## All haill the HARDWAREI

## Chapter 4

# Gates and Circuits (with some transistors thrown in for good measure) 

## Abstractions and more abstractions ...

## Computers

Made of lots of different circuits (CPU, memory, controllers, etc.)

## Circuits

Made from gates combined to perform more complicated tasks

## Gates

Devices that perform basic logical operations on electrical signals. They're built out of transistors

## Transistors

Very small electronic switches

## How do we describe the behavior of gates and

## circuits?

1. Boolean expressions

Uses Boolean algebra, a mathematical notation for expressing two-valued logic
2. Logic diagrams

A graphical representation of a circuit; each gate has its own symbol
3. Truth tables

A table showing all possible input value and the associated output values

### 4.2 Gates

There are six basic gates:

- NOT
- AND
- OR
- XOR
- NAND
- NOR

Typically, logic diagrams are black and white with gates distinguished only by their shape
We use color for emphasis (and fun)

## NOT Gate (a.k.a. inverter)

A NOT gate accepts one input signal (0 or 1) and returns the opposite signal as output

Boolean Expression Logic Diagram Symbol

$$
\mathrm{X}=\mathrm{A}^{\prime}
$$



Truth Table

| $\mathbf{A}$ | $\mathbf{X}$ |
| :---: | :---: |
| 0 | 1 |
| 1 | 0 |

Figure 4.1 Various representations of a NOT gate

## AND Gate

An AND gate accepts two input signals If both are 1 , the output is 1 ; otherwise, the output is 0


Figure 4.2 Various representations of an AND gate

## OR Gate

An OR gate accepts two input signals If both are 0 , the output is 0 ; otherwise, the output is 1
Boolean Expression Logic Diagram Symbol

Truth Table

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Figure 4.3 Various representations of a OR gate

## XOR Gate

If both inputs are the same, the output is 0 ; otherwise, the output is 1


XOR is called the exclusive OR
Pronunciations: zor, ex-or


| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| $\mathbf{0}$ | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| $\mathbf{1}$ | 1 | 1 |

## QUIZ: recognize the gate!

$$
X=A \cdot B
$$

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$$
X=A+B
$$

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$
\begin{array}{l|l|l|}
\hline 1 & 0 & 0 \\
\hline 1 & 1 & 1 \\
\hline X=A^{\prime} \quad X & X \\
\hline
\end{array}
$$

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## QUIZ: draw the gate symbols!

$$
X=A \cdot B
$$

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$$
X=A+B
$$

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$
X=A^{\prime} \quad X=A^{\oplus}
$$

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## QUIZ: elementary properties

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |


| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |


| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

A AND $0=$ ?
AAND $1=$ ?
Etc.

## NAND Gate

If both inputs are 1 , the output is 0 ; otherwise, the output is 1

| Boolean Expression$X=(A \cdot B)^{\prime}$ | Logic Diagram Symbol | Truth Table |  |  |
| :---: | :---: | :---: | :---: | :---: |
|  |  | A | B | X |
|  |  | 0 | 0 | 1 |
|  |  | 0 | 1 | 1 |
|  |  | 1 | 0 | 1 |
|  |  | 1 | 1 | 0 |

Figure 4.5 Various representations of a NAND gate

## NOR Gate

The NOR gate accepts two input signals If both are 0 , the output is 1 ; otherwise, the output is 0


Figure 4.6 Various representations of a NOR gate

## Review of Gates

A NOT gate inverts its single input
An AND gate produces 1 if both input values are 1
An OR gate produces 0 if both input values are 0
An XOR gate produces 0 if input values are the same
A NAND gate produces 0 if both inputs are 1
A NOR gate produces a 1 if both inputs are 0

## Quiz

- What are the 3 ways we use to describe gates and circuits?
- Use the 3 ways to describe the NAND gate - Hint: Describe AND first!


## Solve in notebook for next time:

End of chapter:
-1 through 10
-18 through 29

## Gates with More Inputs

Gates can be designed to accept three or more input values A three-input AND gate, for example, produces an output of 1 only if all input values are 1


## QUIZ

Draw the gate symbols for:

- 4-input OR
- 5-input NAND
- 3-input NOR
- 4-input XOR


## QUIZ

Draw the gate symbols for:

- 4-input OR
- 5-input NAND
- 3-input NOR
- 4-input XOR

How many lines does each of the truth tables have?

