

Overview

- What are embedded systems?
 - Computers masquerading as non-computers



Casio Camera
Watch



Nokia 7110
Browser
Phone



Sony
Playstation 2



Philips DVD player



Philips TiVo Recorder

Slide courtesy of Prof. Stephen A. Edwards of Columbia University

Characteristics of Embedded Software/Systems

- Software tightly coupled to hardware
- Software has specific purpose
- Software automatically starts at power on and runs until power off
- Interfaces: sensors and actuators

Embedded System Challenges

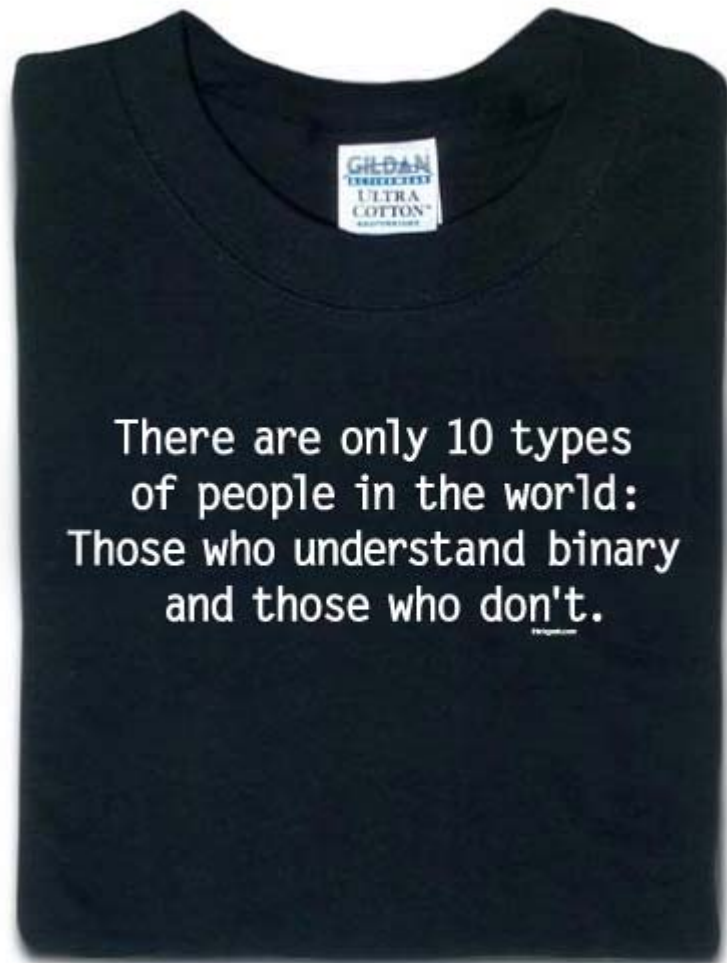
- Differs from general-purpose computing
 - Real-time constraints
 - Power constraints
 - Exotic hardware
 - Concurrency
 - Control systems
 - Signal processing
 - User interface
 - Laws of physics



SR-71

Slide courtesy of Prof. Stephen A. Edwards of Columbia University

Binary: Bits, Bytes, Words, ...



