B38DB: Digital Design and Programming
Introduction & Overview

Mustafa Suphi Erden

Heriot-Watt University
School of Engineering & Physical Sciences
Electrical, Electronic and Computer Engineering
Room: EM 2.01
Phone: 0131-4514159
E-mail: m.s.erden@hw.ac.uk

Course Structure

- **Lecturer: Mustafa Suphi Erden**
  - EM 2.01, m.s.erden@hw.ac.uk
- Lecture:
  - Monday (Online): 10:00-11:00
- Laboratory:
  - Friday (Online): 10:00-12:00
- Tutorial:
  - Thursday (Online): 9:00-10:00
- Tutors:
  - Nathan Western [NW] (nw29@hw.ac.uk)
  - Borja Marin [BM] (bm86@hw.ac.uk)

Aim of Course

- **To develop knowledge and skills in digital design and programming targeted at implementation**
- *Theoretical* learning
  - Digital circuits (as in computers) and building blocks
  - Combinational logic, sequential logic, datapath components such as adders, multipliers, arithmetic logic units (ALUs), and finally register transfer level (RTL) design
  - Simple data-path (core of a processor)
- *Practical* learning
  - Design and implementation of digital circuits using a logic simulator (LabSim)

Course Assessment

- **Course-Work (100%)**:

  1) Lab-1: 15% – Week 3
  2) Lab-2: 15% – Week 5
  3) Lab-3: 20% – Week 8
  4) <u>Take-Home Test:</u> 30% – Week 10

  6) Lab-4: 20% – Week 11

Lab report and test submissions will be through Turnitin in Vision.

Lab reports are always **due in two weeks** after the lab.

Labs

- Laboratory Worksheets will be available in Vision in advance of the lab hours.
- **There will be dedicated online lab sessions where tutors will be available to help you and answer your questions.**
- Laboratory Worksheets will have parts to fill in and report. You need to fill these in and submit the file through Turnitin in Vision.
- Lab work are manageable for you to do on your own any time you wish. You do not need to wait for the lab sessions to do the lab work. Lab sessions are just to support you in case you have any problems.
- Submission Deadline: **Two weeks after** each lab session.

Planning

[B38DB] Edinburgh Campus Planning – 2021
Virtual Classroom

Week/ Date	**Learning Units** **<u>Chapters</u>** **Videos: V1-V15** **Lecture Slides: LS1-LS14**	**Videos and Lecture Slides**	**Live Online Lecture Sessions** *(Videos should be watched prior to the lecture!)*	**Assessments (Live Online Lab Sessions)**
Week 1 13th Sept	Introduction – Digital Signals Combinational Logic Design – Logic Gates	V1, V2 LS1, LS2	<u>Lecture:</u> Monday, 10am-11am	
Week 2 20th Sept	Decoders, Multiplexers Combinational Logic Design – Karnaugh Maps	V3, V4 LS3, LS4	<u>Lecture:</u> Monday, 10am-11am <u>Tutorial:</u> Thursday, 9am-10am	<u>Lab 0</u> (0%) Self-Study: LogiSim Intro and Videos Friday: 10am-12pm
Week 3 27th Sept	Sequential Logic Design – Flip Flops, Finite State Machines	V5, V6 LS5, LS6	<u>Lecture:</u> Monday, 10am-11am <u>Tutorial:</u> Thursday, 9am-10am	<u>Lab 1</u> (15%) Decoders and Multiplexers Friday: 10am-12pm
Week 4 4th Oct	Datapath Components – Registers	V7 LS7	<u>Lecture:</u> Monday, 10am-11am <u>Tutorial:</u> Thursday, 9am-10am	
Week 5 11th Oct	Sequential Logic Design – Controller Design	V8 LS8	<u>Lecture:</u> Monday, 10am-11am <u>Tutorial:</u> Thursday, 9am-10am	<u>Lab 2</u> (15%) Multi Function Registers Friday: 10am-12pm
Week 6 18th Oct	Consolidation Week			

