THE UNIVERSITY OF TEXAS AT DALLAS

CS4341 Digital Design & Computer Design

Lecture Notes 1

Omar Hamdy
Assistant Professor
Department of Computer Science

About Your Instructor (Me!)

- Assistant Professor of Computer Science at UTD Since ..... August 2021
- Adjunct Professor at number of universities including FIT, UMUC, U. Cumberlands
- PhD in Computer Engineering from University of Victoria, BC, Canada
- Industry experience in different industries and with different companies like Verizon, Avaya, IBM
- Research in Digital Biometrics and Forensics
- Like fishing, astronomy and soccer
- Best way to contact me is by EMAIL (omar.hamdy@utdallas.edu)

What Do We Want to Achieve?

- To design and build better computer systems that can help us:
  - Solve Problems
  - Gain Insight

- How do computers do that?

Orchestrating Electrons!!

Transformation Hierarchy (Abstractions)

Algorithms:
Step-by-step procedure that is guaranteed to terminate where each step is precisely stated, and can be carried out by a computer

SW/HW Interfaces
Instruction Set Architecture (ISA)
- Operate on data
- Move data
- Change control flow

Logic Gates
Building blocks of the microarchitecture

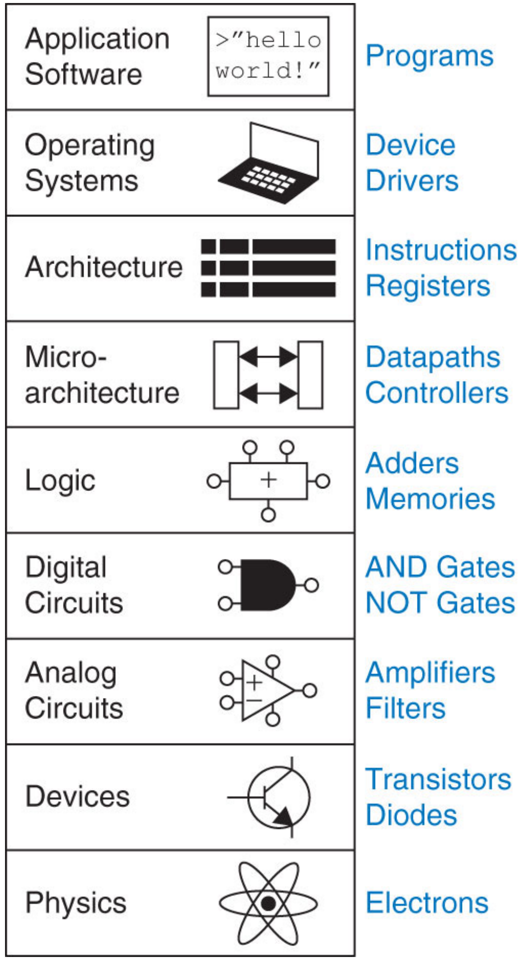
Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro Architecture
Logic Gates
Devices
Electrons

