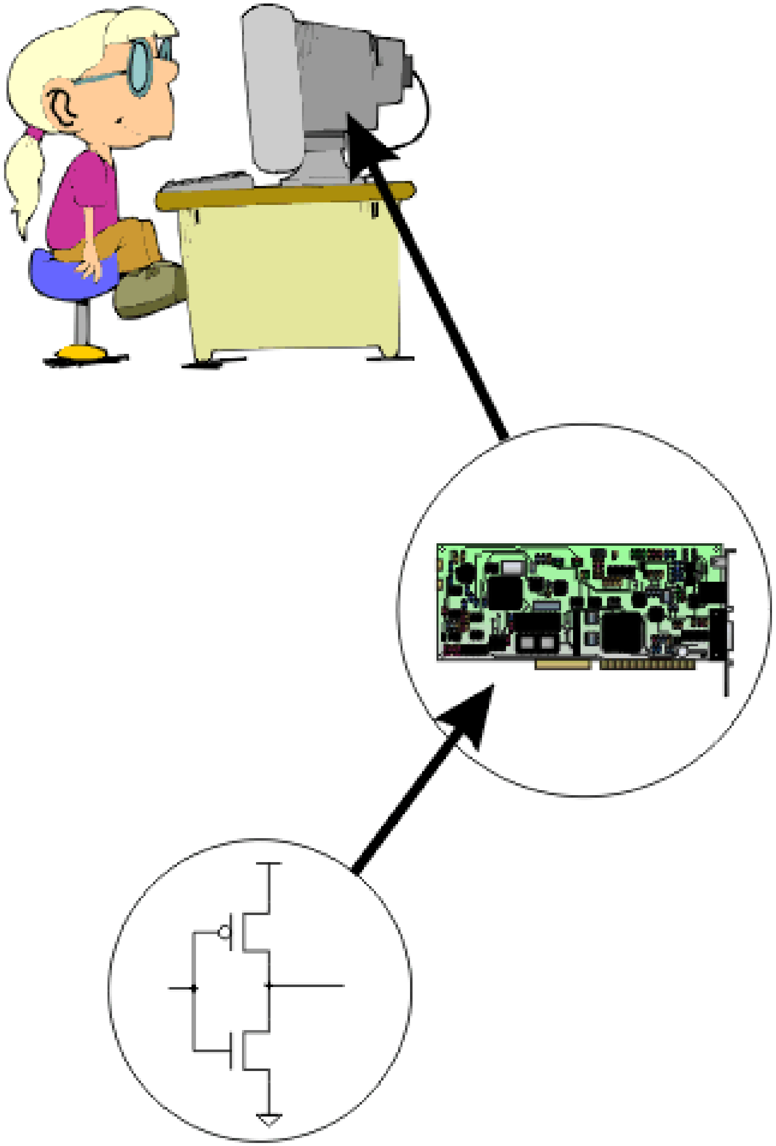
# Introduction to Computer Engineering

ENGR1202

Computer Engineering Lecture 1 Notes



# Engineers Have a Sense of Humor

Moo-shoe pork

# Computers are *Everywhere*

Q: *Where are computers today?*

On your desktop (of course!)

