CS221: Digital Design

Dr. A. Sahu
Dept of Comp. Sc. & Engg.
Indian Institute of Technology Guwahati

1

Outline

- Course, Attendance, Reference Book
- What do we study in this course?
- Why should this be studied?
- How is the course structured?

2

Instructors

- My self: A Sahu (ABS)
  – Pre MidSem Part
- Prof. Hemangee Kapoor (HK)
  – Post Midsem Part

3

CS221: Course Structure

- Part I : pre Mid Sem part **(ABS)**
  – Number system, Boolean Algebra, Logic Minimization and realization, K-Map, MQ method
  – Combinational Block, ROM, PLA, FPGA
  – HDL model of Combinational Block
- Part II : Post Mid Sem part **(HK)**
  – Flip-flops, registers, counters,
  – Finite state model Synthesis of synchronous sequential circuits
  – Number representation: fixed and floating point;
  – Addition, subtraction, multiplication and division of numbers.
  – Current trends in digital design: ASIC, FPGA..

4

CS221

- 75% Attendance is Mandatory
- Up to Mid Semester: Part I
  – 3 Quizs (3%,3% &4%), Scheduled
  – 1 Mid Semester Exam
- Post Mid Semester : Part II
  – 1 Quiz (Bonous Marks), Surprise
  – End Semester Exam (50%), 10% from Pre mid sem part
- Mark Distribution
  – [Q1 (3) +Q2(3) +Q4(4)+MS(30)] =40
  – [Q4(*)+ ES(50+10)] =60

5

CS221

- Up to Mid Semester: Part I
  – 3 Quizs (3%,3% &4%), Scheduled
  – 1 Mid Semester Exam
- Do attend all the classes
- There will be a Quiz Exam
  – Announced Quiz → No surprise
  – 6^{th} Class, 12^{th} Class and 18^{th} Class

6

CS221: Text Books

- Text Book
  - M. Morris **Mano** and M. D. Ciletti, *Digital Design*, 4/e, Pearson Education, 2007.
  - R. H. **Katz** and G. Boriello, *Contemporary Logic Design*, 2/e, Prentice Hall of India, 2009.
  - Anand **Kumar**, *Fundamental of Digital Circuits*, PHI, 2012
- *References*

7

Web, marks, timing, venue

- Course website: **http://jatinga.iitg.ernet.in/~asahu/cs221/**
- Mark distribution
  - 25% : 3 Prog. Assignment + 2 Quiz + Home work
  - 35% : Mid Semester
  - 40% : End Semester
- Class Timing & Venue
  - Venue: 1201, Timing : Slot B
  - Mon:9-10, Tue:10-11, Wed:11-12, Other one hour is make up slot

8

CS221 Course Objectives

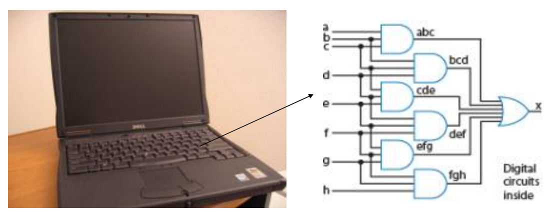
To learn -

- **Basic principle behind digital system**
- **Efficient design and reuse.**
- Interfacing with external world
- **Working of components inside a computer**
- Analog vs digital

9

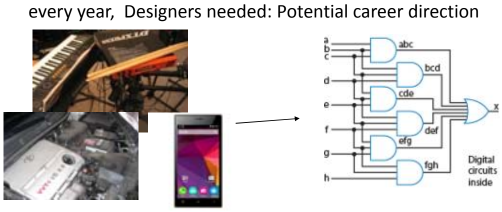
Why Study Digital Design?

- **Look "under the hood" of computers**
  - Solid understanding --> confidence, insight, even better programmer when aware of hardware resource issues



Why Study Digital Design?

- **Electronic devices becoming digital**
  - Enabled by shrinking and more capable chips
  - Enables: Better devices: Better sound recorders, cameras, cars, cell phones, medical devices,...
  - New devices: Video games, PDAs, ...
  - Known as "embedded systems" : Thousands of new devices every year, Designers needed: Potential career direction



Why Study Digital Design?

