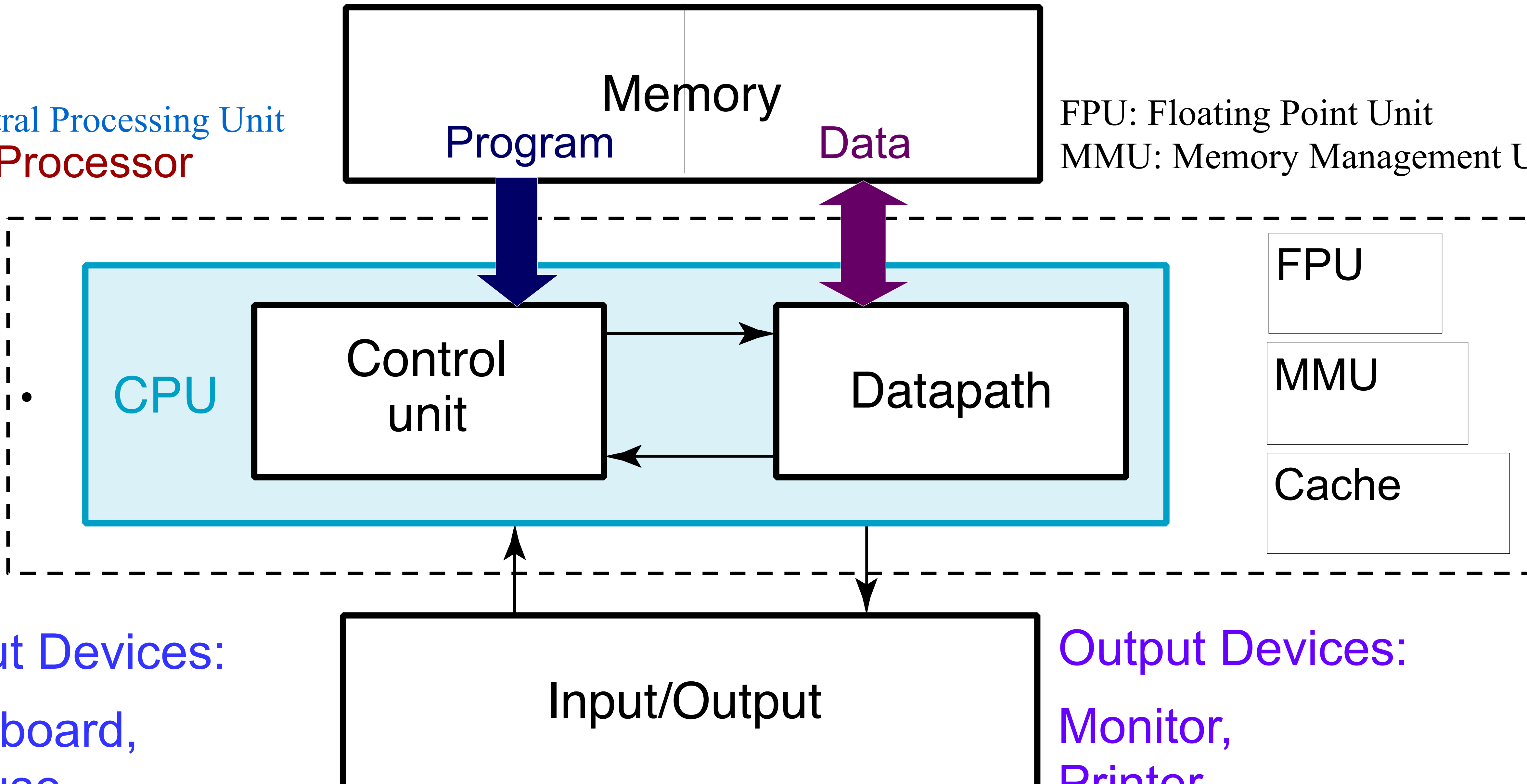
A Generic Digital Computer

CPU: Central Processing Unit

Processor

FPU: Floating Point Unit

MMU: Memory Management Unit



Input Devices:

Keyboard,
Mouse,
Scanner,
Microphone

Input/Output

Hard disk, Modem,
Network Card

Output Devices:

Monitor,
Printer,
Speaker

2. Number Systems Representation

- Positive radix, positional number systems
- A number with *radix (base) r* (or b) is represented by a string of digits (**n** in integer and **m** in fraction):

$$\boldsymbol{A_{n-1}A_{n-2}\cdots A_1A_0 \,.\, A_{-1}A_{-2}\cdots A_{-m}}$$

Integer Part

Fraction Part

in which $\mathbf{0} \leq A_i < r$ and "." is the *radix point*.