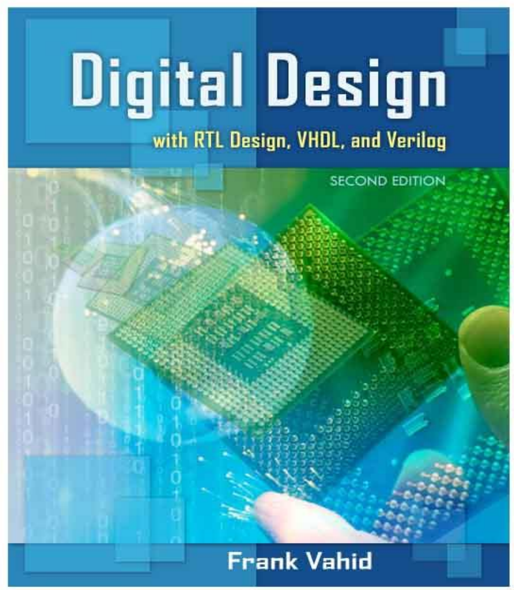
Planning

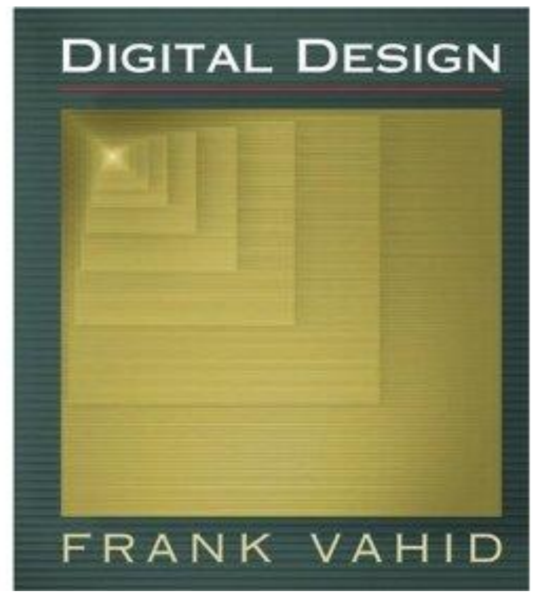
[B38DB] Edinburgh Campus Planning – 2021
Virtual Classroom

Week/ Date	**Learning Units** **<u>Chapters</u>** **Videos: V1-V15** **Lecture Slides: LS1-LS14**	**Videos and Lecture Slides**	**Live Online Lecture Sessions** *(Videos should be watched prior to the lecture!)*	**Assessments (Live Online Lab Sessions)**
Week 7 25th Oct	Sequential Logic Design – Controller Examples; Adders and Comparators	V9, V10 LS9, LS10	<u>Lecture:</u> Monday, 10am-11am <u>Tutorial:</u> Thursday, 9am-10am	
Week 8 1st Nov	Datapath components – Subtractors and Counters	V11 LS11	<u>Lecture:</u> Monday, 10am-11am <u>Tutorial:</u> Thursday, 9am-10am	<u>Lab 3</u> (20%) Adders Friday: 10am-12pm
Week 9 8th Nov	Datapath components –ALU, Register file	V12 LS12, LS13	<u>Lecture:</u> Monday, 10am-11am <u>Tutorial:</u> Thursday, 9am-10am	
Week 10 15th Nov	High-Level State Machines	V13 LS14	<u>Lecture:</u> Monday, 10am-11am <u>Tutorial:</u> Thursday, 9am-10am	<u>Take-Home Test</u> (30%) Friday: 10am-12pm
Week 11 23rd Nov	Simple Data Path; Simple Data Path Processing	V14 V15 LS14	<u>Lecture:</u> Monday, 10am-11am <u>Tutorial:</u> (Test Solutions) Thursday, 9am-10am	<u>Lab 4</u> (20%) ALU Friday: 10am-12pm
Week 12 29th Dec	Verilog Intro		<u>Lecture:</u> Monday, 10am-11am	

Recommended Main Textbooks

- Frank Vahid, “Digital Design”, 2nd edition, Wiley, 2011.
- Frank Vahid, “Digital Design”, 1st edition, Wiley, 2007. http://www.cs.ucr.edu/~vahid/dd/
- Slides from these textbooks are used with permission for educational use with this course.