Satellites
DVD players
Video recorders
Musical instruments
Portable music players
Cell phones
Cameras
TVs
Smart Phone

1995 1997 1999 2001 2003 2005 2007 2015

Digital Design: Motivation

- Implementation basis for modern computing devices
  - Constructing large systems from small components
  - Another view of a computer: **controller + datapath**
- Inherent parallelism in hardware
  - Parallel computation beyond
  - Counterpoint to software design
  - Furthering our understanding of computation

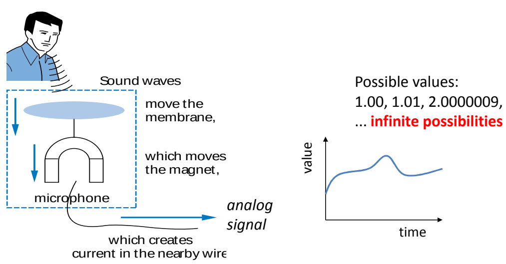
Digital Design

- What is digital ?
  - Digital camera, Digital TV, Digital Watch, Digital Radio, Digital City (e-city), Digital Photo Frame …etc
  - Which gives the things in countable form
  - Scene (analog) to Image (digital)
- Why digital ?
  - Countable form, makes easy to manage
  - Easy management makes more useful and versatile
- What digit ?
  - How to count: Decimal digit: 0 to 9

1.2

What Does "Digital" Mean?

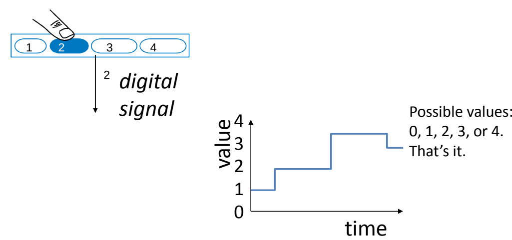
- Analog signal
  - Infinite possible values
    - Ex: voltage on a wire created by microphone



1.2

What Does "Digital" Mean?

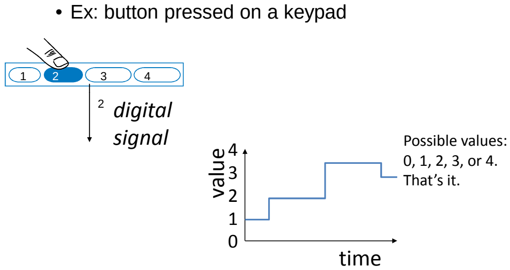
- Digital signal
  - **Finite possible values**
    - Ex: button pressed on a keypad



1.2

What Does "Digital" Mean?

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



What Digit? => Number System

- Famous Number System: Dec, Rom, Bin
- Decimal System: 0 -9
  - May evolves: because human have 10 finger
- Roman System
  - May evolves to make easy to look and feel
  - Pre/Post Concept: (IV, V & VI) is (5-1, 5 & 5+1)
- Binary System, Others (Oct, Hex)
  - One can cut an apple in to two

Design

- **What is design?**
  - Given problem spec, solve it with available components
  - While meeting quantitative (size, cost, power) and qualitative (beauty, elegance)

19

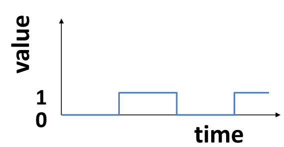
Logic Design

- **What is logic design?**
  - Solution of Design..from Prev Slide?
  - Choose digital logic components to perform specified control, data manipulation, or communication function and their interconnection
  - Which logic components to choose? Many implementation technologies (fixed-function components, *programmable devices*, individual transistors on a chip, etc.)
  - Design optimized/transformed to meet design constraints

20

Digital Signals with Only Two Values: Binary

- ***Binary*** digital signal -- only *two* possible values
  - Typically represented as **0** and **1**
  - One *B*inary dig*it* is a ***bit***

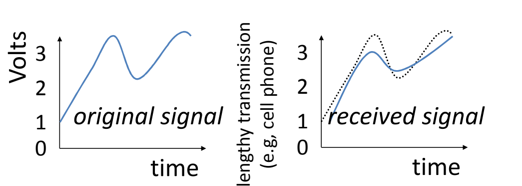


Digital Signals with Only Two Values: Binary

- We'll only consider *binary* digital signals
- Binary is popular because
  - Transistors, the basic digital electric component, 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)
  - Can be theoretically proved two is enough to represent all values. So not required to go Ternary, Quad nary…

Example of Digitization Benefit

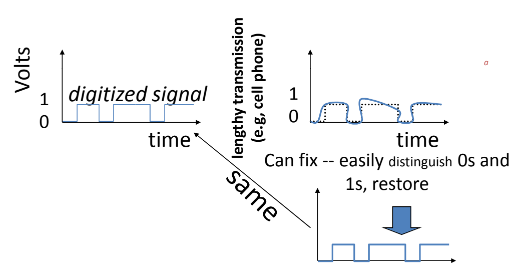
- **Analog signal (e.g., audio) may lose quality**
  - Voltage levels not saved/copied/transmitted perfectly



How fix -- higher, lower, ?