In your microwave oven

Controlling automobiles

In a GPS

In a camera

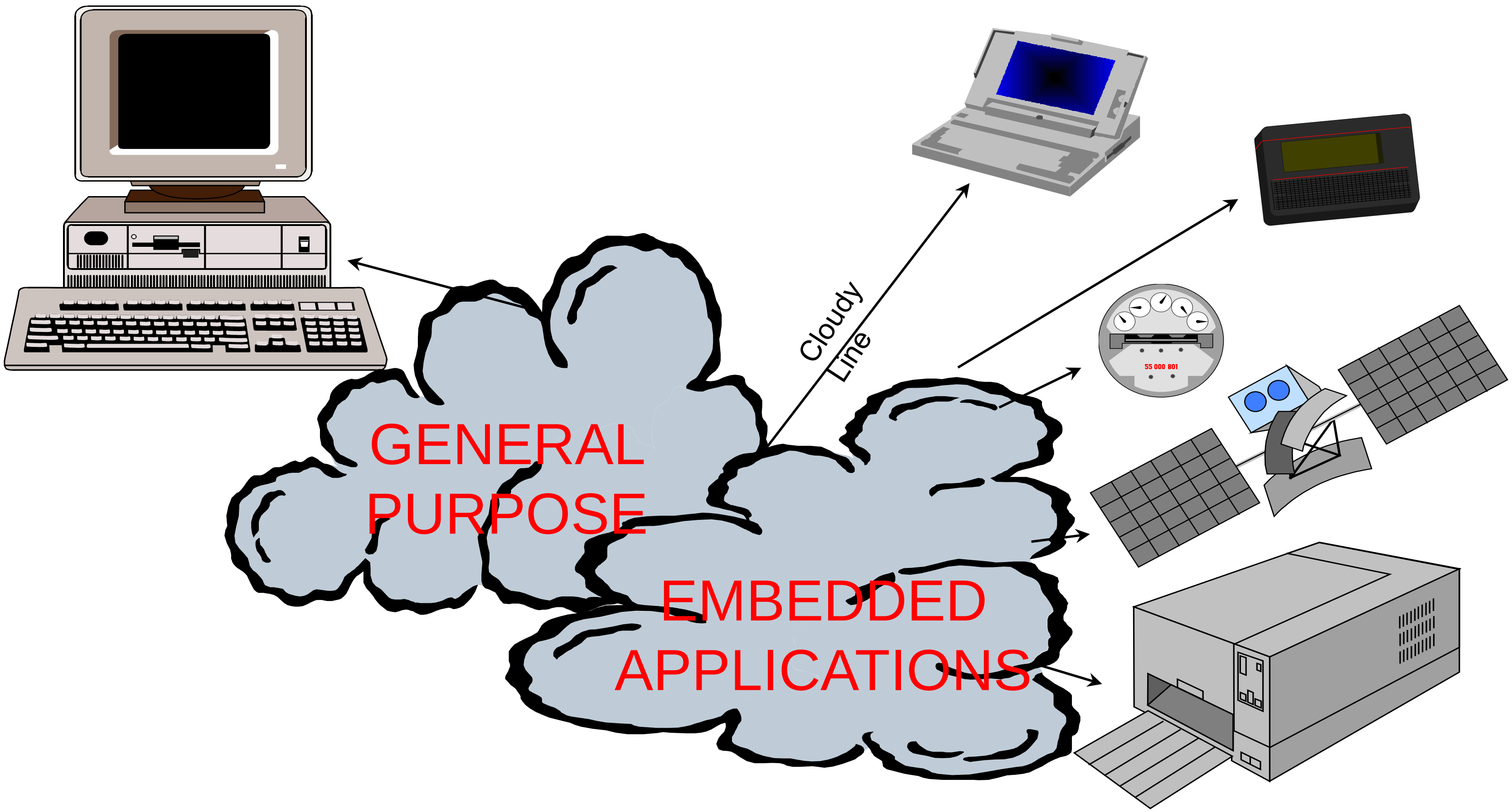
In a iPhone

In a Nintendo 3DS

In a Wii U . . .

Everywhere!


# What is Embedded?



# What is an Embedded System?

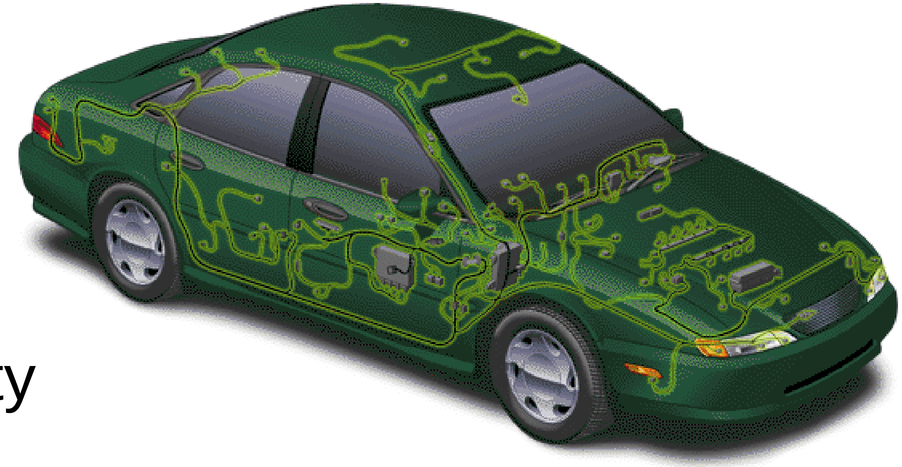
A microprocessor based device which has:

- Pre-defined, specific functions
- Constrained resources (memory, power)
- Application runs from ROM

Computer purchased as part of some other piece of equipment:

- Typically dedicated software (may be user-customizable)
- Often replaces previously electromechanical components
- Often no “real” keyboard
- Often limited display or no general-purpose display device