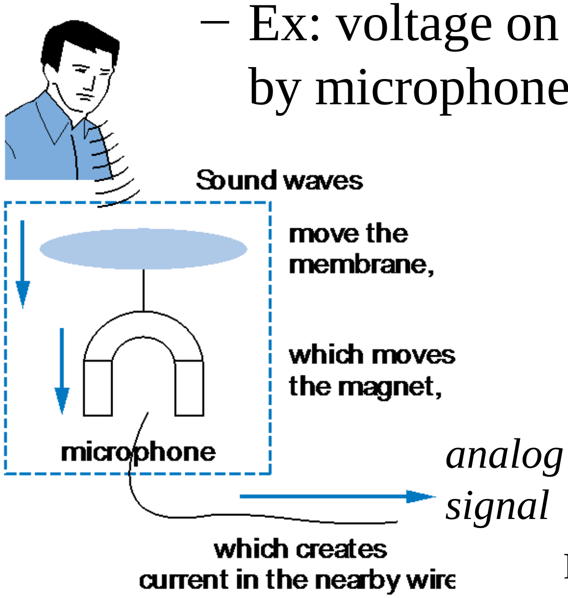


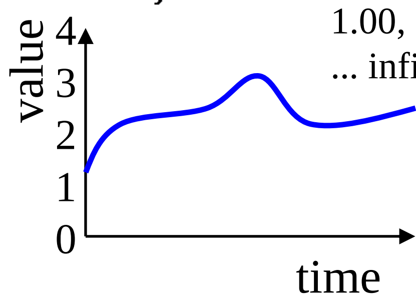
Why Study Digital Design?

- Electronic devices becoming digital
- Analogue systems → Digital Systems
  - Shrinking of components
  - More capable components (chips)
  - Better devices: Better sound recorders, cameras, cars, cell phones, medical devices,...
  - New devices: Video games, PDAs, ...
- Embedded systems (Wikipedia)
  - "A **computer system** with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints"
  - "low power consumption, small size, rugged operating ranges, and **low per-unit cost**"
  - "limited processing resources, which make them significantly **more difficult to program** and to interface with" (Wikipedia)

What Does "Digital" Mean?

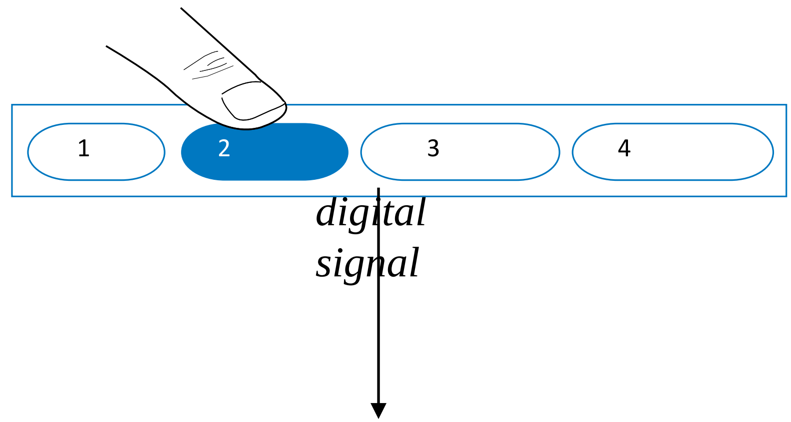
- Analog signal
  - Infinite possible values
    - Ex: voltage on a wire created by microphone

