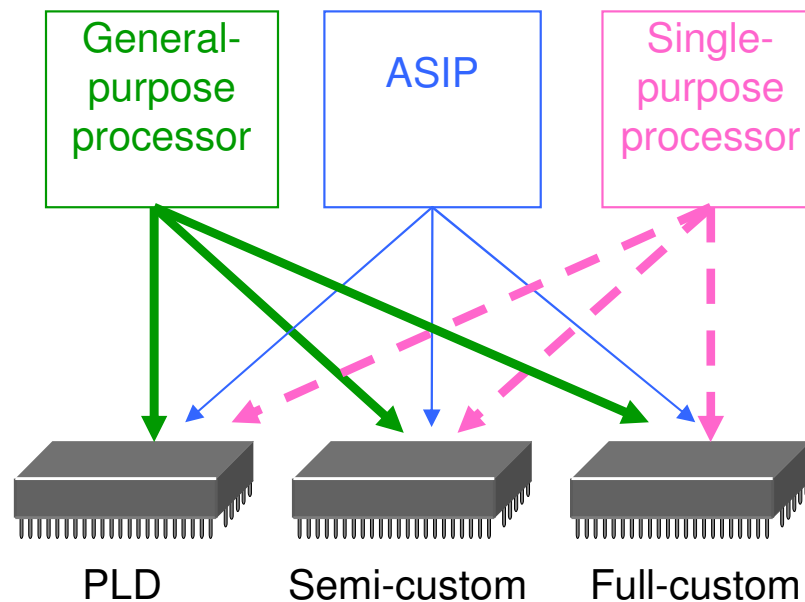

Review...

Summary of chapter 1

- Chapter 1 (Introduction):
 - Know the terms and ideas behind the metrics that measure an embedded system.
 - Unit cost, NRE, latency, throughput, etc...
 - Moore's law and it's effect
 - Know the time-to-market, designer-productivity gap, and mythical man-month terms
 - Know the IC technologies and their trade-offs
 - Know the different types of processors and their trade-offs.

Recall...

A processor (any processor) is a way of working with and transferring data, it can be implemented in any IC technology...

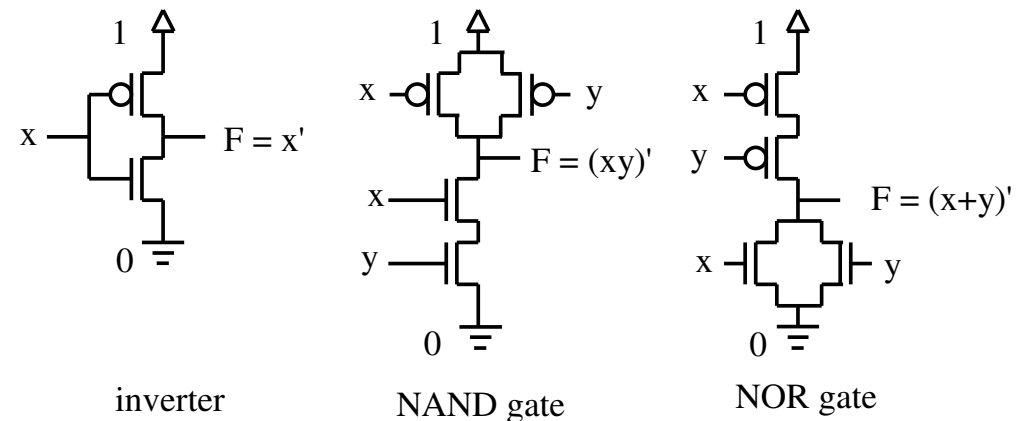
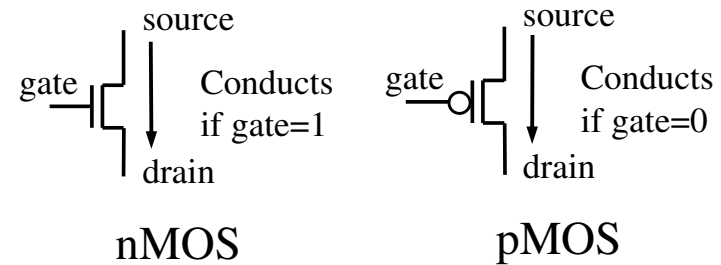


