

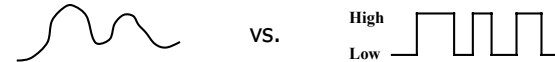
Introduction to Digital Systems

- This course is about ...
 - Logic and **digital circuits/gates** to implement logic
- What you will get at the end of this semester ...
 - A systematic way to **specify, design, and analyze** digital systems
 - Fundamentals to subsequent courses, e.g., VHDL, VLSI, ...
- What I expect from you ...
 - **Get involved** and work hard
- Grading policies, labs, office hours, contact info., etc.
 - See handouts ...

S. Jay Yang, CE, RIT

Analog vs. Digital Systems

- **Analog/Digital** systems process time-varying signals that take on **continuous/discrete 2-phase** values.



- Discrete 2-phase (a.k.a. Boolean, binary) values:
 - **high/low, true/false, or 1/0.**
- Digital computers:
 - computation via logical representation to fulfill the tasks
- Once analog systems that is digital today:
 - Audio recording (CDs, DAT, mp3)
 - Phone system switching
 - Automobile engine control
 - Movie effects, still and video cameras....

S. Jay Yang, CE, RIT

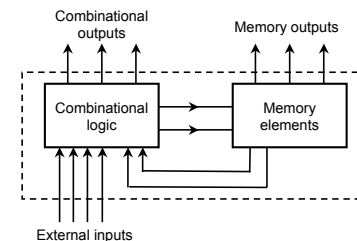
Advantages of Digital Over Analog Systems

- **Reproducibility** of the results and accuracy
- More **reliable** due to better immunity to noise
- **Ease of design**: no special math skills needed to visualize the behavior of small digital (logic) circuits
- **Flexibility and functionality**
- **Programmability**
- **Speed**: a complete complex digital integrated circuits (IC) can produce an output in less than 2 nanoseconds (2 ns or 2×10^{-9} seconds)
- **Economy**: very low cost ICs due to repetitive and mass production of integration of millions of digital logic elements on a single miniature chip

S. Jay Yang, CE, RIT

What is digital logic?

- Use combinations of binary TRUEs & FALSEs to resemble the way "we" solve problems – hence combinational logics
- We may use steps of logical thinking and/or past decisions (i.e. memory) to solve problems – hence sequential logics



S. Jay Yang, CE, RIT

Number Systems

- Positional number systems:

$$\text{Number: } D_r = d_{n-1}d_{n-2}\cdots d_1d_0d_{-1}\cdots d_{-n}$$

$$\text{Value: } D_r = \sum_{i=-n}^{n-1} d_i \times r^i$$

- Examples:

- decimal:

$$5185.68_{10} = 5 \times 10^3 + 1 \times 10^2 + 8 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2}$$

$$= 5 \times 1000 + 1 \times 100 + 8 \times 10 + 5 \times 1 + 6 \times 0.1 + 8 \times 0.01$$

- binary (radix=2, digits={0, 1})

$$10011_2 = 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 19_{10}$$

MSB LSB

$$101.001_2 = 1 \times 4 + 0 \times 2 + 1 \times 1 + 0 \times 0.5 + 0 \times 0.25 + 1 \times 0.125 = 5.125_{10}$$

S. Jay Yang, CE, RIT

Common number systems

System	Radix	Set of Digits	Example
Decimal	r=10	{0,1,2,3,4,5,6,7,8,9}	255 ₁₀
Binary	r=2	{0,1}	1111111 ₂
Octal	r= 8	{0,1,2,3,4,5,6,7}	377 ₈
Hexadecimal	r=16	{0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F}	FF ₁₆

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

S. Jay Yang, CE, RIT

Radix-r to decimal conversion

- Simply expand using the definition ...

$$D_r = \sum_{i=-n}^{n-1} d_i \times r^i$$

- Examples:

- 1101.101₂ = 1×2³ + 1×2² + 1×2⁰ + 1×2⁻¹ + 1×2⁻³
- = 8 + 4 + 1 + 0.5 + 0.125 = 13.625₁₀

- 572.6₈ = 5×8² + 7×8¹ + 2×8⁰ + 6×8⁻¹
- = 320 + 56 + 16 + 0.75 = 392.75₁₀

- 2A.8₁₆ = 2×16¹ + 10×16⁰ + 8×16⁻¹
- = 32 + 10 + 0.5 = 42.5₁₀

- 132.3₄ = 1×4² + 3×4¹ + 2×4⁰ + 3×4⁻¹
- = 16 + 12 + 2 + 0.75 = 30.75₁₀

- 341.24₅ = 3×5² + 4×5¹ + 1×5⁰ + 2×5⁻¹ + 4×5⁻²
- = 75 + 20 + 1 + 0.4 + 0.16 = 96.56₁₀

S. Jay Yang, CE, RIT

Decimal to binary conversion

- Separate the decimal number into whole and fraction portions.
- To convert the whole number portion to binary, use successive division by 2 until the quotient is 0. The remainders form the answer, with the first remainder as the **least significant bit (LSB)** and the last as the **most significant bit (MSB)**.