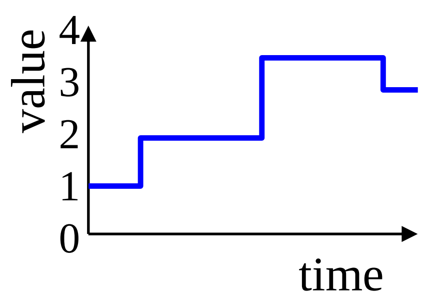




Possible values:
1.00, 1.01, 2.0000009,
... infinite possibilities

- Digital signal
  - Finite possible values
    - Ex: button pressed on a keypad

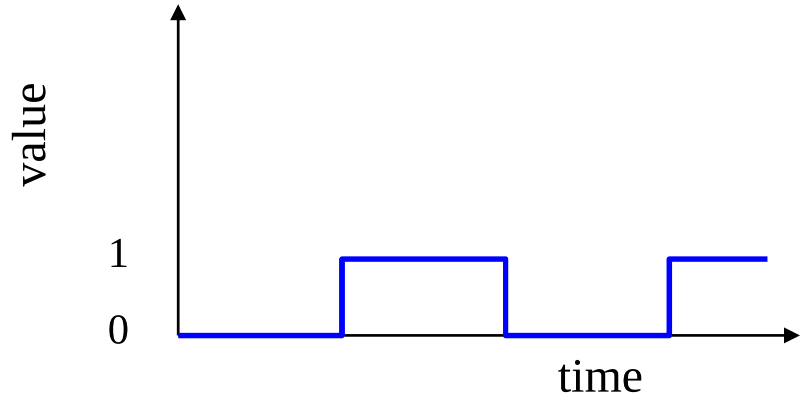




Possible values:
0, 1, 2, 3, or 4.
That's it.

Digital Signals with Only Two Values: <u>Binary</u>

- ***Binary*** digital signal – only *two* possible values
  - Typically represented as **0** and **1**
  - One *b*inary dig*it* is a ***bit***
  - We'll only consider *binary* digital signals
  - Binary is popular because
    - Transistors, the basic digital electric components, operate using *two* voltages
    - Storing/transmitting one of *two* values is easier than three or more (e.g., loud beep or quiet beep, reflection or no reflection)