# A Customer View

Reduced Cost

Increased Functionality

Improved Performance

Increased Overall Dependability

# Designing a Microcontroller into a System

Power supply

Clock signal generator

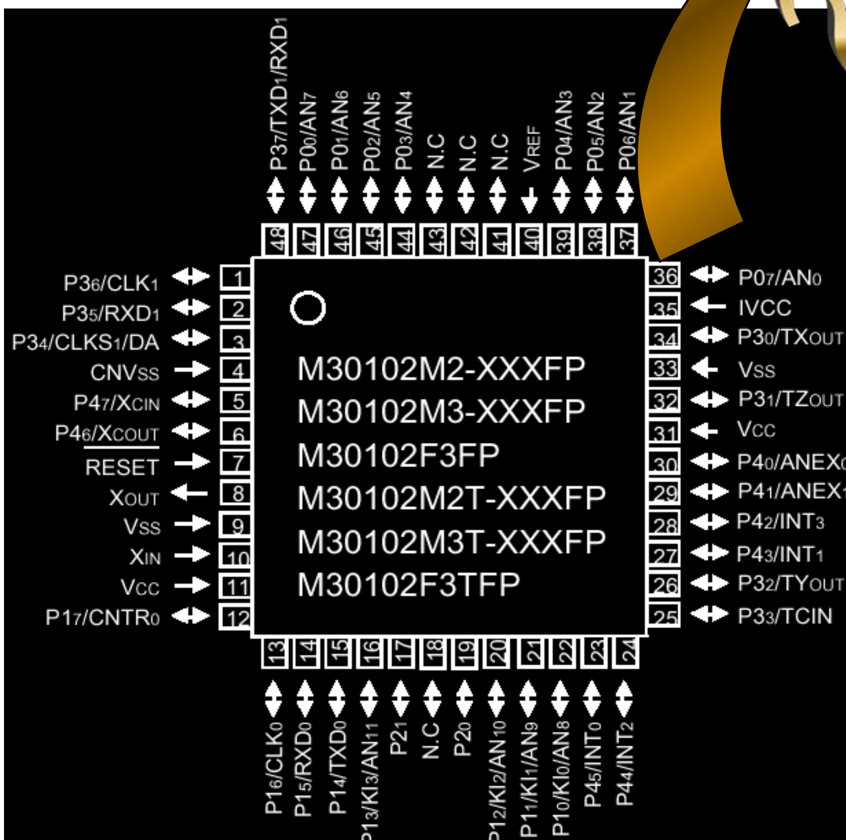
Reset controller

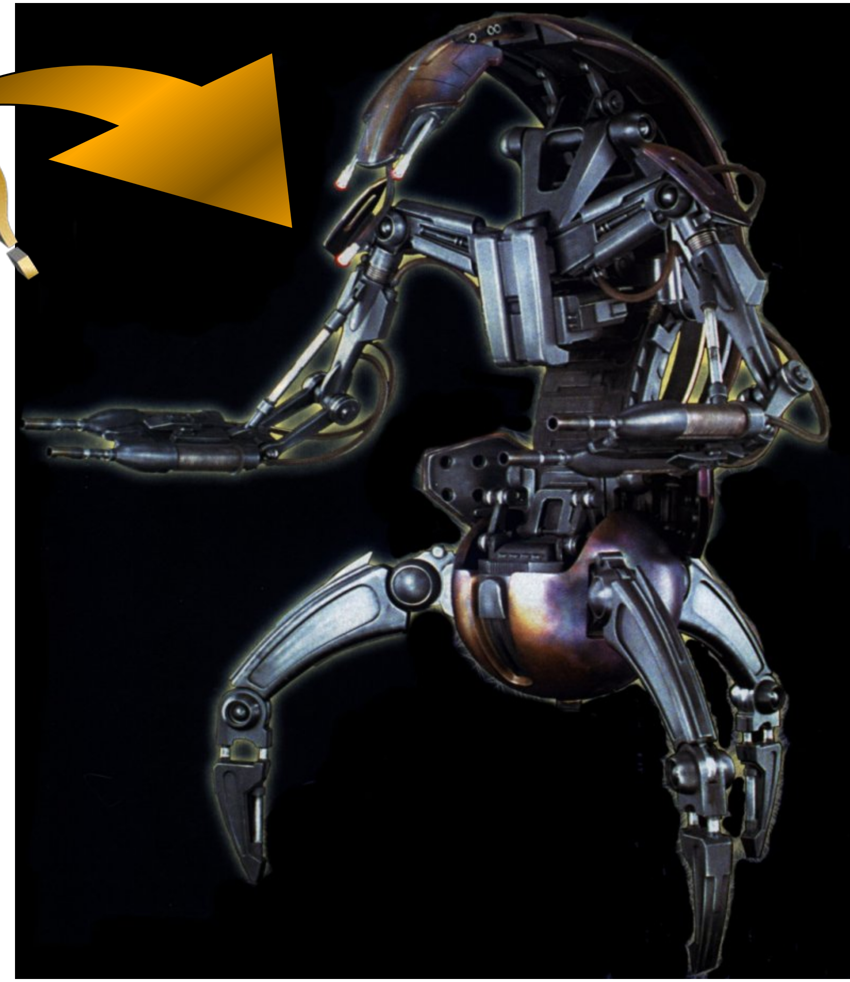
Memory

Digital interfacing

Analog interfacing

Communications



# How do we represent data in a computer?

At the lowest level, a computer is an electronic machine.

- works by controlling the flow of electrons

Easy to recognize two conditions:

1. presence of a voltage – we'll call this state "1"
2. absence of a voltage – we'll call this state "0"

Could base state on *value* of voltage,
but control and detection circuits more complex.

- compare turning on a light switch to measuring or regulating voltage

We'll see examples of these circuits in later chapters.