## QUIZ

Draw the gate symbols for:

- 4-input OR
- 5-input NAND
- 3-input NOR
- 4-input XOR

How many lines does each of the truth tables have?

Describe in your own words each of the truth tables.

## QUIZ

A computer represents numbers in 8-bit two's complement. Design a circuit that will detect the number zero (the output of the circuit becomes 1 if and only if all 8 bits are 0 ):

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

## Extra-credit QUIZ

A computer represents numbers in 8-bit two's complement. Design a circuit that will detect the number -128 Hint: -128 is 10000000 in two's comp.

| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

## From gates to circuits

Find the logic diagram of the circuit described by the following truth table:


Hint: The table is similar to which of the fundamental gates presented last time?

## SOLUTION

| A | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | $\mathbf{0}$ |
| 1 | 1 | 1 |

Logic Diagram Symbol

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| $\mathbf{B}$ | 0 | 1 |
| 1 | 1 | 1 |

Having only one 0 in the output column, the circuit most resembles the OR gate!
It is different from the OR gate only in this respect: ...

Write the Boolean expression:
Draw the diagram:


## For next time:

Read the entire Section 4.2 Gates.
Understand the examples given in the text and quizzes.

## Comments and solutions for Ch. 2 homework

Remember: There are 3 layers of computer abstraction that we examine in this chapter:


## Gates

Transistors

### 4.3 Constructing Gates

Transistor = device that acts either as a wire that conducts electricity or as a resistor that blocks the flow of electricity, depending on the voltage level of an input signal

Acts like a switch, but w/o moving parts:

- Switch open
- Switch closed

Made of a semiconductor material

- Neither good conductor of electricity nor a good insulator


## Transistors



Figure 4.8 The connections of a transistor

A transistor has three terminals:

- A collector/source, typically connected to the positive terminal of a power source (5 volts, 3.5 volts, etc.)
- An emitter/drain, typically connected to the "ground" (0 volts)
- A base/gate, which controls the flow of current between source and emitter


## The names of transistor terminals <br> -setting the record straight FYI-



Bipolar Junction Transistor (BJT)
Collector, Base, Emmiter


Field-Effect Transistor (FET)
Drain, Gate, Source
Source


Ground

## How transistors operate as switches



The easiest gates to create are the NOT, NAND, and NOR

| NOT gate | NAND gate | NOR gate |
| :---: | :---: | :---: |
|  |  |  |

We can explain their operation for any combination of inputs! We do this by replacing the transistors with switches!

If there is no path from output to Ground, $\mathrm{V}_{\text {out }}=1$ If there is a path from output to Ground, $\mathrm{V}_{\text {out }}=0$


## QUIZ

The AND gate is obtained as a NAND followed by an inverter. Draw its transistor diagram!


## QUIZ

Draw its switch diagram.
Show the states of all switches for $\mathrm{V} 1=0$ and $\mathrm{V} 2=1$.


### 4.4 Circuits

## Combinational circuit

The input values explicitly determine the output

## Sequential circuit

The output is a function of the input values and the existing state of the circuit

We describe the circuit operations using
Boolean expressions
Logic diagrams
Truth tables

## Combinational Circuits

Gates are combined into circuits by using the output of one gate as the input for another


## Combinational Circuits



Three inputs require $2^{3}=8$ rows to describe all possible input combinations
Boolean expression is: $\quad \mathrm{X}=\mathrm{AB}+\mathrm{AC}$

## Another Combinational Circuit



## Another Combinational Circuit

Consider the following Boolean expression $\mathrm{A}(\mathrm{B}+\mathrm{C})$


Does this truth table look familiar?