Example of Digitization Benefit

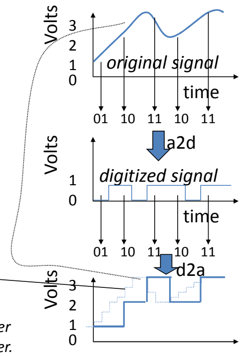
- **Digitized version enables near-perfect save/copy/transmi**
  - Digital signal : high distinguishing things
  - But we can distinguish 0s from 1s



Example of Digitization Benefit

- **Digitized version**
  - "Sample" voltage at particular rate, save sample using bit encoding
  - Voltage levels still not kept perfectly
  - But we can distinguish 0s from 1s

Let bit encoding be:
1 V: "01"
2 V: "10"
3 V: "11"



Digitized signal not perfect re-creation, but higher sampling rate and more bits per encoding brings closer.

Digital Vs Analog

- Analog : Scene/Land scape
- Digital : Image

- Digital : Require more effort to be perfect and most of time it is doable
- Monitor : VGA, HD, FHD, UHD,
- Camera : 1MP, 2 MP, …., 24MP
- One thing we don't like is
  - Digital Zoom of Camera..really waste…..
  - Optical Zoom: capture at the time phtographs…

26

Advance Digital Logic Design

- Complicated Circuit (Processor, Memory)
- IC, MSI, LSI, VLSI, ULSI…
- CAD Tool to accelerate
  - Design, Model, Simulate, Validate, Test
  - Placing, Routing, Power distribution
- VHDL/Verilog Simulation
- Schematic Editor
- Embedded System (Area, Power, Cost)
- Computer Architecture
- Advanced Computer Architecture

27

Number System

- Number System
  - Decimal, Binary, Octal, Hex
- Conversion (one to another)
  - Decimal to Binary, Octal, Hex & Vice Versa
  - Binary to HEX & vice versa
- Other representation
  - Signed, Unsigned, Complement
- Operation
  - Add, Sub, Mul, Div, Mod

28

Convergent Series

- The reciprocals of powers of any n produce a convergent series

$$\sum_{i=1}^{\infty}\frac{1}{n^i}=\frac{n}{n-1}$$

- Special case n=2

$$\sum_{i=1}^{\infty}\frac{1}{2^i}=2$$

- Cake problem
- Convergence ratio test:
  - $|a_{i+1}/a_i|<1$

29

Unknown Number Problem: Version 1

- **Problem:** Given an unknown integer X in the range R_{min} and R_{max} ($R_{min} \le X < R_{max}$) , We need to find the value of X by asking Boolean queries of type **a==x, a>x, a<x, a>=x and a<=x**
- **Goal:** is to minimize the number of question to find the value of X
- Is the problem definition **clear?**

Approach

- **Problem:** Given an unknown integer X in the range R_{min} and R_{max} ($R_{min} \le X < R_{max}$), We need to find the value of X by asking Boolean queries of type **a==x**, **a>x**, **a<x, a>=x and a<=x**
- Start from R_{min} and go upto R_{max}, one by one

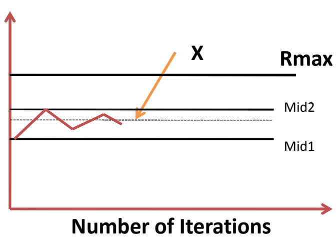
```
for(a=Rmin, a<Rmax; a++){
    if(x==a) break;
}
//Print  value of X is a
//Number of step required is X-Rmin
```

Approach-2

- **Is there any better approaches**
- Why to test one by one?
- Test at middle and set new range
  - Mid =(R_{min}+R_{max})/2
  - If (x==Mid) found
  - If (X>Mid) R_{min}=Mid+1 else R_{max}=Mid
- Binary Search….