- The string of digits represents the power series:

$$(\text{Number})_r = \left(\sum_{i=0}^{i=n-1} A_i \, r^i\right) + \left(\sum_{j=-1}^{j=-m} A_j \, r^j\right)$$

+ ive exponents (Integer Portion)

- ive exponents (Fraction Portion)

Examples

- Decimal (base 10):

Radix Powers

2 1 0 -1 -2

$$(724.56)_{10} = 7 \times 10^2 + 2 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2}$$

$$= 700 + 20 + 4 + 0.5 + 0.06 = 724.56$$

- (base 5):

Radix Powers

2 1 0 -1

$$(312.4)_5 = 3 \times 5^2 + 1 \times 5^1 + 2 \times 5^0 + 4 \times 5^{-1}$$

$$= 75 + 5 + 2 + 0.8 = 82.8$$

Number Systems – Examples

For these 3 systems, Radix is a Power of 2

System	General	Decimal	Binary	Octal	Hexadecimal
Radix (Base)	r	10	2	8	16
Digits: 0 to (r-1)	0, 1, …, (r – 1)	0, 1, …, 9	0, 1	0,…, 7	0, …, 15
0	r0	1	1		
1	r1	10	2		
2	r2	100	4		
Integer 3	r3	1000	8		
Power of 4	r4	10,000	16		
Radix 5	r5	100,000	32		
-1	r -1	0.1	0.5		
-2	r -2	0.01	0.25		
-3	r -3	0.001	0.125		
-4	r -4	0.0001	0.0625		
-5	r -5	0.00001	0.03125		
Fraction					

Commonly Used Bases

Name	Radix	Digit takes values:	Example
Decimal (d)	10	0,1,2,3,4,5,6,7,8,9	2786.94
Binary (b)	2	0,1*	1011.01
Octal	8	0,1,2,3,4,5,6,7	2764.35
Hexa-Decimal (H)	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F 10,11,12,13,14,15	57AC.F4D

*The Binary digit is called a bit

Bits take the values 0,1 only (OFF/ON) ⇒ can be manipulated with Simple, Reliable Circuits

Integers (0 – 16) Represented in Various Bases

0-15 : 4 2 1 digits

More Concise

Decimal (Base 10)	Binary (Base 2)	Octal (Base 8)	Hexadecimal (Base 16)
00	00000	00	00
01	00001	01	01
02	00010	02	02
03	00011	03	03
04	00100	04	04
05	00101	05	05
06	00110	06	06
07	00111	07	07
08	01000	10	08
09	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10

The larger the Radix the fewer the digits needed to represent a given number

LS digit goes through all possible Combinations, then the more significant digit, etc.

Octal & Hexa offer a more concise way of representing long bit (binary) sequences in computer systems

Binary Numbers: Special Powers of 2

2^{10} (1024) is Kilo, denoted "K"

2^{20} (1,048,576) is Mega, denoted "M"

2^{30} (1,073, 741,824) is Giga, denoted "G"

2^{40} (1,099,511,628) is Tera, denoted “T"

Positive Powers of 2

- Powers of 2

Exponent	Value	
0	1	
1	2	
2	4	
3	8	
4	16	
5	32	
6	64	
7	128	
8	256	
9	512	
10	1024	1 K

Exponent	Value	
11	2,048	2 K
12	4,096	4 K
13	8,192	8 K
14	16,384	16 K
15	32,768	= 215 = 25 x 210 = 32 K
16	65,536	64 K
17	131,072	128 K
18	262,144	256 K
19	524,288	512 K
20	1,048,576	1 M
21	2,097,152	2 M