1 bit : 0/1

8 bits = byte

2 bytes = word

2 word = double word

8 bits => 256

16 bits => 64k

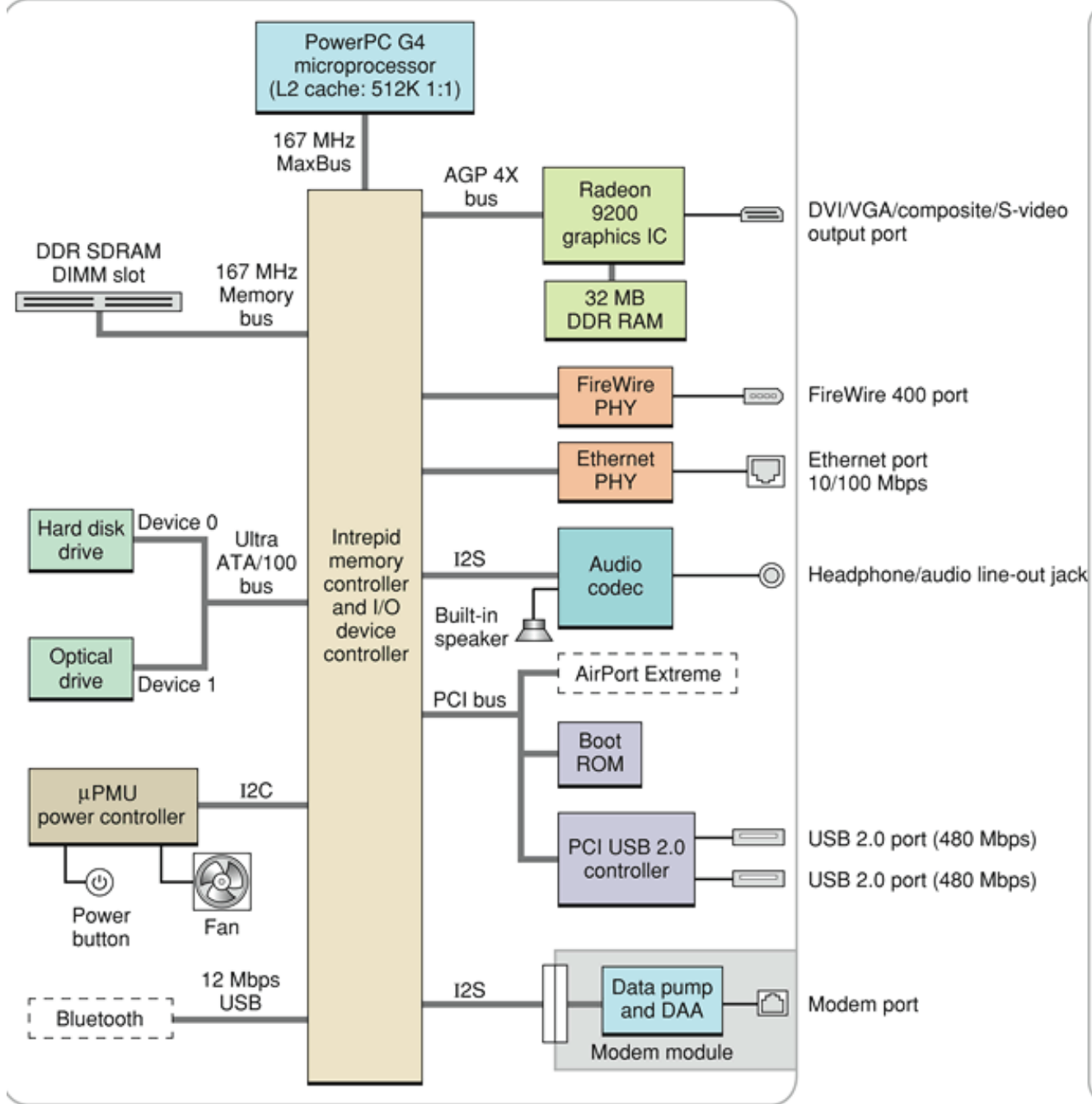
20 bits => 1M

24 bits => 16M

32 bits => 4G

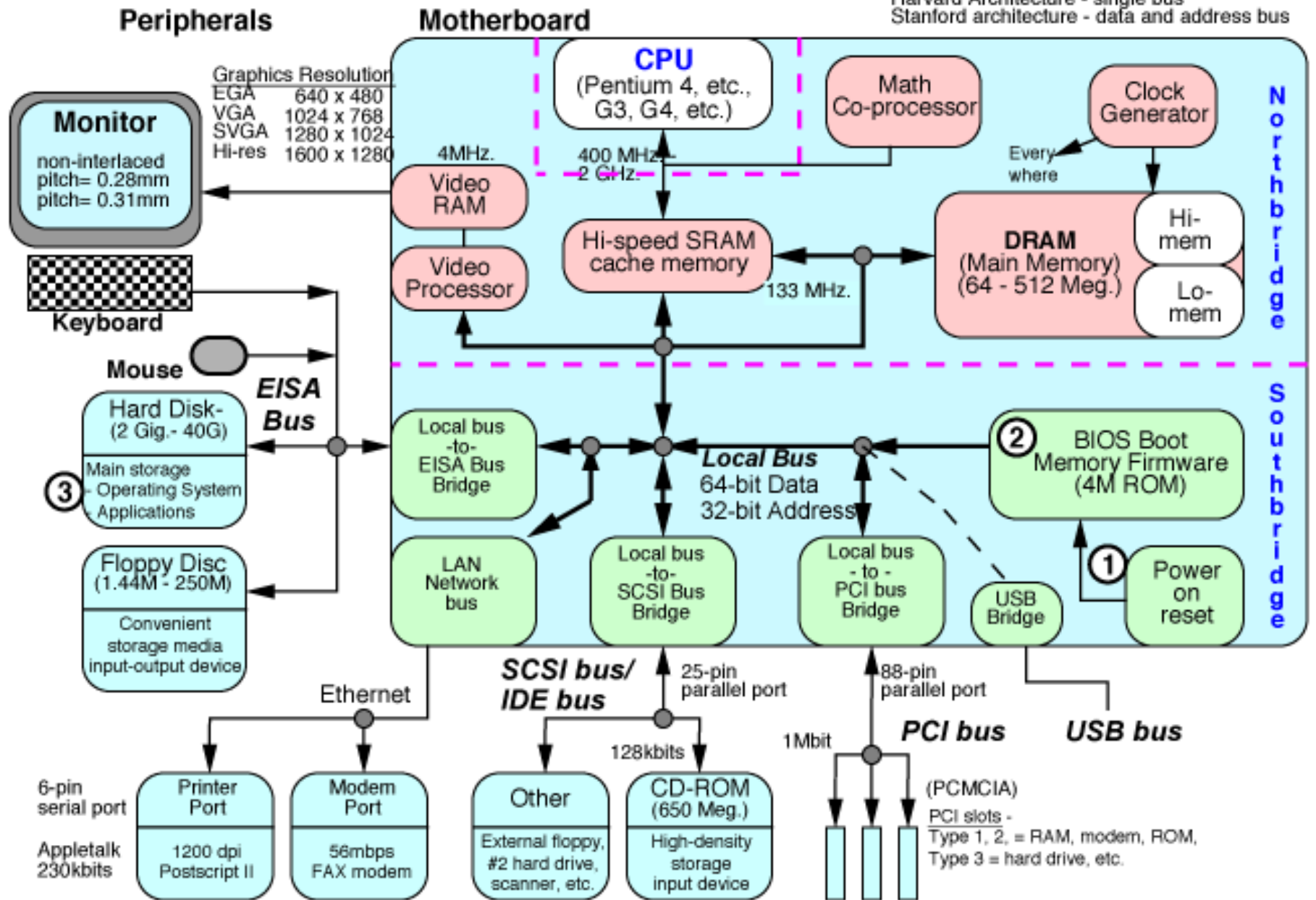
Early Microprocessors

- 1971 Intel 4004 - ~ 2000 Transistors
- Motorola 6800
 - > MOS 6502 Apple II (formed by staff from M)
 - > 68000 Mac, Amiga, Atari, Sun, Sili Graphics
- Intel 4004 -> 8008 -> 8080
 - > Zilog Z-80 Most popular ever, still made
 - > 8086 -> 8088 IBM PC



PC Block Diagram

Harvard Architecture - single bus
Stanford architecture - data and address bus



Interfaces

- Serial
 - USB, RS-232 / 485, FireWire (IEEE 1394), SPI, I2C
 - SATA
- Parallel
 - Parallel port, SCSI, IDE, EIDE, ATA
- Bus
 - ISA, EISA, PCI, PCI express, AGP
- Wireless
 - WiFi (802.11), WiMax (802.16), BlueTooth,

Languages

- Machine language
- Assembly language
- Compiled languages
- Intermediate language languages
- Interpreted languages