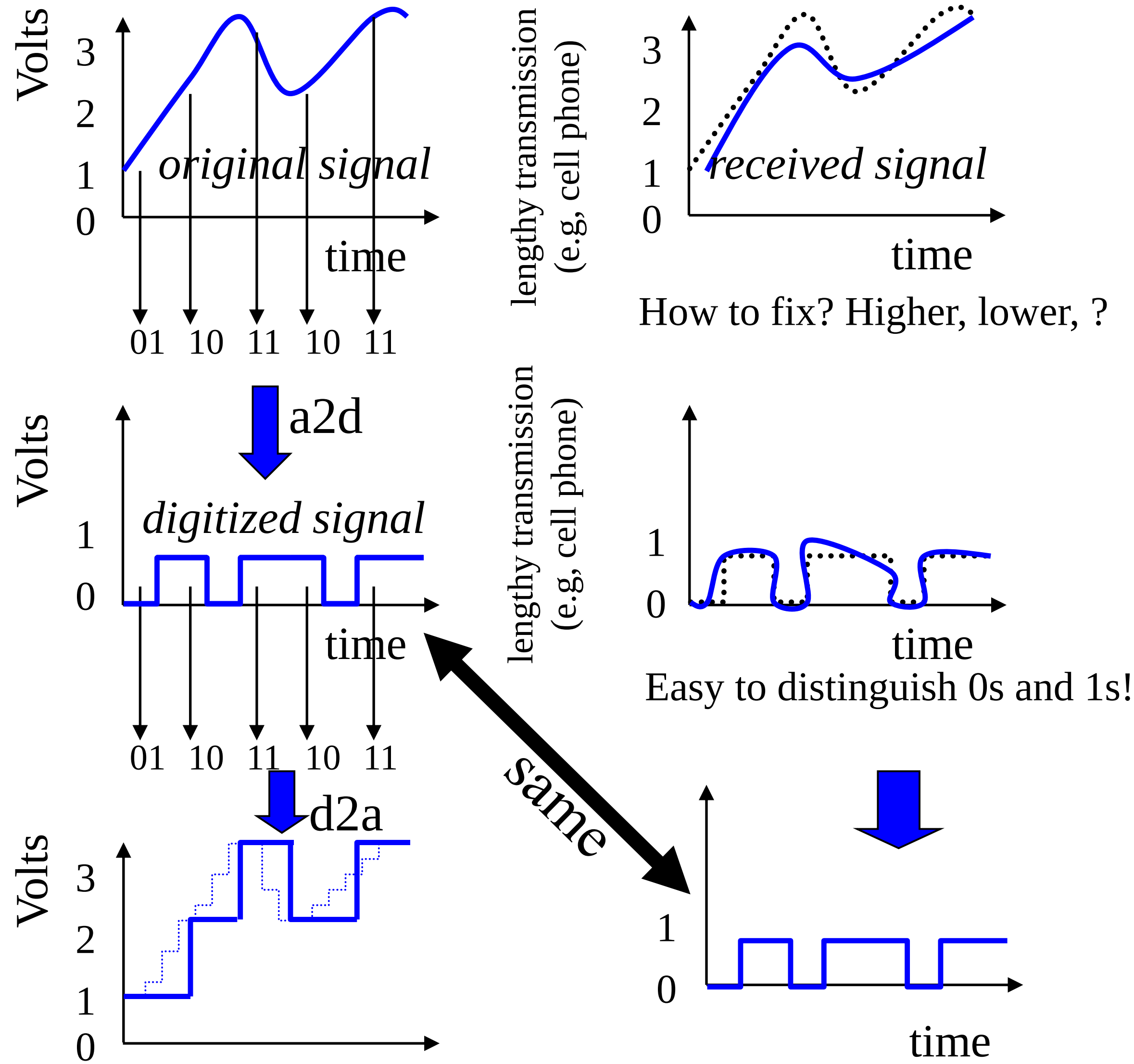


Benefit of Digitalization

- Analog signal (e.g., audio) may loose quality
  - Voltage levels **not perfectly**
    - Saved
    - Copied
    - Transmitted
  - Digitized version **near-perfectly**
    - Saved
    - Copied
    - Transmitted
  - How come?
  - Voltage levels still not be kept perfectly
  - But we can distinguish **0**s from **1**s

Example of Digitization Benefit

Let bit encoding be:
- 1 V: “01”
- 2 V: “10”
- 3 V: “11”



Higher sampling rate and more bits per encoding brings it closer

Digitized Audio: Compression Benefit

- Digitized audio can be compressed
  - e.g., MP3s
  - A CD can hold about 20 songs uncompressed, but about 200 compressed
- Compression is also done on digitized pictures (jpeg), movies (mpeg), and more.
- Digitization has many other benefits…

Example compression scheme:

00 --> 0000000000
01 --> 1111111111
1X --> X

0000000000 0000000000 0000001111 1111111111

00 00 10000001111 01

How to Encode Text: ASCII, Unicode

- ASCII: 7- (or 8-) bit encoding of each letter, number, or symbol
- Unicode: Increasingly popular 16-bit encoding
  - Encodes characters from various world languages

Symbol	Encoding
R	1010010
S	1010011
T	1010100
L	1001100
N	1001110
E	1000101
0	0110000
.	0101110
<tab>	0001001

Symbol	Encoding
r	1110010
s	1110011
t	1110100
l	1101100
n	1101110
e	1100101
9	0111001
!	0100001
<spa ce>	0100000