Summary of chapter 2

- Chapter 2 (Single Purpose Processors):
 - Know how to evaluate and create basic gates using transistors
 - Know the difference between combinational and sequential components
 - Know how to create a finite state machine with a datapath from a given algorithm

CMOS transistor implementations

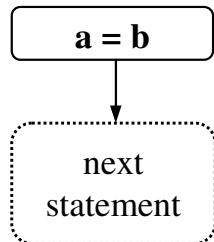
- Complementary Metal Oxide Semiconductor
- We refer to logic levels
 - Typically 0 is 0V, 1 is 5V
- Two basic CMOS types
 - nMOS conducts if gate=1
 - pMOS conducts if gate=0
- Basic gates
 - Inverter, NAND, NOR



State diagram templates

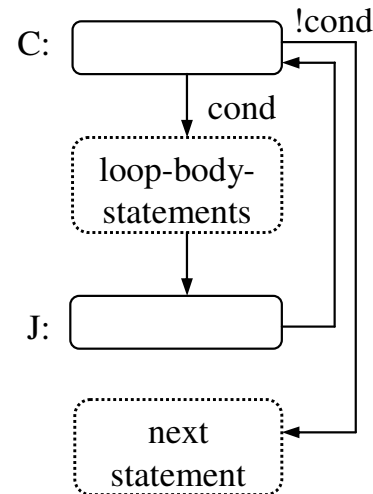
Assignment statement

```
a = b  
next statement
```



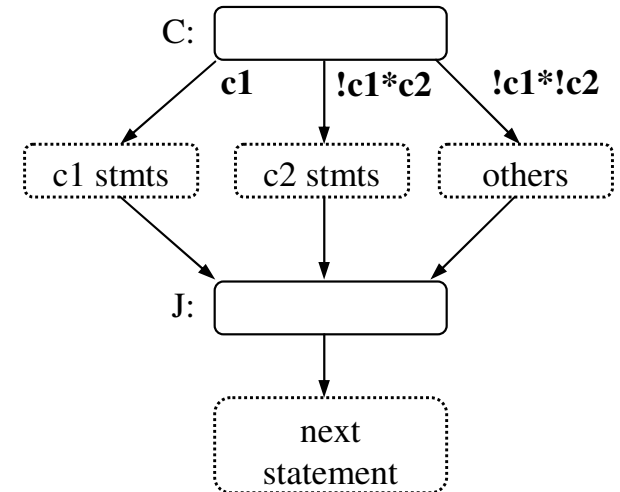
Loop statement

```
while (cond) {  
    loop-body-  
    statements  
}  
next statement
```



Branch statement

```
if (c1)  
    c1 stmts  
else if c2  
    c2 stmts  
else  
    other stmts  
next statement
```