- Example: Convert 179₁₀ to binary:

$$179 / 2 = 89 \text{ remainder } 1 \text{ (LSB)}$$

$$/ 2 = 44 \text{ remainder } 1$$

$$/ 2 = 22 \text{ remainder } 0$$

$$/ 2 = 11 \text{ remainder } 0$$

$$/ 2 = 5 \text{ remainder } 1$$

$$/ 2 = 2 \text{ remainder } 1$$

$$/ 2 = 1 \text{ remainder } 0$$

$$/ 2 = 0 \text{ remainder } 1 \text{ (MSB)}$$

$$\Rightarrow 179_{10} = 10110011_2$$

S. Jay Yang, CE, RIT

Decimal to binary conversion – cont.

- To convert decimal fractions to binary, repeat multiplication by 2 until the fractional product is 0 (or until the desired number of binary places). The whole digits of the multiplication results produce the answer, with the first as the MSB, and the last as the LSB.
- Example: Convert 0.3125_{10} to binary

		Result Digit	
$.3125 \times 2$	$= 0.625$	0	(MSB)
$.625 \times 2$	$= 1.25$	1	
$.25 \times 2$	$= 0.50$	0	
$.5 \times 2$	$= 1.0$	1	(LSB)
$\Rightarrow 0.3125_{10} = .0101_2$			

S. Jay Yang, CE, RIT

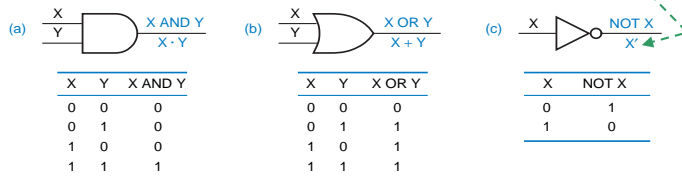
Many representations of digital logic

- Truth tables (✓)
 - Boolean expressions (✓)
 - Logic gate diagrams (✓)
 - Transistor level diagrams
 - Parts placement diagrams
 - High level description languages, e.g., VHDL
- (✓): focuses of this course

S. Jay Yang, CE, RIT

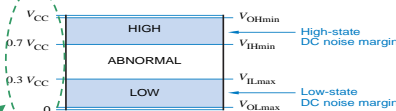
Digital logic representations

- Switching algebra:
 - Elements: $\{0,1\}$
 - Operators: $\{', +, \cdot, \dots\}$



Sometimes indicated as \bar{X} , $\sim X$ or $\neg X$

Depending on the fabrication technology - V_{cc} can be 5v or 3V

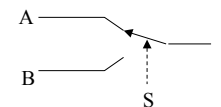


Copyright © 2000 by Prentice Hall, Inc. Digital Design Principles and Practices, 3/e

S. Jay Yang, CE, RIT

Representations: truth tables & Boolean expressions

- Truth table: provides a list of every possible combination of binary inputs to a digital circuit and the corresponding outputs
- Boolean expression: express logic in a functional format
- Example: Multiplexer
 - $Z = S' \cdot A + S \cdot B$
 - S determines whether output Z equals input A or B



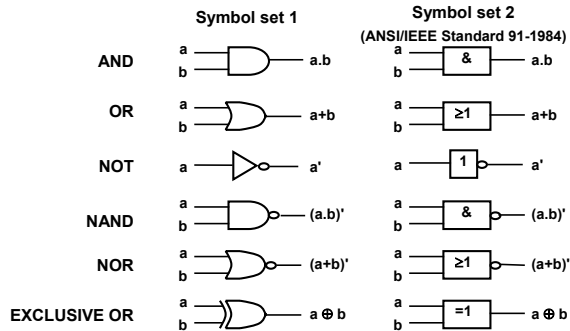
Truth table for the multiplexer function.

S	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

S. Jay Yang, CE, RIT

Representation: logic gate diagrams

- Logic gates: basic building blocks of digital circuits

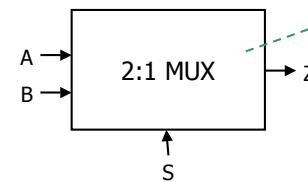
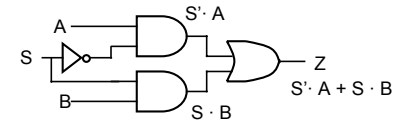


S. Jay Yang, CE, RIT

Representation: logic gate diagrams

- Example: multiplexer

- $Z = S' \cdot A + S \cdot B$



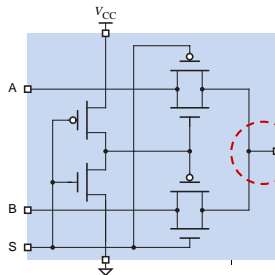
Can be used for bigger circuits

S. Jay Yang, CE, RIT

Representation: transistor level diagrams

- Example: multiplexer

- $Z = S' \cdot A + S \cdot B$



The # gates/transistors (fan-out) that can be driven depends on the size & # of the driving transistors

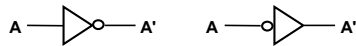
S. Jay Yang, CE, RIT

Representation: parts placement diagrams

- Parts: real world hardware that realizes digital logics
- The transistor-transistor logic (TTL) 74LS family of digital integrated circuits produces two voltage levels:
 - < 0.5V represents low voltage L (0)
 - > 2.7V represents high voltage H (1)
- Parts placement diagrams:
 - Parts connected via wires – lab manual
 - Help to **debug** your circuits!!