## Circuit equivalence

Two circuits that produce the same output for identical input
Boolean algebra allows us to apply provable mathematical principles to help design circuits
$A(B+C)=A B+B C$ (distributive law) so circuits must be equivalent


## Properties (laws) of Boolean Algebra

| Property | AND | OR |
| :--- | :--- | :--- |
| Commutative | $A B=B A$ | $A+B=B+A$ |
| Associative | $(A B) C=A(B C)$ | $(A+B)+C=A+(B+C)$ |
| Distributive | $A(B+C)=(A B)+(A C)$ | $A+(B C)=(A+B)(A+C)$ |
| Identity | $A 1=A$ | $A+0=A$ |
| Complement | $A\left(A A^{\prime}\right)=0$ | $A+\left(A^{\prime}\right)=1$ |
| DeMorgan's law | $\left.(A B)^{\prime}\right)=A^{\prime}$ OR $B^{\prime}$ | $(A+B)^{\prime}=A^{\prime} B^{\prime}$ |

## DeMorgan's law applied directly to gates



$$
\text { (a) } \overline{X_{1} X_{2}}=\bar{X}_{1}+\bar{X}_{2}
$$


(b) $\overline{x_{1}+x_{2}}=\bar{x}_{1} \bar{x}_{2}$

# Using double-complement to "set up" DeMorgan 

Any two-level AND-OR circuit had a 2-level NAND-only equivalent!


## DeMorgan’s law QUIZ

Apply DeMorgan's Law directly on the gate diagrams below to obtain equivalent circuits:


## Solve in notebook for next time:

End of chapter:

- 11, 12
- 46 through 50
$-55$


## QUIZ

The XOR operation can be implemented with AND, OR and NOT gates:


## Very Useful Combinational Circuit: the Adder

At the digital logic level, addition is performed in binary
Addition operations are carried out by special circuits called adders

## Half-Adder truth table

The result of adding two binary digits could produce a carry value
Recall that $1+1=10$
in base two

Half adder
A circuit that computes the sum of two bits
and produces the correct carry bit


## Half Adder



## Circuit diagram

## Boolean expressions <br> Sum $=A \oplus B$ <br> Carry $=\mathrm{A} \cdot \mathrm{B}$

## How many

transistors are here?

## Full Adder

## This adder takes the Carry-in value into account!

Truth Table


| A | B | Carry- <br> in | Sum | Carry- <br> out |
| :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | 50 |

## Adding multiple bits <br> - "Ripple Carry" Adder -



# Very Useful Combinational Circuit: the Multiplexer 

MUX = A circuit that uses a few input control signals to determine which of several output data lines is routed to its output.
It is nothing but an electronic rotary switch!


## Multiplexer symbol and truth table



Figure 4.11 A block diagram of a multiplexer with three select control lines

| S0 | S1 | S2 | F |
| :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | D0 |
| 0 | 0 | 1 | D1 |
| 0 | 1 | 0 | D2 |
| 0 | 1 | 1 | D3 |
| 1 | 0 | 0 | D4 |
| 1 | 0 | 1 | D5 |
| 1 | 1 | 0 | D6 |
| 1 | 1 | 1 | D7 |

The control lines S0, S1, and S2 determine which of eight other input lines
(D0 ... D7)
are routed to the output (F)

## "Lookup table" with MUX

| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{C}$ | $\mathbf{A}(\mathbf{B}+\mathbf{C})$ |
| :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |



## QUIZ

Connect the MUX input to implement a prime number detector (i.e. the output F is 1 iff $\mathrm{S}_{2} \mathrm{~S}_{1} \mathrm{~S}_{0}$ are the binary code of a prime number)


### 4.5 Circuits as Memory a.k.a. Sequential Circuits

A sequential circuit is one whose output depends not only on the current values of its inputs, but also on the past sequence of those inputs (history).

It can be used to store information, i.e. as memory.

## Putting things in perspective: Delay-Line memories



Mercury memory of UNIVAC I (1951) Sources: Wikipedia, UNIVAC manual

How many Bytes of memory total?
(see notes)

## The S - R latch



Figure 4.12 An S-R latch
There are several ways to build S - R latches using various kinds of gates, but there's always feedback.