Creating the datapath

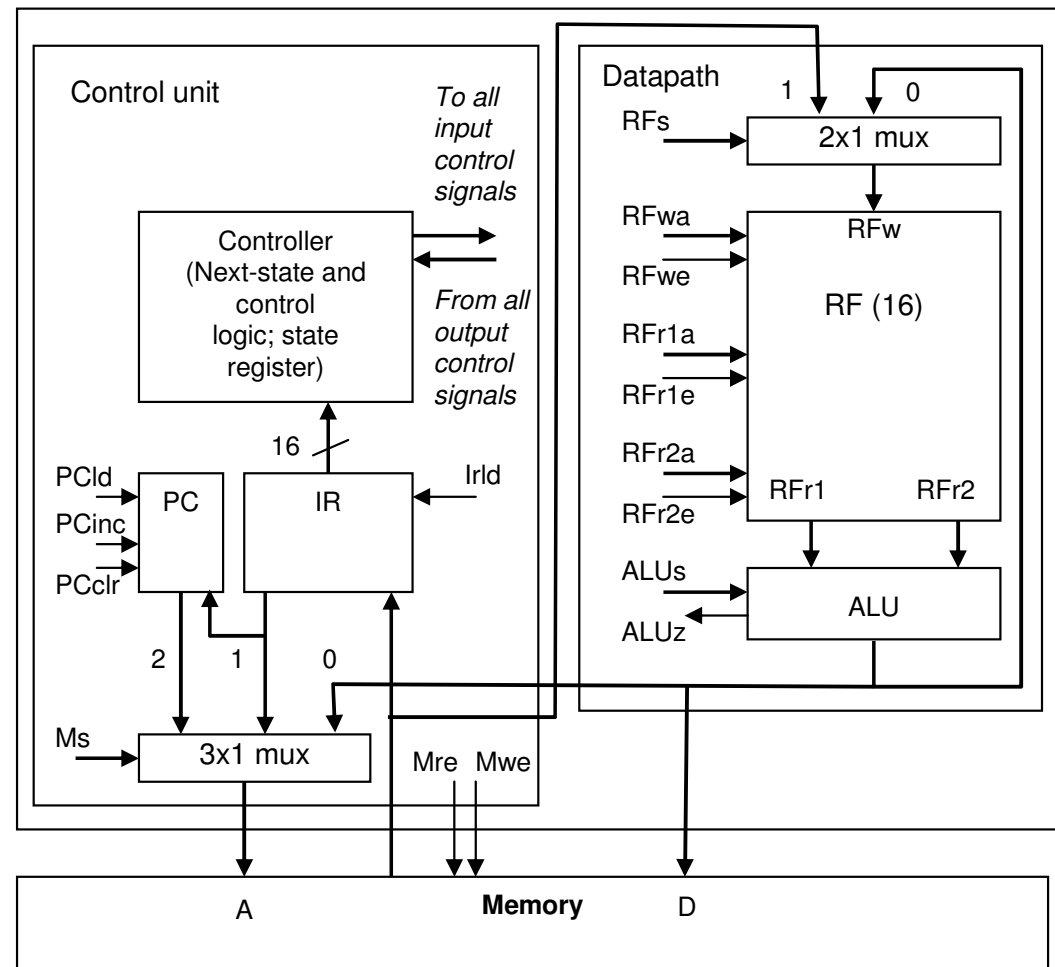
1. Create a register for any variable
2. Create a functional unit for each arithmetic operation
3. Connect the ports, registers and functional units
 1. Based on reads and writes
 2. Use multiplexers for multiple sources
4. Create unique identifier
 - for each datapath component control input and output

Summary of chapter 3

- Chapter 3 (General Purpose Processors):
 - Know terms, like: pipelining, VLIW, Princeton architecture, Harvard architecture
 - Know how to program in assembly
 - Know the main components that make up a general purpose processor
 - Know the different tools available for development and debugging
 - Instruction Set Simulator, Emulator, Compiler, etc...

Architecture of a Simple Microprocessor

- Storage devices for each declared variable
 - register file holds each of the variables
- Functional units to carry out the FSM operations
 - One ALU carries out every required operation
- Connections added among the components' ports corresponding to the operations required by the FSM
- Unique identifiers created for every control signal