Approach-2

- Test at middle and set new range
  - Mid =(R_{min}+R_{max})/2; If (x==Mid) found
  - If (X>Mid) R_{min}=Mid+1 else R_{max}=Mid



Rmax=255, Rmin=0, X=155

Mid1=127, Rmin=128
Mid2=191, Rmax=191
Mid3=159, Rmax=159
Mid4=143, Rmin=144
Mid5=151, Rmin=152
Mid6=155 ….Done

Approach-2

- **Is there any better approaches**
- Why to test one by one?
- Test at middle and set new range
  - Mid =(R_{min}+R_{max})/2
  - If (x==Mid) found
  - If (X>Mid) R_{min}=Mid+1 else R_{max}=Mid

```
while (Rmin<Rmax){
   mid=(Rmin+Rmax)/2;
   if(x==mid)return found;//print mid
   if (x>mid) Rmin=mid+1;
    else Rmax=mid;
}
```

Analysis: Approach-2

- Test at middle and set new range
  - Mid =(Rmin+Rmax)/2
  - If (x==Mid) found
  - If (X>Mid) Rmin=Mid+1 else Rmax=Mid
- Number of test:
  - 2 per iterations
  - Number of iteration : $\mathbf{Log_2 (R_{max}\text{-}R_{min})}$

Unknown Number Problem: Version 2

- **Problem:** Given an unknown integer X, We need to find the value of X by asking Boolean queries of type **a==x**, **a>x**, **a<x, a>=x and a<=x**
- **Goal:** is to minimize the number of question to find the value of X
- Is the problem definition **clear?**

Approach-1

- **Problem:** Given an unknown integer X, We need to find the value of X by asking Boolean queries of type **a==x**, **a>x**, **a<x, a>=x and a<=x**
- Start from 1 and go upto X, one by one

```
a=1;
while(a<X){{
     if(x==a) break;
     a=a+1;
}
//Print  value of x is a
//Number of step required is a-Rmin
```

Approach-2

- **Problem:** Given an unknown integer X, We need to find the value of X by asking Boolean queries of type a==x, a>x, a<x
- Is there any better approaches?
- Start from 1 but go at faster pace and find a range
  - Instead of **a = a+1 ,** use a = a+100

```
a=1;
while(a<X){a=a+100;}
// x will be between [a-100]<= X < a
Find Using previous method: Binary
search for X between Rmin and Rmax
```

Approach-2

- Start from 0 but go at faster pace and find a range : Instead of a = a+1 use a = a+M

```
a=1;
while(a<X){
     a = a + M;
}
// x will be between [a-M]<= X < a
Find Using previous method: Binary
search for X between Rmin and Rmax
```

- How good it is ?
- Number of steps: X/M + $\log_2$ M
- Can it be done better?

Approach-3

- Start from 0 but go at faster pace and find a range : Instead of a = a+1, use a = a*2

```
a=1;
while(a<X){
     a = a * 2;
}
// x will be between [a/2]<= X < a
Find Using previous method: Binary
search for X between Rmin and Rmax
```

- How good it is ?
- Number of steps **ceil($\log_2$ X) + ceil($\log_2$ X)**
  ≈ $\log_2$X

What Digit? => Number System

- Famous Number System: Dec, Rom, Bin
- Decimal System: 0 -9
  - May evolves: because human have 10 finger
- Roman System
  - May evolves to make easy to look and feel
  - Pre/Post Concept: (IV, V & VI) is (5-1, 5 & 5+1)
- Binary System, Others (Oct, Hex)
  - One can cut an apple in to two

41

Significant Digits

Binary: 11101101

Most significant digit *Least significant digit*

Decimal: 1063079

Most significant digit *Least significant digit*

Decimal (base 10)

- Uses positional representation
- Each digit corresponds to a power of 10 based on its position in the number
- The powers of 10 increment from 0, 1, 2, etc. as you move right to left
  - $1,479 = 1 * 10^3 + 4 * 10^2 + 7 * 10^1 + 9 * 10^0$

Binary (base 2)

- Two digits: 0, 1
- To make the binary numbers more readable, the digits are often put in groups of 4
  - $1010 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$
    $= 8 + 2$
    $= 10$
  - $1100\ 1001 = 1 * 2^7 + 1 * 2^6 + 1 * 2^3 + 1 * 2^0$
    $= 128 + 64 + 8 + 1$
    $= 201$