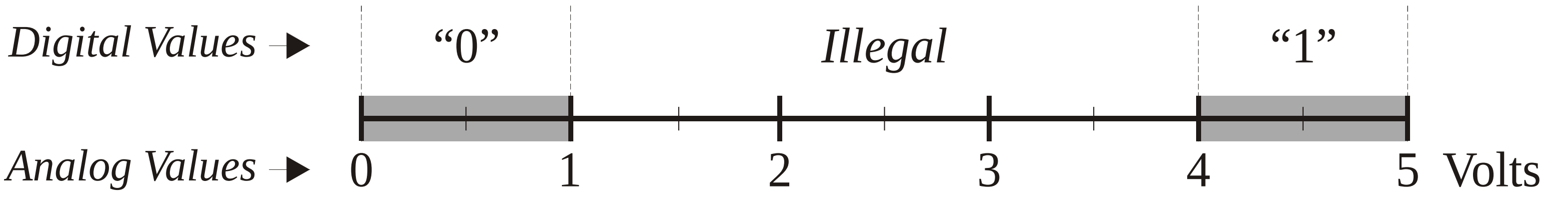
# Computer is a binary digital system.

Digital system:

- finite number of symbols

Binary (base two) system:

- has two states: 0 and 1

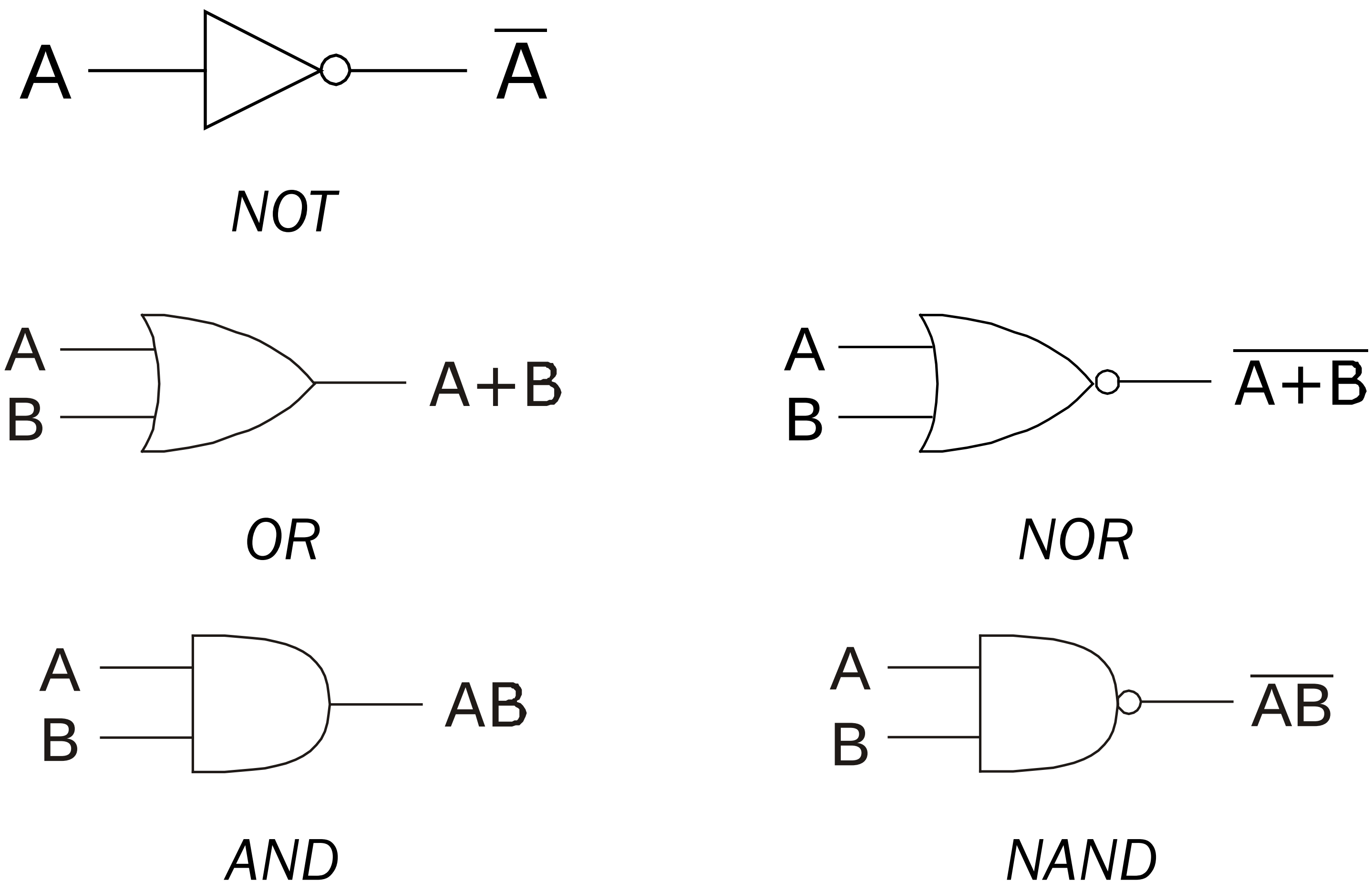



Basic unit of information is the *binary digit*, or ***bit***.

Values with more than two states require multiple bits.