Summary of chapter 4

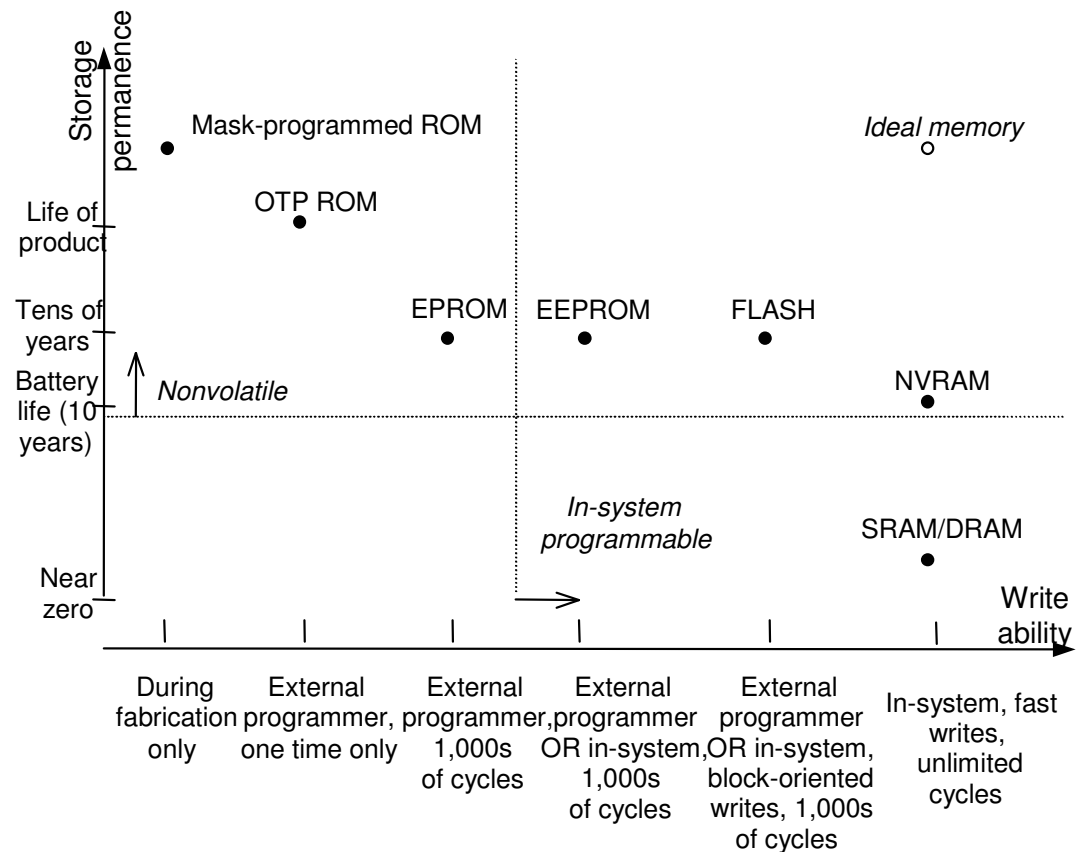
- Chapter 4 (peripherals):
 - Timers, counters, watchdogs
 - What is the resolution and range of a given timer? What are watchdogs for?
 - UART's (Universal Asynchronous Receiver Transmitter)
 - PWM (Pulse Width Modulator)
 - How can you control a motor's speed using a PWM?
 - D/A and A/D converters
 - remember: $e/V_{\max} = d/(2n - 1)$ and keep in mind how to convert when the range does not start at zero
 - Other peripherals (LCD controller, Stepper controller, Keypad Controller)

Summary of chapter 5

- Chapter 5 (memory):
 - Memory Write Ability and Storage Permanence
 - What kind of memory has excellent storage permanence?
What kind has excellent write-ability?
 - Common Memory Types
 - Know the trade-offs of the different types of memory
 - Composing Memory
 - How do you make a 16-bit wide 32K deep memory using 8-bit wide 8K memories?
 - Memory Hierarchy and Cache
 - Look at the quiz on cache

Write ability/ storage permanence

- Traditional ROM/RAM distinctions
 - ROM
 - read only, bits stored without power
 - RAM
 - read and write, lose stored bits without power
- Traditional distinctions blurred
 - Advanced ROMs can be written to
 - e.g., EEPROM
 - Advanced RAMs can hold bits without power
 - e.g., NVRAM
- Write ability
 - Manner and speed a memory can be written
- Storage permanence
 - ability of memory to hold stored bits after they are written