How to Encode Numbers: Binary Numbers

- Working with binary numbers
  - In base ten, helps to know powers of 10
    - one, ten, hundred, thousand, ten thousand, ...
  - In base two, helps to know powers of 2
    - one, two, four, eight, sixteen, thirty two, sixty four, one hundred twenty eight
    - Count up by powers of two

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1

Octal (base 8)

- Shorter & easier to read than binary
- 8 digits: 0, 1, 2, 3, 4, 5, 6, 7,
- Octal numbers

$136_8 = 1 * 8^2 + 3 * 8^1 + 6 * 8^0$
$= 1 * 64 + 3 * 8 + 6 * 1$
$= 94_{10}$

Hexadecimal (base 16)

- Shorter & easier to read than binary
- 16 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- “0x” often precedes hexadecimal numbers

$0x123 = 1 * 16^2 + 2 * 16^1 + 3 * 16^0$
$= 1 * 256 + 2 * 16 + 3 * 1$
$= 256 + 32 + 3$
$= 291$

Fun of the Binary World

- How is Halloween Festival is same as Christmas
  - $31_{\text{Octal}} = 25_{\text{Decimal}}$ ==> $31_{\text{Oct}} = 25_{\text{Dec}}$
  - == >31 Oct = 25 Dec
- There are 10 type of people
  - Who know Binary and don’t
- There are DEAD people know Hexadecimal.
  - How may people know Hexadecimal?
- What is your FACE value in decimal?

48

Counting

Decimal		Binary		Octal		Hexadecimal
0		00000		0		0
1		00001		1		1
2		00010		2		2
3		00011		3		3
4		00100		4		4
5		00101		5		5
6		00110		6		6
7		00111		7		7
8		01000		10		8

Counting

Decimal		Binary		Octal		Hexadecimal
9		01001		11		9
10		01010		12		A
11		01011		13		B
12		01100		14		C
13		01101		15		D
14		01110		16		E
15		01111		17		F
16		10000		20		10

Fractional Number

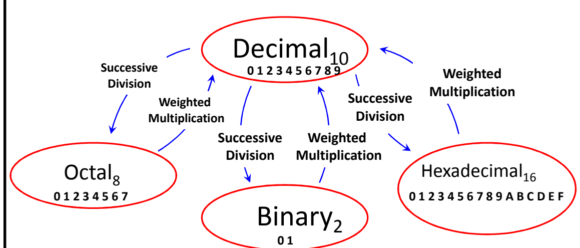
- Point: Decimal Point, Binary Point, Hexadecimal point
- Decimal

  247.75 $= 2x10^2+4x10^1+7x10^0+7x10^{-1}+5x10^{-2}$
- Binary

  $10.101= 1x2^1+0x2^0+1x2^{-1}+0x2^{-2}+1x2^{-3}$
- Hexadecimal

  $6A.7D=6x16^1+10x16^0+7x16^{-1}+Dx16^{-2}$

Converting To and From Decimal



52

Decimal ↔ Binary

a) Divide the decimal number by **2**; the remainder is the LSB of the **binary** number.

b) If the quotation is zero, the conversion is complete. Otherwise repeat step (a) using the quotation as the decimal number. The new remainder is the next most significant bit of the **binary** number.

Decimal ↔ Binary

a) Multiply each bit of the **binary** number by its corresponding bit-weighting factor (i.e., Bit-0→2^0=1; Bit-1→2^1=2; Bit-2→2^2=4; etc).

b) Sum up all of the products in step (a) to get the decimal number.

Decimal to Binary : Subtraction Method

- Goal
  - Good for human
  - Get the binary weights to add up to the decimal quantity
    - Work from left to right
    - (Right to left – may fill in 1s that shouldn't have been there – try it).

Desired decimal number: **12**

32	16	8	4	2	1	
1						=32 too much
0	1					=16 too much
0	0	1				=8 ok, keep going
0	0	1	1			=8+4=12 DONE
0	**0**	**1**	**1**	**0**	**0**	answer

Decimal to Binary : Division Method

- Good for computer: Divide decimal number by 2 and insert remainder into new binary number.
  - Continue dividing quotient by 2 until the quotient is 0.
- Example: Convert decimal number 12 to binary

12 div 2 = (Quo=6 , Rem=0) LSB
6 div 2 = (Quo=3, Rem=0)
3 div 2 = (Quo=1,Rem=1)
1 div 2 = (Quo=0, Rem=1) MSB

$12_{10} = 1\,1\,00_{2}$