BROWN
School of Engineering

DIGITAL ELECTRONICS SYSTEM DESIGN

FALL 2019

PROF. IRIS BAHAR

SEPTEMBER 4, 2019

LECTURE 1: INTRODUCTION

INSTRUCTORS & TAS

- Iris Bahar
  - Prof. of Engineering, Prof. of CS
  - Office: CIT449
  - Research interests: energy-efficient computing, computer architecture, robotics, emerging computing technologies
  - Teaching interests: digital design, robotics, emerging technologies, VLSI
- Graduate TAs: Jiwon Choe and Pratistha Shakya
- We will also have several undergraduate students helping with labs

COURSE TOPICS

- This course covers the fundamentals of digital logic design:
  - Boolean algebra, logic minimization
  - Gates, logic families, flip-flops
  - Sequential circuits, finite state machines
  - Static and dynamic memory design
  - Binary arithmetic
  - Programmable logic (CPLDs and FPGAs)
  - CAD for schematic capture and timing analysis
- Prerequisites: Not open to freshman. Some familiarity with electronics and/or discrete math (e.g., ENGN520, CSCI220)
  - See me if you are unsure of your eligibility

COURSE GOALS

- Design combinational and sequential logic for a wide range of systems
- Understand CMOS transistors and their use in logic circuits
- Realize logic designs in an appropriate choice of discrete logic, CPLD, or FPGA
- Use CAD tools for schematic capture, logic simulation, and programmable designs
- Understand role and advantages of hardware description languages
- Make use of memory and simple processors

CLASS MEETING TIMES

- Lectures M,W 3:00-4:20pm, B&H 153
- Lab space: B&H 196 (the fishbowl)
- Office hours:
  - Mondays 4:30-5:30pm, ERC lobby
  - Tuesdays 10-11am, CIT449 (my office)
  - by appointment
- TA Hours:
  - Expect 25-35 hrs/wk of lab staffing total from the TAs
  - Hours will be posted next week

LABORATORY ASSIGNMENTS

- This is a lab-intensive course (14 lab assignments in all)
- This is no set due date for any particular lab
  - Labs are grouped into clusters with due dates assigned to each cluster:
    - Group 1: labs 0-3, due by Oct. 6
    - Group 2: labs 4-9, due Nov. 17
    - Group 3: labs A, B, due Dec. 6
    - Group 4: labs C, D, due Dec. 13
    - Grace period during the last week before finals (see lab manual for details)
- Required labs: 1, 2, (7 or 8), 9, (B, C, or D) + logic analyzer + schematic
- ***Pace yourself throughout the semester*** so you can complete approximately1 lab per week
- You don't need to complete all labs to get a good grade.
  - Need a minimum of 57 points on the lab part
  - Aim to complete about 11-12 labs

GRADING

- Grade distribution:
  - Laboratory assignments: 55%
  - Midterm exam: 15%
  - Final exam: 25%
  - Class participation: 5%
- Labs are graded pass/fail and are worth a variable number of points depending on difficulty
- You need 57 out of a possible 101 points on the lab portion to pass the class
- You must receive a passing grade on both lab and exams portions of the course to get a passing grade

LAB MANUAL, KITS, TEXTBOOK, ETC.

- Lab manual and kits are required for the course
  - Pick up both from George Worth (B&H 325)
  - $60 (payable by check to Brown University)
  - $50 rebate at end of semester if you return the major parts of the kit
- Textbook
  - John F. Wakerly, *Digital Design: Principles and Practices, 5Ed.*
  - *OPTIONAL*
  - *Very useful textbook, but hefty list price (sorry)*
- *Course webpage:*

  www.brown.edu/Departments/Engineering/Courses/En163/home.html

COLLABORATION POLICY

- Laboratory assignments are to be done alone
  - You may collaborate on labs only by discussing them generally with classmates and TAs
  - TAs will give hints or suggestions only
  - Make sure you understand the problem and its solution for each lab (or you may not be able to answer questions from the TA)
  - All labs need to be built, debugged, and demonstrated on your own boards. Copying someone else's software is also not allowed.
  - You are responsible for taking your own data (for labs 2, 6, 9)
- Copying or using someone else's design as your own will not be tolerated!