Question:
What does this ASCII bit sequence represent?
1010010 1000101 1010011 1010100

R E S T

Base Sixteen: Another Base Used by Digital Designers

		8	A	F
16^4	16^3	16^2	16^1	16^0

8 → 1000, A → 1010, F → 1111

hex	binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

hex	binary
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

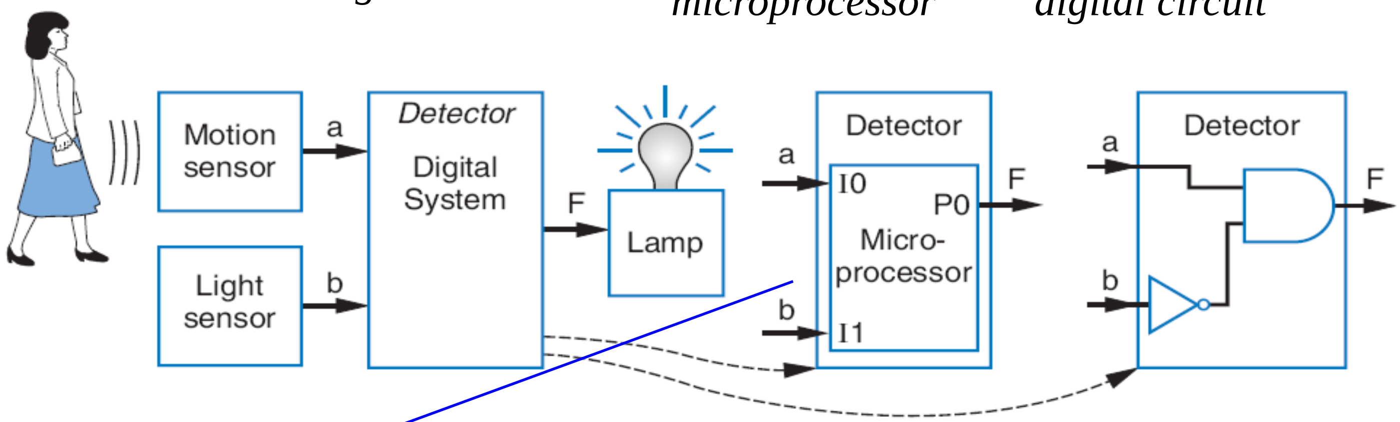
- Nice because each position represents four "base two" positions
  - Used as compact means to write binary numbers
- Known as ***hexadecimal***, or just ***hex***

Q: Write 11110000 in hex

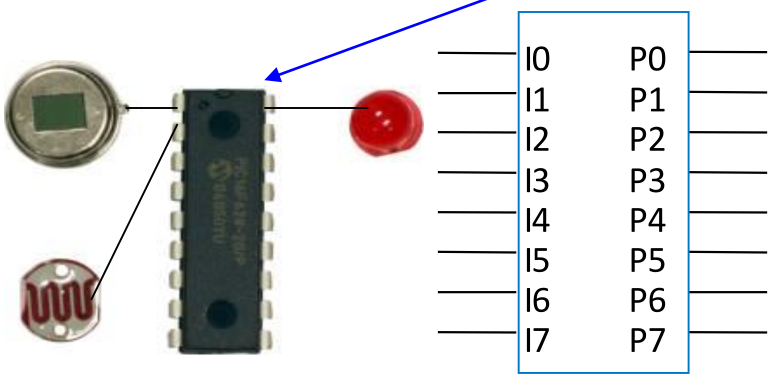
F0

Implementing Digital Systems: Programming Microprocessors vs. Designing Digital Circuits

Desired motion-at-night detector *Programmed microprocessor* *Custom designed digital circuit*



- Microprocessors a common choice to implement a digital system
  - Easy to program
  - Cheap (as low as $1)
  - Available now




```
void main()
{
  while (1) {
    P0 = I0 && !I1;
    // F = a and !b,
  }
}
```