System Software:
Runtime System: VM, OS, etc

Micro Architecture
Implementation of the ISA

Devices
Physical electronic circuit of the logic gates . Example Transistors

Textbook Abstractions Reference Model

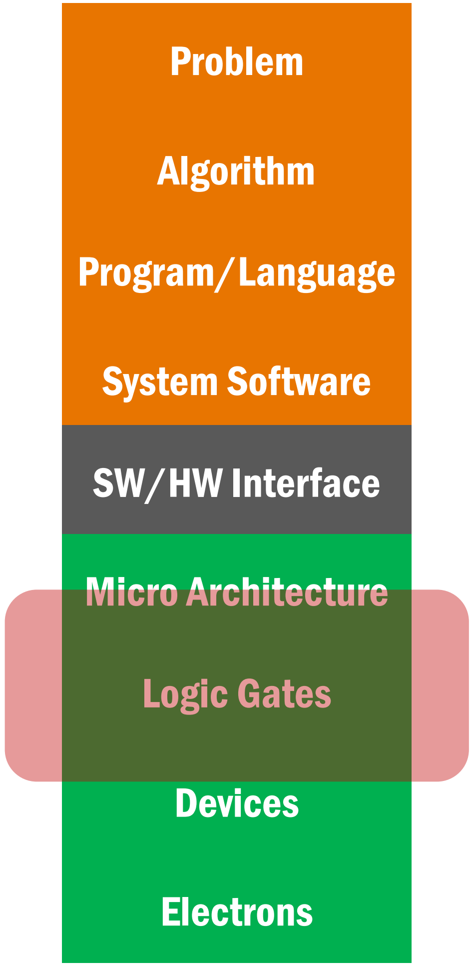


Logic Gates

- Problem
- Algorithm
- Program/Language
- System Software
- SW/HW Interface
- Micro Architecture
- Logic Gates
- Devices
- Electrons

- Combinational Logic (Ability to perform an operation):
  - Uses basic gates: AND/NAND, OR/NOR, Buffer/NOT, XOR/XNOR
  - Used for data operations: Addition, Subtraction, Multiplication, Division
  - Time is not a parameter
- Sequential Logic (Ability to control an operation):
  - Allows for controls and flow branching
  - Used to implement memory systems
  - Time is a parameter
- Combining combinations (operations) and sequential (control) we create ALUs, then core computer processors.

Class Focus (CLOs)



- **Ability to analyze, minimize and design gate-level combinational logic circuits using Boolean algebra and 3 and 4 variable Karnaugh Maps**
- **Ability to analyze and design simple synchronous sequential circuits**
- **Ability to analyze, design and utilize digital logic components such as adders, multiplexers, decoders, registers, and counters**
- **Ability to understand RAM and ROM memory components, and utilize these in digital logic design**
- **Ability to design computer components such as Arithmetic-Logic-Unit (ALU) and data path**
- **Ability to understand the basics of hardware description languages such as Verilog or VHSIC Hardware Design Language (VHDL)**

Design Objectives & Goals

Purpose is to design and build architectures that are:

1. Reliable, secure and safe
2. Energy efficient
3. Predictable and low latency
4. Specialized for key industries and domains (AI/ML, Genomics, VR, etc)

Evaluation Criteria for the Design

- Functionality (meets the specifications)
- Reliability
- Space Requirements
- Cost
- Expandability
- Comfort Level (User Friendly)
- Security
- Others

Analog vs. Digital Systems

- Analog means continuous. It expresses the smooth gradual change of parameters.
- We live in an analog world
- Digital means to operate using parameters that have limited set “discrete” values.
- Digital parameters change by “jumping” from one allowed value to the other.