DIVERSITY AND INCLUSION

- It is our intent that students from all diverse backgrounds be well-served by this course.
- The diversity the students bring to this class is a resource, strength, and benefit.
- We aim to present materials and provide lab space that is inclusive and respectful of diversity
- Likewise, we expect all students in class to be respectful of diversity and do their part in creating an inclusive environment.
- Your suggestions are encouraged and appreciated.

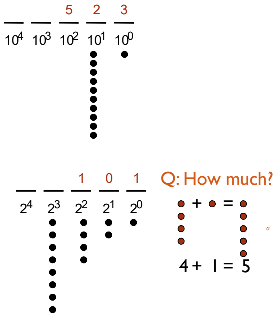
BOOLEAN ALGEBRA AND BINARY REPRESENTATION

BINARY VS. DIGITAL SYSTEMS

- Digital system: Finite number of values
- Binary (base 2) system: uses 2 states
  - Basic unit of information: Binary digit (i.e., *bit*)
  - Two values: 0, 1
  - 0 and 1 represented by *voltage ranges*
    - *Don't need to be exact*
    - *Electronic circuits don't need to be perfect*

ENCODING NUMBERS IN BINARY

- Each position represents a quantity; symbol in position means how many of that quantity
  - Base ten (decimal)
    - Ten symbols: 0, 1, 2, ..., 8, and 9
    - More than 9 -- next position
      - So each position is a power of 10
    - Nothing special about base 10 -- used because we have 10 fingers
  - Base two (binary)
    - Two symbols: 0 and 1
    - More than 1 -- next position
      - So each position is a power of 2



Q: How much?

ENCODING NUMBERS IN BINARY

- 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
      - (Note: unlike base ten, we don't have common names, like "thousand," for each position in base ten -- so we use the base two name)
    - Q: 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

512 256 128 64 32 16 8 4 2 1

CONVERTING FROM DECIMAL TO BINARY: SUBTRACTION METHOD

- Goal
  - 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

CONVERTING FROM DECIMAL TO BINARY: SUBTRACTION METHOD

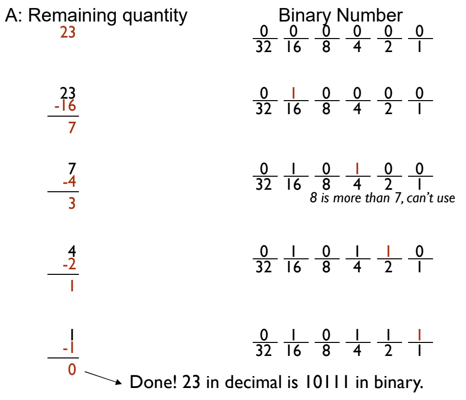
- Subtraction method
  - To make the job easier (especially for big numbers), we can just subtract a selected binary weight from the (remaining) quantity
    - Then, we have a new remaining quantity, and we start again (from the present binary position)
    - Stop when remaining quantity is 0

Remaining quantity: **12**

32	16	8	4	2	1	
1						32 is too much
0	1					16 is too much
0	0	1				**12 – 8 = 4**
0	0	1	1			**4-4=0** DONE
0	**0**	**1**	**1**	**0**	**0**	answer

BINARY CONVERSION EXAMPLE

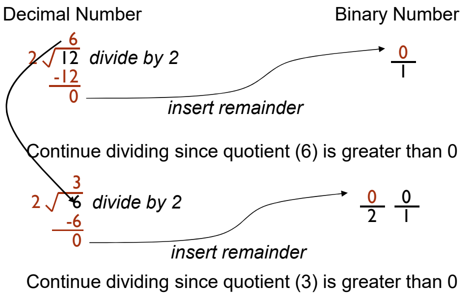
- Q: Convert the number “23” from decimal to binary



Done! 23 in decimal is 10111 in binary.