- A collection of two bits has four possible states:
  00, 01, 10, 11
- A collection of three bits has eight possible states:
- *A collection of **n** bits has $\mathbf{2^n}$ possible states.*

# Basic Logic Gates



# Building a Truth Table

**AND**

| | |
|---|---|
| | |

**OR**

| | |
|---|---|
| | |

**NOT**

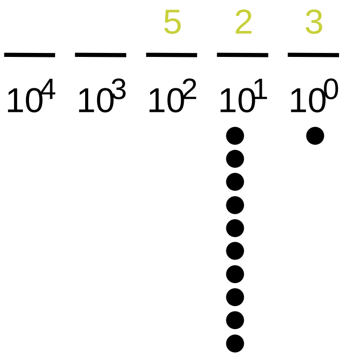
| | |
|---|---|
| | |

# How to Encode Numbers: Binary Numbers

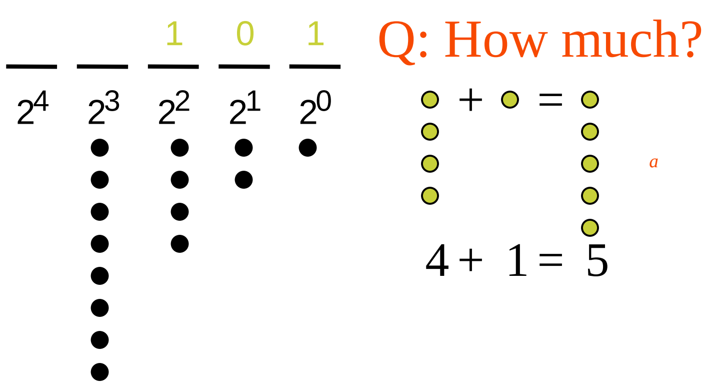
This slide from F. Vahid, *Digital Design*, 2007

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 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 power of 2



# How to Encode Numbers: Binary Numbers

This slide from F. Vahid, *Digital Design*, 2007

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 ten 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

a

# Converting from Decimal to Binary Numbers

This slide from F. Vahid, *Digital Design*, 2007

- 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).
- 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

Desired decimal number: **17**

| | | | | | |
|---|---|---|---|---|---|
| __ | __ | __ | __ | __ | __ |
| 32 | 16 | 8 | 4 | 2 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | __ | __ | __ | __ | __ | = 32 |
| 32 | 16 | 8 | 4 | 2 | 1 | too much |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | __ | __ | __ | __ | = 16 (17-16= 1) |
| 32 | 16 | 8 | 4 | 2 | 1 | ok, keep going |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | __ | = 8, 4, 2 |
| 32 | 16 | 8 | 4 | 2 | 1 | too much |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | =1-1=0 |
| 32 | 16 | 8 | 4 | 2 | 1 | DONE |

| | | | | | | |
|---|---|---|---|---|---|---|
| **0** | **1** | **0** | **0** | **0** | **1** | answer |
| 32 | 16 | 8 | 4 | 2 | 1 | |

# Converting from Decimal to Binary: Example

This slide from F. Vahid, *Digital Design*, 2007

Q: Convert the number “29” from decimal to binary

| A: Remaining quantity | Binary Number |
|---|---|
| 29 | 0/32 0/16 0/8 0/4 0/2 0/1 |
| 29 − 16 = 13 | 0/32 1/16 0/8 0/4 0/2 0/1 |
| 13 − 8 = 5 | 0/32 1/16 1/8 0/4 0/2 0/1 |
| | *8 is more than 7, can’t use* |
| 5 − 4 = 1 | 0/32 1/16 1/8 1/4 0/2 0/1 |
| 1 − 1 = 0 | 0/32 1/16 1/8 1/4 0/2 1/1 |

*a*

Done! 29 in decimal is 10111 in binary.

# Converting Decimal to Binary Practice

- Convert decimal 70 to binary

| $n$ | $2^n$ |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 4 |
| 3 | 8 |
| 4 | 16 |
| 5 | 32 |
| 6 | 64 |
| 7 | 128 |
| 8 | 256 |
| 9 | 512 |
| (K) 10 | 1024 |
| (M) 20 | 1048576 |

# More Converting Decimal to Binary Practice

- Convert decimal 255 to binary

| $n$ | $2^n$ |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 4 |
| 3 | 8 |
| 4 | 16 |
| 5 | 32 |
| 6 | 64 |
| 7 | 128 |
| 8 | 256 |
| 9 | 512 |
| (K) 10 | 1024 |
| (M) 20 | 1048576 |

# Converting from Binary ← → Hexadecimal

- Every four bits is a hex digit.
  - start grouping from right-hand side

**011101010001111010011010111**

| 011 | 1010 | 1000 | 1111 | 0100 | 1101 | 0111 |
|---|---|---|---|---|---|---|
| **3** | **A** | **8** | **F** | **4** | **D** | **7** |

Every hex digit is represented by 4-bits.