- Tradeoffs

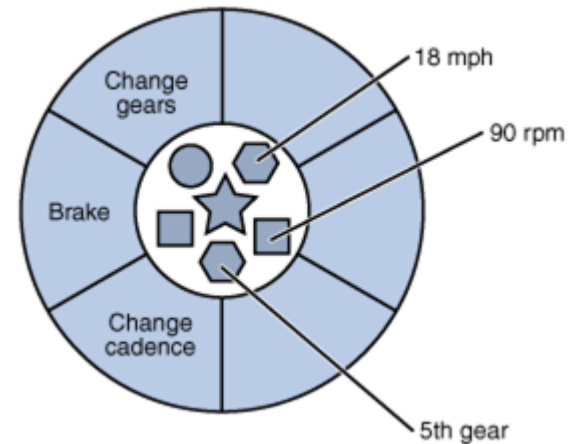
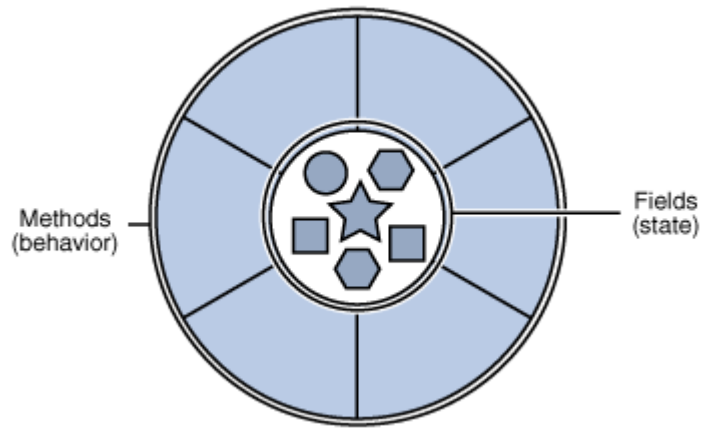
Object Oriented Programming

- Information Hiding
- Abstraction
- Encapsulation
- Polymorphism
- Inheritance

Concepts and Terms

- Object
- Class
- Instance / Instantiate
- Interface: Message / Method / Function

Object: State and Behavior



Example Class

Header File: Bicycle.h

```
class Bicycle
{
    private int cadence;
    private int speed;
    private int gear;

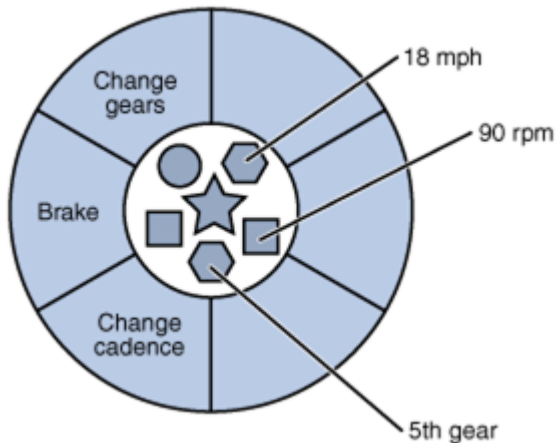
    public Bicycle();

    public void changeCadence(int newValue);

    public void changeGear(int newValue);

    public void speedUp(int increment);

    public void applyBrakes(int decrement);
}
```