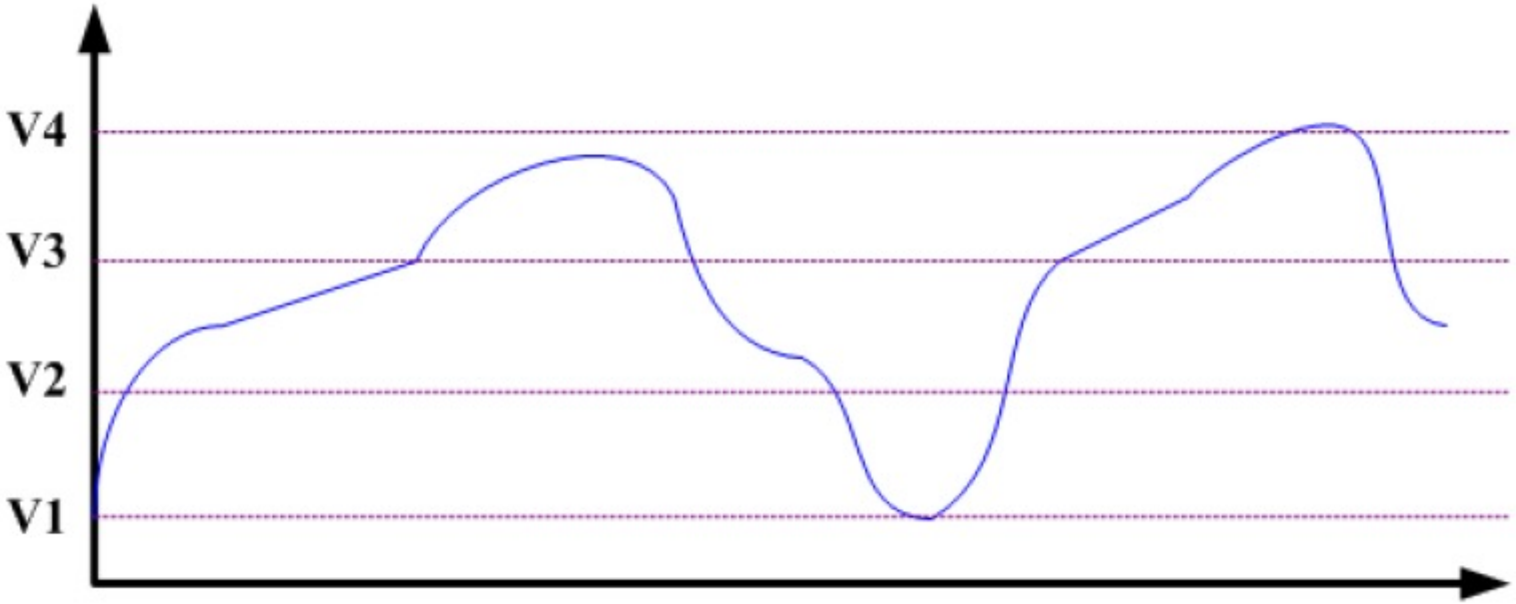
Digitization

- Since we live in an analog world, it is common to convert analog into digital.
- It is easier to work with digital than analog:
  - Easier processing and transmitting.
  - Better storage (compactly)
  - Less noise (unwanted voltage) impacted

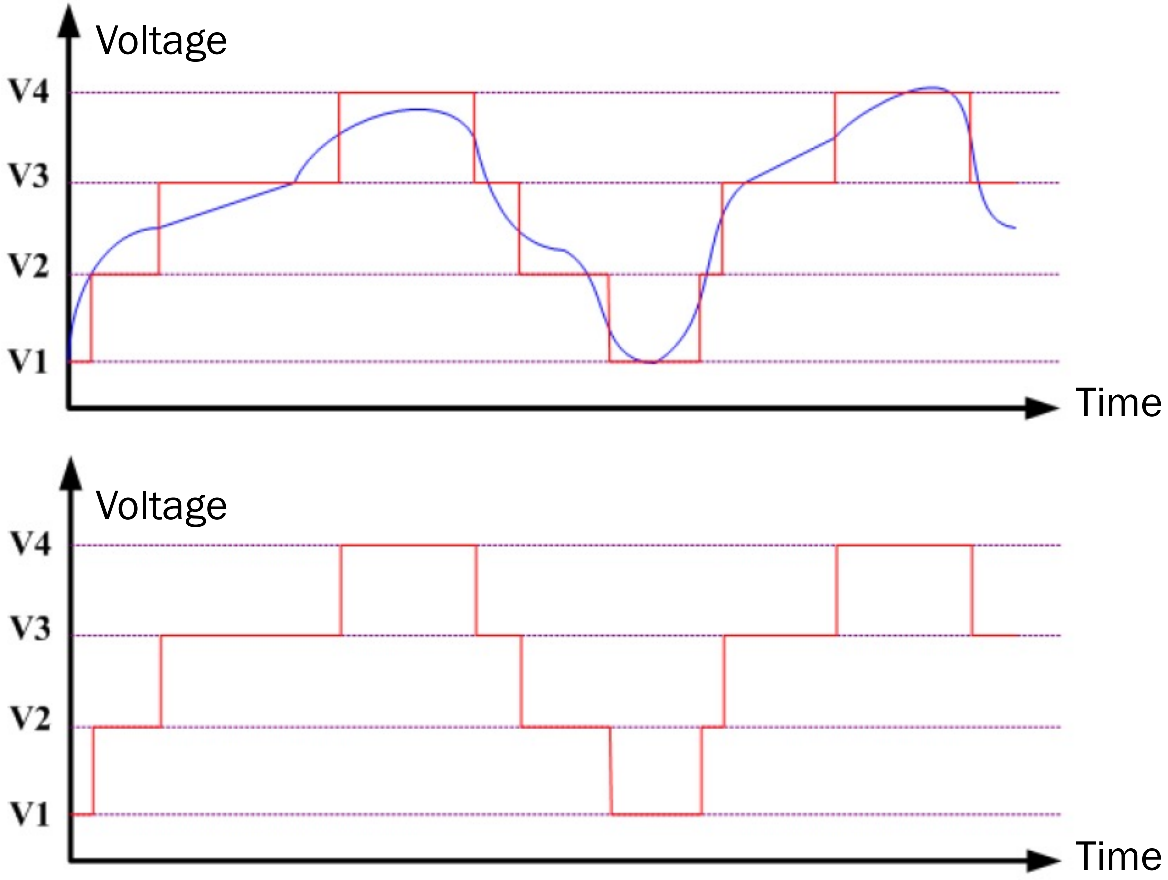
Any disadvantage then?!