- start with 1$^{st}$ hex digit from right-hand side

| **1** | **F** | **8** | **C** | **5** | **3** | **E** |
|---|---|---|---|---|---|---|
| **1** | **1111** | **1000** | **1100** | **0101** | **0011** | **1100** |

*This is not a new machine representation, just a convenient way to write the number.*

# Converting from Hexadecimal to Decimal

- Every hex digit position has a base value
  - multiply the value at the position by the base value

**8 4 D 7**

$$8 \times 16^3 + 4 \times 16^2 + 13 \times 16^1 + 7 \times 16^0 =$$
$$8 \times 4096 + 4 \times 256 + 13 \times 16 + 7 \times 1 =$$
$$32768 + 1024 + 208 + 7 = \boxed{34007}$$

Another method is to convert to binary first (easy) then convert to decimal:

- $84D7h = 1000\ 0100\ 1101\ 0111_2 =$
  $1+2+4+16+64+128+1024+32768 = \boxed{34007}$

# Practice Converting from Hex to Decimal

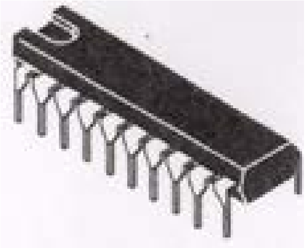
| 6 | F | 6 | A |
|---|---|---|---|

# Electronics Packaging

- There are several packaging technologies available that an engineer can use to create electronic devices.
- Some are suitable for inexpensive toys but not miniature consumer products, and some are suitable for miniature consumer products but not inexpensive toys.
- These packages have metal leads that are the conductive wire that connect electricity from the outside world to the silicon inside the package.
- Leads between packages are connected with small copper traces on a printed circuit board (PCB), and the package leads are soldered to the PCB.

# Examples of Electronics Packages

Dual In-line Package (DIP) Older technology, requires the metal leads to go through a hole in the printed circuit board.

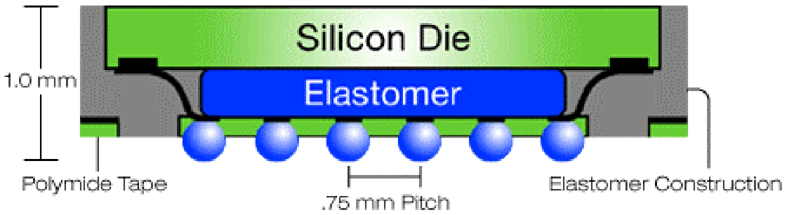



Dual Flat Pack (DFP) - A fairly recent technology, metal leads solder to the surface of the printed circuit board.

