Memory Mapped IO

(and the CerfBoard)
The problem

- How many IO pins are available on the 8051?
- What if you are using interrupts, serial, etc...?
- We want a consistent interface to I/O devices
External Data Memory

- 4k byte RAM chip

- Interface:
  - Bi-directional data bus
  - Address bus. How big?
  - /wr and /rd
  - /ce = Chip Enable
The 8051 interface

- P0 does double duty
  - When ALE is high, P0 is the lower 8 bits of the address
  - When it’s low, P0 is the data bus
  - We need a latch
- P3.6 = /wr
- P3.7 = /rd
Block Diagram

8051

Latch

RAM

P0

ALE

P2

/rd /wr

Data

A[0:7]

A[7:8]

/wr /rd
External Data Memory Read

ALE’s falling edge
Latches the address

When /rd is low the
Data bus must be stable
External Data Memory Write

ALE’s falling edge
Latches the address

Data output is stable
While /wr is low
Use the MOVX instruction to access external data memory

MOV R0, #external_address
MOVX A, @R0  # uses only 8-bit address for external RAM

Or
MOV DPL, #external_address_high
MOV DPH, #external_address_low
MOVX A, @DPTR;
Hooking up I/O Devices

- Why not put I/O stuff in the address space?
- Let’s hook up a DAC (Digital to Analog Converter)
  - 4 analog outputs (address bits?)
  - 8 data bits
  - We can only write to it
Memory Mapped I/O

How do we use it?

MOV A, 45 ; what we want to send
MOV R0, 1 ; select output #2
MOVX @R0, A ; write it

What if we did:

MOV R0, 4

What output will we write to?
More devices

- More DACs
- RAM
- ADC (Analog to Digital)
- LCD
- Keyboards
- More IO pins
- Disk drives
The CerfBoard
Overview

- StrongARM 1100 processor
  - 200mhz
- Ethernet, USB
- 32MB RAM, 16meg Flash ROM
- Compact Flash slot
- 3 RS232 Serial Ports (one for console)
- 16 GPIO pins
**Memory Mapped I/O**

- *Lots* of devices are memory mapped on the CerfBoard
  - GPIO registers
  - Power Management Registers
  - Serial Registers
  - Interrupt Control Registers
  - Compact Flash