Implementation File: Bicycle.cpp

```
#include "Bicycle.h"

Bicycle::Bicycle()
{
    cadence = 0;
    speed = 0;
    gear = 1;
}

public void
Bicycle::changeCadence(int newValue)
{
    cadence = newValue;
}

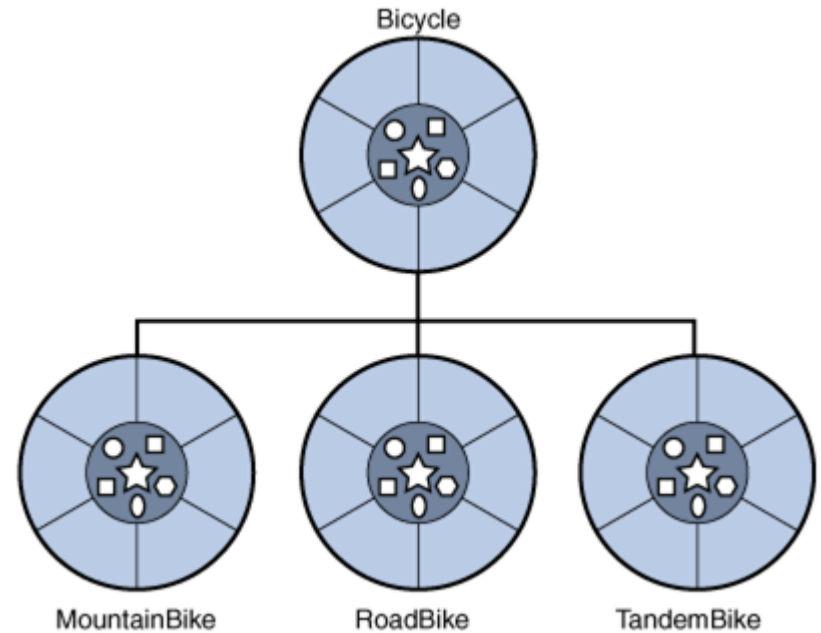
public void
Bicycle::changeGear(int newValue)
{
    gear = newValue;
}

public void
Bicycle::speedUp(int increment)
{
    speed += increment;
}

public void
Bicycle::applyBrakes(int decrement)
{
    speed -= decrement;
}
```

Abstraction, Inheritance and Polymorphism

- Bicycle is the abstract idea of a two wheel, person-powered vehicle
- MountainBike, RoadBike, and TandemBike inherit the characteristics of a Bicycle but are more specific
- If you can ride any Bicycle, you can probably ride all of the others - Polymorphism



Classes and Objects

Header File: Bicycle.h

```
class Bicycle
{
    private int cadence;
    private int speed;
    private int gear;

    public Bicycle();

    public void changeCadence(int newValue);
    public void changeGear(int newValue);
    public void speedUp(int increment);
    public void applyBrakes(int decrement);
}
```



<http://janeheller.mblogs.com/bicycle-yellow.png>

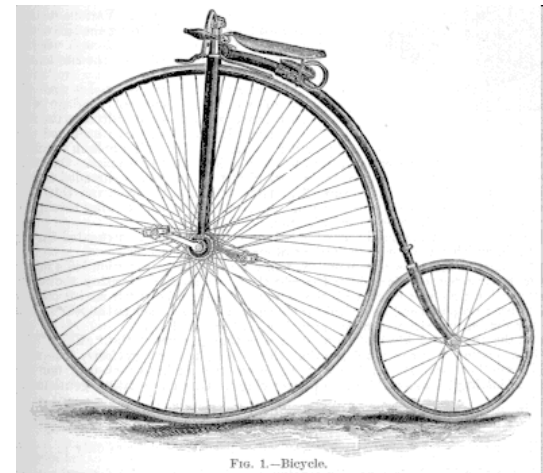
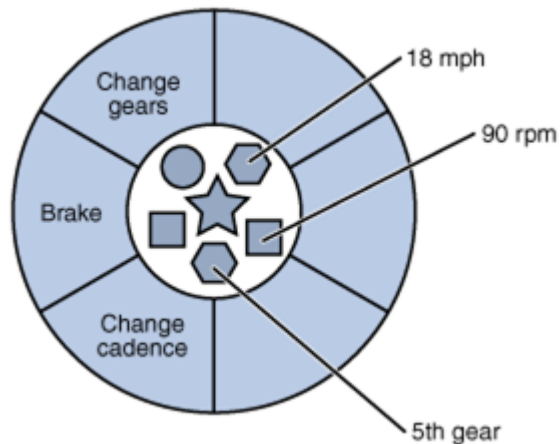


FIG. 1.—Bicycle.

<http://www.uh.edu/engines/bicycle.gif>

Grading

- Homework
- Midterm exam (early November)
- Project
 - Project idea – due Sept 1st
 - Project white paper – due Sept 15th
 - Literature survey – due Oct 27th
 - Final presentation – last class period
 - Final project report – last day of classes