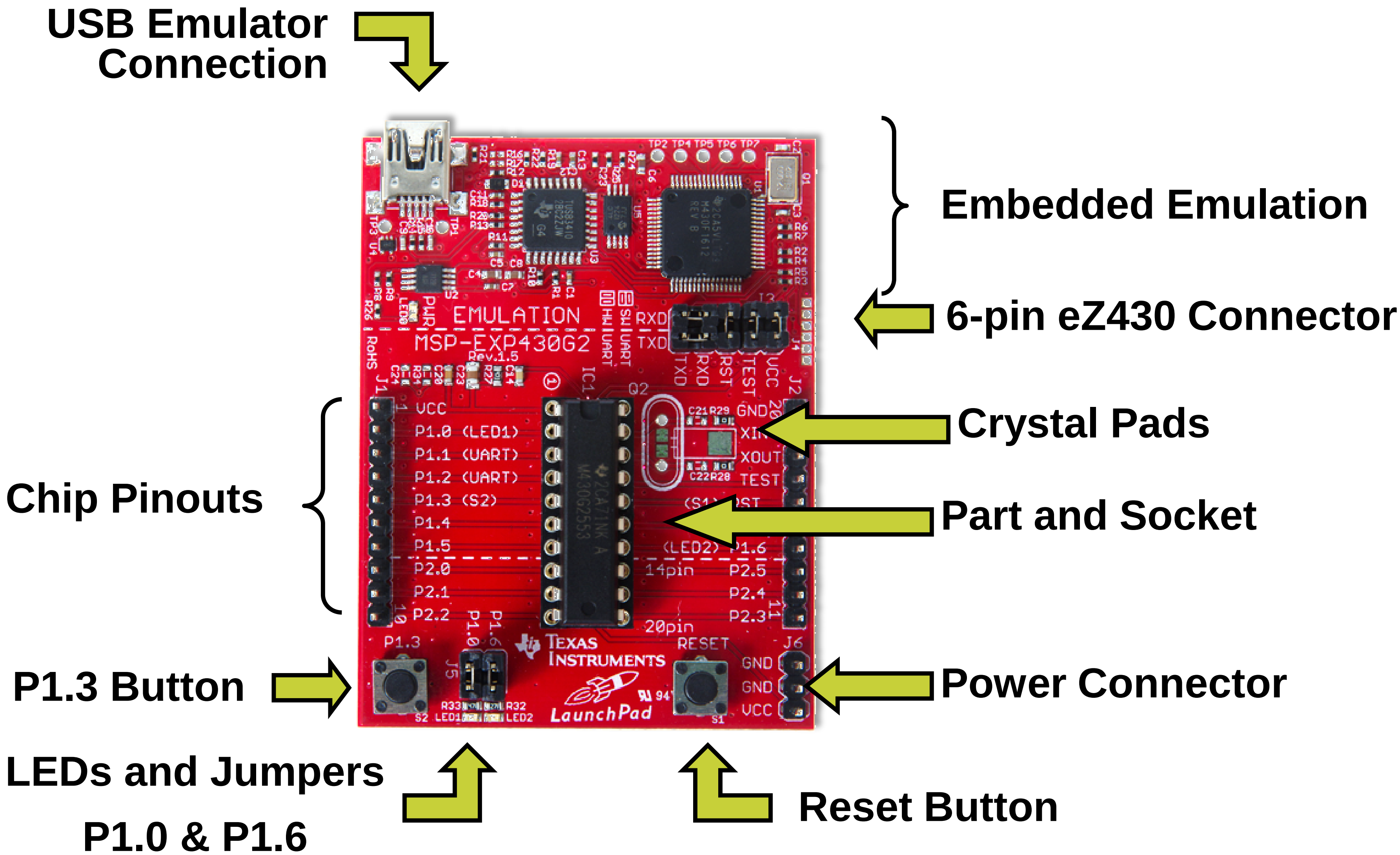

# Examples of Electronics Packages

Quad Flat Pack (QFP) - like the Dual Flat Pack, except here are metal leads are on four sides.

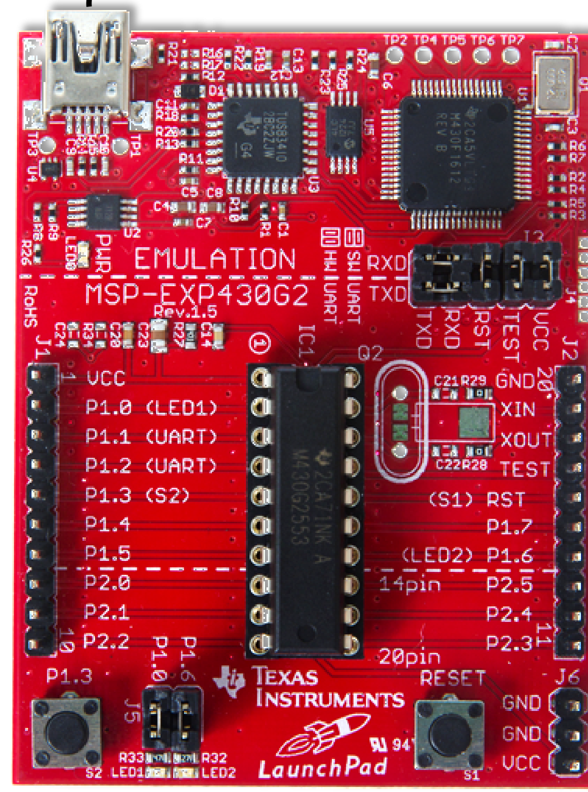
Ball Grid Array (BGA) - The connections to the component are on the bottom of the chip, and have balls of solder on these connections.

