

Transistor: Building Block of Computers

- Microprocessors contain lots of transistors
- Intel 8086 (1978): 29 thousand
- Intel 80186 (1982): 55 thousand
- Intel 80386 (1985): 275 thousand
- Intel 80486 (1989): 1.1 million
- Intel Pentium (1993): 3.1 million
- Intel Pentium II (1998): 7.5 million
- Intel Pentium III (2001): 45 million
- Intel Pentium 4 (2006): 184 million
- Intel Core 2 Duo (2006): 291 million
- Intel Quad Core i7 (2011): 1.1 billion
- Intel 8-core Xeon (2012): 2.3 billion

Transistor: Building Block of Computers

- Logically, each transistor acts as a switch
- Combined to implement logic functions (gates)
- AND, OR, NOT
- Combined to build higher-level structures
- Adder, multiplexer, decoder, register, memory ...
- Adder, multiplier ...
- Combined to build simple processor
- LC-3


n-type MOS Transistor
- MOS = Metal Oxide Semiconductor
- two types: n-type and p-type


## - n-type

- when Gate has positive voltage, Gate $=1$ short circuit between \#1 and \#2 (switch closed)
- when Gate has zero voltage, open circuit between \#1 and 12 (switch open)

Terminal \#2 must be connected to GND (OV).


6

## Physical Transistor




Logic analyzer view of waveforms

Transistor Output (Actual)


Actual waveform is not ideal!

Copyight © The McGaw HIll Companies, Inc. Permisson required tor repocouction or of isplay.

## Logic Gates

- Use switch behavior of MOS transistors to implement logical functions: AND, OR, NOT.
- Digital symbols:
- recall that we assign a range of analog voltages to each digital (logic) symbol

- assignment of voltage ranges depends on electrical properties of transistors being used otypical values for "1": $+5 \mathrm{~V},+3.3 \mathrm{~V},+2.9 \mathrm{~V}$ ofrom now on we'll use +2.9 V


## 

## CMOS Circuit

- Complementary MOS
- Uses both n-type and p-type MOS transistors
- p-type
-Attached to + voltage
-Pulls output voltage UP when input is zero
- n-type -Attached to GND
-Pulls output voltage DOWN when input is one
- For all inputs, make sure that output is either connected to GND or to +, but not both!




## DeMorgan's Law

- Converting AND to OR (with some help from NOT) - Consider the following gate:


| $A$ | $B$ | $\bar{A}$ | $\bar{B}$ | $\bar{A} \cdot \bar{B}$ | $\overline{\mathrm{~A}} \cdot \overline{\mathrm{~B}}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |

Same as A OR B!

To convert AND to OR (or vice versa), invert inputs and output.

## More than 2 Inputs?

- AND/OR can take any number of inputs.
- $\mathrm{AND}=1$ if all inputs are 1 .
- $O R=1$ if any input is 1 .
- Similar for NAND/NOR.
- Can implement with multiple two-input gates, or with single CMOS circuit.



## 

## Summary

- MOS transistors are used as switches to implement logic functions.
- n-type: connect to GND, turn on (1) to pull down to 0
- p-type: connect to +2.9 V , turn on (0) to pull up to 1
- Basic gates: NOT, NOR, NAND
- Logic functions are usually expressed with AND, OR, and NOT
- DeMorgan's Law
- Convert AND to OR (and vice versa) by inverting inputs and output



## Building Functions from Logic Gates

- Combinational Logic Circuit
- output depends only on the current inputs
- stateless
- Sequential Logic Circuit
- output depends on the sequence of inputs (past and present)
- stores information (state) from past inputs
- We'll first look at some useful combinational circuits, then show how to use sequential circuits to store information.