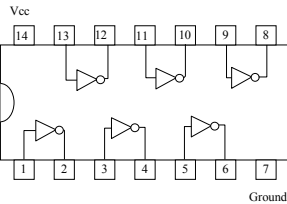
S. Jay Yang, CE, RIT

TTL IC example: Inverter



A	A'
0	1
1	0

Truth table

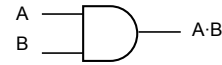


Top View of a TTL 74LS family 74LS04 Hex Inverter IC Package

May be accessed via
http://www.ce.rit.edu/reference-material/EECC341/data_sheets.shtml

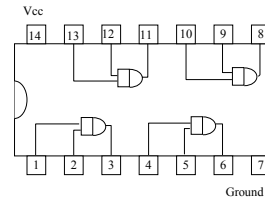
S. Jay Yang, CE, RIT

TTL IC example: AND gate



A	B	A · B
0	0	0
0	1	0
1	0	0
1	1	1

Truth table



Top View of a TTL 74LS family 74LS08 Quad 2-input AND Gate IC Package

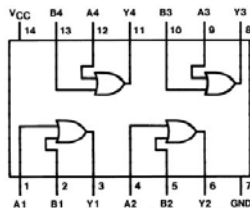
S. Jay Yang, CE, RIT

TTL IC example: OR gate



A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

Truth table



Top View of a TTL 74LS family 74LS08 Quad 2-input OR Gate IC Package

S. Jay Yang, CE, RIT

TTL IC example: NAND gate

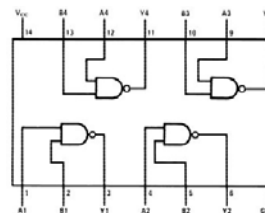


- NAND gate is **self-sufficient**: can build any logic circuit with it, including AND/OR/NOT.
- Example: implement NOT using NAND



A	B	(A·B)'
0	0	1
0	1	1
1	0	1
1	1	0

Truth table



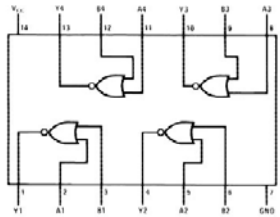
Top View of a TTL 74LS family 74LS00 Quad 2-input NAND Gate IC Package

S. Jay Yang, CE, RIT

TTL IC example: NOR gate

$$A \text{ NOR } B = (A+B)'$$

- NOR gate is also **self-sufficient**.
- Question: how to implement NOT using NOR?



A	B	$(A+B)'$
0	0	1
0	1	0
1	0	0
1	1	0

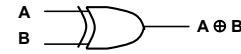
Truth table

Top View of a TTL 74LS family 74LS02 Quad 2-input NOR Gate IC Package

S. Jay Yang, CE, RIT

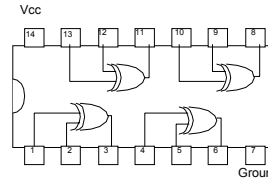
TTL IC example: XOR gate

$$A \oplus B = A B' + A' B$$



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

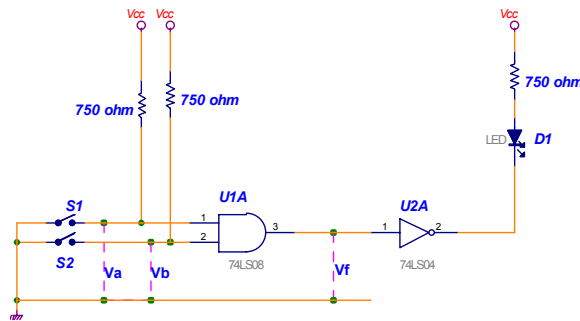
Truth table



Top View of a TTL 74LS family 74LS86 Quad 2-input XOR Gate IC Package

S. Jay Yang, CE, RIT

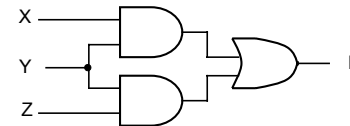
Circuit to determine AND Gate



S. Jay Yang, CE, RIT

Self exercise

- Download student version PSPICE from <http://www.orcad.com>, and draw a logic gate diagram like the one below.
- Write down the Boolean expression of F.



S. Jay Yang, CE, RIT