# LaunchPad Development Board



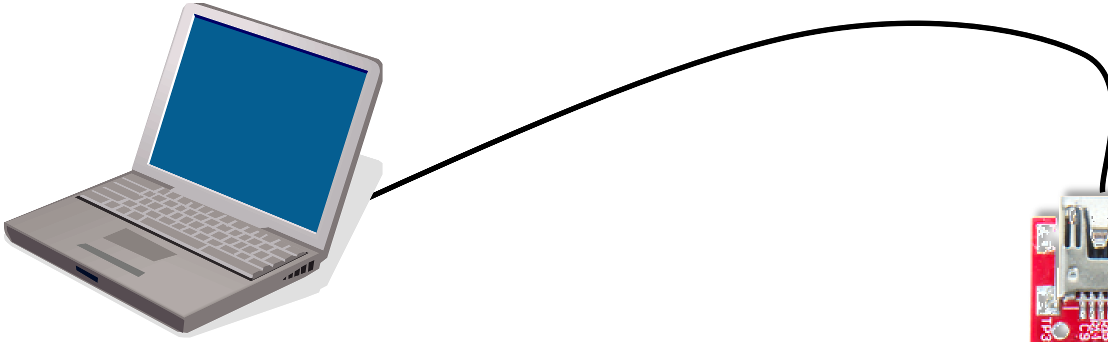
# Assignment 1: Setup

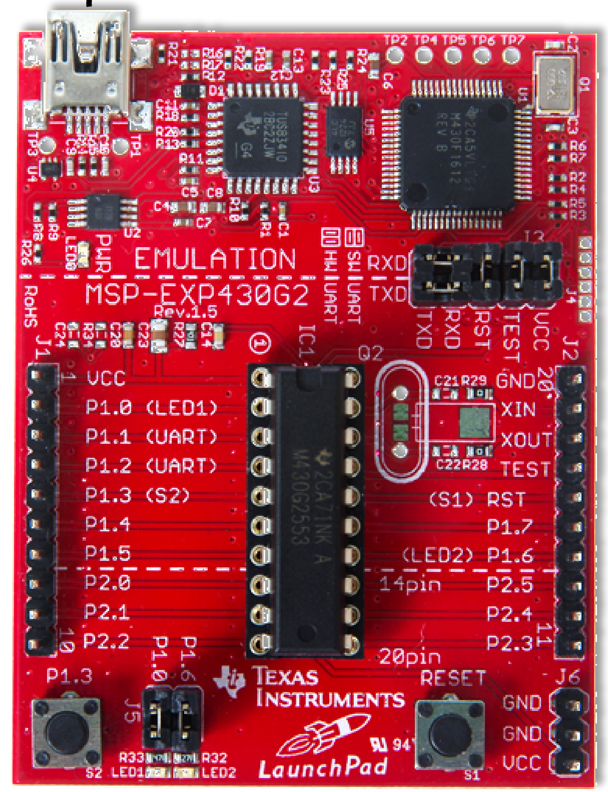



- **Download and install tools and documentation on your PC**
- **Review kit contents**
- **Connect hardware**
- **Test preloaded software**

**(Launchpad Quick Start Guide steps 1-3)**

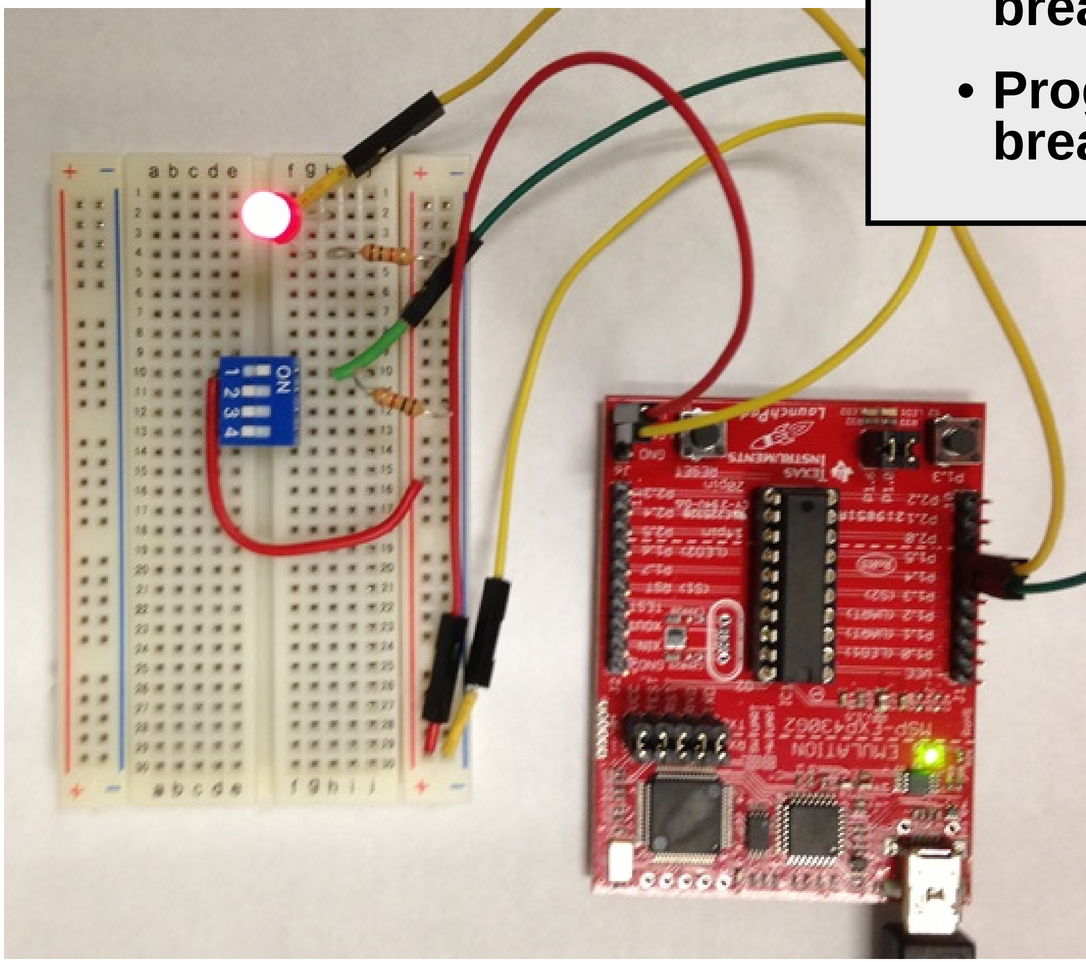
# Assignment 2: Program






- **Program operation of Launchpad LEDs**

# Assignment 3: Hardware and Software



- Wire breadboard
- Program operation of breadboard switch and LED
- Program operation of breadboard switch and LED

# Assignment 4: Setup

- Wire breadboard (solder board?)
- Program to play music or an annoying tune