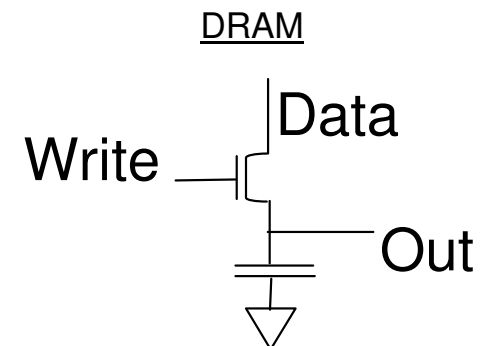
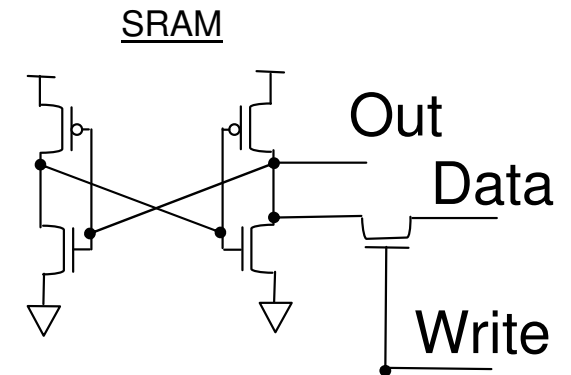


Write ability and storage permanence of memories, showing relative degrees along each axis (not to scale).

Basic types of RAM

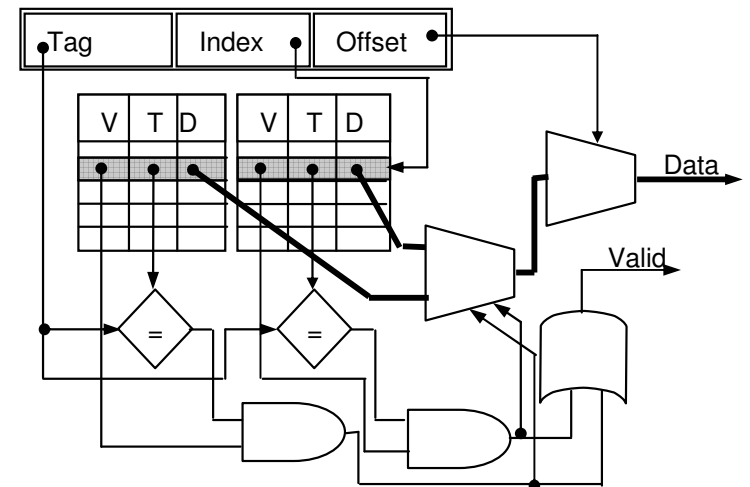
- SRAM: Static RAM
 - Memory cell uses flip-flop to store bit
 - Requires 6 transistors
 - Holds data as long as power supplied
- DRAM: Dynamic RAM
 - Memory cell uses MOS transistor and capacitor to store bit
 - More compact than SRAM
 - “Refresh” required due to capacitor leak
 - word’s cells refreshed when read
 - Typical refresh rate 15.625 microsec.
 - Slower to access than SRAM

memory cell internals



Set-associative mapping

- Compromise between direct mapping and fully associative mapping
- Index same as in direct mapping
- But, each cache address contains content and tags of 2 or more memory address locations
- Tags of that **set** simultaneously compared as in fully associative mapping
- Cache with set size N called N-way set-associative
 - 2-way, 4-way, 8-way are common



Summary of chapter 6

- Chapter 6:
 - Interfacing basics (also remember parity which was covered in 4)
 - Microprocessor interfacing
 - I/O Addressing
 - **Interrupts**
 - Direct Memory Access
 - Arbitration
 - Hierarchical buses
 - Protocols
 - Serial
 - Parallel
 - Wireless

What kind of questions might you see from Chapter 6?

- Look at F'03 quarter's final, which only covered the second half of the course.
- Consider what you did on the homework and quiz.
- Don't worry about ISA bus in book
- Know the concepts:
 - interrupts
 - DMA
 - arbitration
 - The basics of the protocols covered

Summary of Chapter 7...

- There won't be specific questions about our digital camera, but there may be questions about the lessons learned. Such as:
 - Going from floating point to fixed point saves time
 - What is system level modeling and how can it save design time?
 - What is a specification for?
 - What is quantization?
 - What is Huffman encoding used for?

Finally...

Fill out teacher evaluations, and good luck on the final!