## S - R latch

## The value of $X$ at any point in time is considered to be the state of the circuit



Figure 4.12 An S-R latch

If $X$ is 1 , we say that the circuit is storing a 1 ; if $X$ is 0 , the circuit is storing a 0
As long as $\mathrm{S}=\mathrm{R}=1$, an $\mathrm{S}-\mathrm{R}$ latch stores a single binary digit,1 or 0 .
The design guarantees that the two outputs $X$ and $Y$ are (almost always) complements of each other.

$$
S-R \text { latch } \begin{aligned}
& \text { To make } X=1, \\
& \text { make } S=0, \text { while } \\
& \text { keeping } R=1 .
\end{aligned}
$$



Real-life check: since we make $S=0$ or $R=0$, we say that the signals $S$ and $R$ are active low.

They are normally denoted $S^{\prime}$ and R'.

Accordingly, this type of S - R latch is called "non-S non-R".

## S - R latch "forbidden" inputs



What happens if both $S$ and $R$ are activated (made 0) at the same time?

Integrated Circuit (a.k.a. IC or chip) = A piece of silicon on which multiple gates have been embedded

Silicon pieces are mounted on a plastic or ceramic package with pins along the edges that can be soldered onto circuit boards or inserted into appropriate sockets


## Integrated Circuits

Integrated circuits (IC) are classified by the number of gates contained in them

| Abbreviation | Name | Number of Gates |
| :---: | :--- | :--- |
| SSI | Small-Scale Integration | 1 to 10 |
| MSI | Medium-Scale Integration | 10 to 100 |
| LSI | Large-Scale Integration | 100 to 100,000 |
| VLSI | Very-Large-Scale Integration | more than 100,000 |

## Integrated Circuits



Figure 4.13 An SSI chip containing NAND gates



VLSI chip: AMD Phenom II CPU contains 768 million transistors

## CPU Chips

The most important integrated circuit in any computer is the Central Processing Unit, or CPU
Each CPU chip has a large number of pins through which essentially all communication in a computer system occurs

## Ethical Issues

Email Privacy
Explain why privacy is an illusion.
Who can read your email?
Do you send personal email from work?
Does everyone in your family use email?

## Who am I?



All the world knows my name. What is it and why do people know it?

## Do you know?

What is the name of the study of materials smaller than 100 nanometers?

Did DeMorgan discover DeMorgan's laws?
Who did the 4th Infantry Division take to war with them?

What is a virtual charity event?

## Chapter review questions

- Identify the basic gates and describe the behavior of each
- Describe how gates are implemented using transistors
- Combine basic gates into circuits
- Describe the behavior of a gate or circuit using Boolean expressions, truth tables, and logic diagrams


## Chapter review questions

- Compare and contrast a half adder and a full adder
- Describe how a multiplexer works
- Explain how an S-R latch operates
- Describe the characteristics of the four generations of integrated circuits

Work in notebook: End-of-chapter 62, 63, 64
Read an take notes: Ethical issues, Email privacy, Trivia

## Homework

Due Wednesday, Feb.22, at beginning of lab, before midterm:

End-of-chapter 39, 40, 56, 59, 60, 65, 66, 67, 68

## Review for Midterm - Ch. 4



## Truth Table

Use a MUXes as "lookup tables" to implement the 1-bit adder

| A | B | Carry- <br> in | Sum | Carry- <br> out |
| :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| $\mathbf{1}$ | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |
|  |  |  |  |  |



Show how a MUX with only 4 data inputs works

Assume $\mathrm{S} 1=0, \mathrm{SO}=1$. Draw the equivalent circuit in this case and explain the value of the output q .


## Extra-credit

 How many transistors are needed to build this MUX?

## Show how an S-R latch works

Unlike the one presented last class, this latch is made of NOR gates.
Assume $\mathbf{R}=\mathbf{0}, \mathbf{S}=1$. Show that:

- the latch is "set", i.e. the output/state X is 1
- the other output has the correct value of 0 (complement of $X$ )



## Show how an S-R latch works (continued)

To do in the notebook:
Examine the other 3 combinations of the inputs $\mathbf{R}, \mathbf{S}$ and explain if the latch operates correctly or not.
What is the "forbidden" combination of inputs for this version of the S-R latch?