CONVERTING FROM DECIMAL TO BINARY: DIVISION METHOD

- Divide decimal number by 2, insert remainder into new binary number.
  - Continue dividing quotient by 2 until quotient is 0.
- Example: Convert decimal number 12 to binary

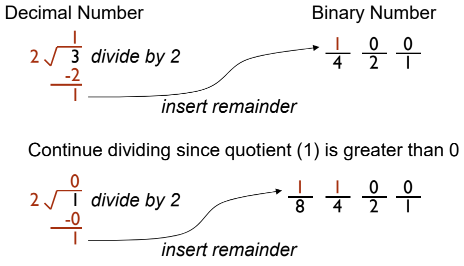


Continue dividing since quotient (6) is greater than 0

Continue dividing since quotient (3) is greater than 0

CONVERTING FROM DECIMAL TO BINARY: DIVISION METHOD

- Example: Convert decimal number 12 to binary (continued)

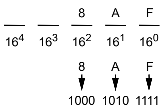


Continue dividing since quotient (1) is greater than 0

Since quotient is 0, we can conclude that 12 is 1100 in binary

---

BASE 16



hex	binary	hex	binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	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

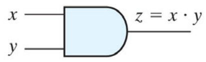
F 0

TRUTH TABLES OF LOGICAL OPERATIONS

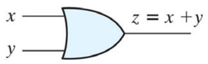
Table 1.8
Truth Tables of Logical Operations

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

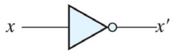
SYMBOLS FOR DIGITAL LOGIC CIRCUITS



(a) Two-input AND gate

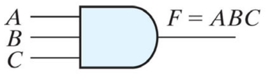


(b) Two-input OR gate

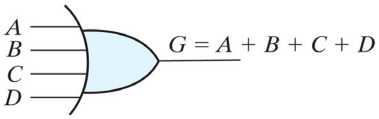


(c) NOT gate or inverter

GATES WITH MULTIPLE INPUTS

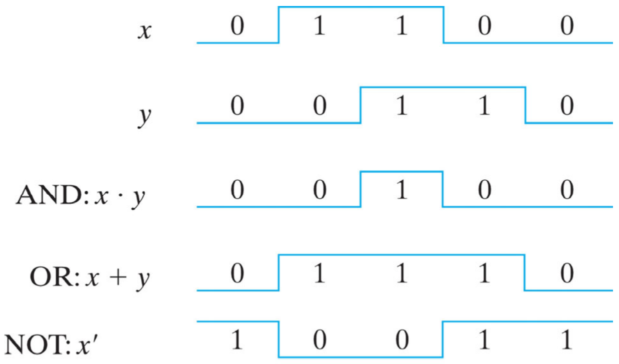


(a) Three-input AND gate



(b) Four-input OR gate

INPUT–OUTPUT SIGNALS FOR GATES



BOOLEAN ALGEBRA

- Set of axioms and theorems to simplify Boolean equations
- Like regular algebra, but in some cases simpler because variables can have only two values (1 or 0)
- Axioms and theorems obey the principles of duality:
  - ANDs and ORs interchanged, 0's and 1's interchanged

BOOLEAN AXIOMS

	Axiom		Dual	Name
A1	$B = 0$ if $B \neq 1$	A1′	$B = 1$ if $B \neq 0$	Binary field
A2	$\overline{0} = 1$	A2′	$\overline{1} = 0$	NOT
A3	$0 \bullet 0 = 0$	A3′	$1 + 1 = 1$	AND/OR
A4	$1 \bullet 1 = 1$	A4′	$0 + 0 = 0$	AND/OR
A5	$0 \bullet 1 = 1 \bullet 0 = 0$	A5′	$1 + 0 = 0 + 1 = 1$	AND/OR

	Theorem		Dual	Name
T1	$B \bullet 1 = B$	T1′	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	T2′	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	T3′	$B + B = B$	Idempotency
T4		$\overline{\overline{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	T5′	$B + \overline{B} = 1$	Complements