Voltage Digitization

Example: We need to digitize an analog voltage signal using 4 voltage levels.



Voltage Digitization



Digitization comes at the cost of accuracy

Representing Information

A computer is a digital system. Therefore, information is represented, processed, and stored in digital format. How?

- Using electric voltage and charge:
  - Used in processors, digital circuits, memory
  - High voltage = 1, Low voltage = 0
- Using magnetic field:
  - Used in magnetic disks
  - Magnetic polarity represents 1 or 0
- Using light:
  - Used in optical disks
  - Surface pit indicates 1 or 0

Number Representation

Consider the following two examples:

(Hello)$_{\text{English}}$ = (Salut)$_{\text{French}}$ = (สวัสดี)$_{\text{Thai}}$

They all mean the same thing but in different “language system”.

Different language systems may or may not share the same alphabet “symbols”

(Gift)$_{\text{English}}$ =  (Gift)$_{\text{German}}$ =  (Gift)$_{\text{Norwegian}}$ = 

Sometimes same words have different meanings in the different “language systems”

Positional Number System

*A positional number system is composed of a **base (radix) r** > 1, and a set of **r** digits.*

Example: Decimal System:

r = 10 or called base 10

Digit set: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Are there other positional number systems (or bases)?

Positional Number System

Binary System:

r = 2 or base 2

Digit set: {0, 1}

Octal System:

r = 8 or base 8

Digit set: {0, 1, 2, 3, 4, 5, 6, 7}

Hexadecimal System:

r = 16 or base 16

Digit set: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Positional Number System

How do we count in Decimal number system?

- Starting at the right most digit, we increment till we reach the base last digit
- We then reset that digit to 0, and increment the digit to its left by 1.

Count 1 to 15 in binary, octal and hexadecimal

r_{10}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r_2	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111
r_8	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
r_{16}	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Integer Representation

Any integer $(N)_r$ can be represented by a finite sequence of concatenated digits:

$(N)_r = (b_{n-1}, b_{n-2}, \ldots, b_1, b_0)_r$ Where:

- Any b_i is an integer such that $0 \leq b_i \leq r-1$
- n is referred to the **length** of the digit string

The digit positions give different meanings (values) to the digits they contain. For example 44: the first 4 means something different from the second 4.

Numerical Values

How to calculate the numerical value of a digit string

In decimal, the numerical value of 972 is calculated as:

- 900 + 70 + 2
- Expressed as: 9x100 + 7x10 + 2x1
- Expressed as: $9 \times 10^2 + 7 \times 10^1 + 2 \times 10^0$

Can you link between the multiplicand 10 and the base?

Can you link between the power and the digit position?

Numerical Values

An integer N of length n and base r is represented as:

$(N)_r = (b_{n-1}, b_{n-2}, \ldots, b_1, b_0)_r$

And has a numerical value of:

$b_{n-1}r^{n-1} + b_{n-2}r^{n-2} + \ldots + b_1r^1 + b_0r^0$

Can be represented using the summation formula:

$$\sum_{i=0}^{n-1} b_i r^i$$

To Do List

- Go over the course syllabus, especially the attendance policy
- Review lecture notes
- Read chapter 1 until 1.4.5