Announcement

- Last time
  - Intel Assembly Code Basics
- Today
  - Brief review of using DS to form 20 bit physical address
  - Programming A/D & D/A devices
  - Next Lecture: Interrupts

Review: Direct (memory) Addressing

The physical address is 20 bits, but the data segment register has only 16 bits. How can we create 20 bit address using 16 bit DS register?

Intel's solution is: 1) shift the DS value 4 bits to the left (16 + 4 = 20)
2) add a 16 bit offset to address 64k locations

Example: How to move 10H to memory location 17000H? Suppose that we would like the segment to start at 10000H.

- MOV AX, 1000H
- MOV DS, AX (MOV DS,1000H is not allowed)
- MOV AX, 10H
- MOV [7000H], AX

16 bit DS Content	**1000 H**
16 bit offset	**7000H**
20bit BUS Address Lines	**17000H**

Review: I/O Port Addressing

The Intel architecture separates I/O port addresses with memory addresses, although memory mapped I/O is also supported.

- IN AL, 40H or OUT 40H, AL //read or write to port 40H

In this mode, only 00H to FFH (256) addresses can be addressed. We can use registers to address up to 65,536 addresses.

- IN AL, DX //8 bit data is input from port whose address is stored in DX
- OUT DX, AL //8 bit data is output from AL to port whose address is stored in DX

Review: Inline Assembly Code

- Given a new external hardware device, assembly code are needed to address their registers. We can embed the need assembly code in C to create a C function:

```
hw_out(port, val) // the port address and the 8 bit value that you want to send out
unsigned int port;
char val;  // used to hold 1 byte of data
{asm{
        mov EDX, port[EBP] //move port address to register EDX
        mov AL, val[EBP] //move value to lower byte of the accumulator
        out  DX, AL //send value to the location pointed by 16 bit port address in DX.
       }
}
```

Note: by convention, during a function call f(x), the parameter x stored on the stack is referenced as x[EBP].

A/D & D/A Card Architecture

- A modern A/D and D/A card has multiple ports starting from a selectable base address. These ports are organized as
  - data register
  - control and configuration
  - status registers
- In the lab, the DAS-1600 Card's base address is set at 300H (using jumpers). Some of the entries in the port function table in the manual is as follows (used by Lab 1):

Address	Read Function	Write Function
Base	A/D (bits 0 - 3) (stores low byte bits 4..7)	starts A/D conversion
Base+1	A/D (bits 4 - 11) (stores high byte bits 0..7)	N/A
Base+2	Channel Mux selection	Set Channel Mux select
Base +4	N/A	D/A Ch 0 bits 0 - 3
Base +5	N/A	D/A Ch 0 bits 4 - 11
Base + 8	status	clear interrupt
Base + 9	status	sets card configuration

High Byte and Low Byte

- The A/D and D/A conversion logic supports 12 bit conversion. However, its data register is only 8 bit wide. Thus, it uses 2 registers. For example, input Ch 0's 12 bits are stored in Base and Base + 1.

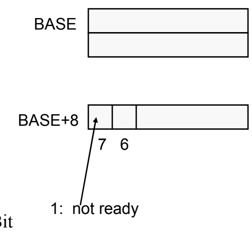
Base	3	2	1	0	x	x	x	x
Base + 1	11	10	9	8	7	6	5	4
	7	6	5	4	3	2	1	0

- You need to
  - Get HighByte from base+1 (bits 4 .. 11 of the 12 bit data)
  - data = (HighByte << 8) to move the data into high byte
    = 11 10 9 8 7 6 5 4 x x x x x x x x
  - Get LowByte from base (bits 0 .. 3 of the 12 bit data plus 4 bits of junk
  - data = (data | LowByte)
    = 11 10 9 8 7 6 5 4 3 2 1 0 x x x x
  - data = (data >> 4) to get rid of the 4 bits of junk
    = 0 0 0 0 11 10 9 8 7 6 5 4 3 2 1 0

Control and Status Register

- We need to initialize the card
  - Set BASEADDR+9 to 0 disable interrupt and DMA.
  - Set BASEADDR+8 to 0 clear trigger flip flop for interrupt
  - Set BASEADDR+2 to 0 select analog input channel 0 (pin 37).

- To sample the analogy voltage and starts the A/D conversion
  - Set BASEADDR to 0



- Once, the A/D conversion starts, your program must wait for the A/D conversion to complete. Bit 7 of BASEADDR+8 tells us whether the conversion is complete. Thus we AND the value with 0x80 (80H) to get Bit 7 and wait until the conversion is done.

- `While(hw_input(BaseADDR+8) & 0x80 != 0) {}`
- `((BaseADDR+8) &(10000000))!=0`
- `Note: for certain cards, you need to re-initialize it for every read. Does your version of DAS1600 needs to do that? Check it out in your lab.`

Digital Representation of Analog Voltage - 1

- The N = 12 bits can be configured by jumpers/switches to represent different analog voltage ranges,
  - e.g., -2.5 V to +2.5 V, -5 V to +5 V and -10 V to +10 V.

- What are these ranges for? What should be the rule in picking a range?

The General Conversion Rule

- Covert the analogy voltages to a N-bit digital representation is a special case of measurement conversion.
- Let's take the range between water freezing and boiling points at sea level. The measure using Celsius is from 0 to 100 while the measure using Fahrenheit is from 32 to 212.
- How can we convert from Celsius to Fahrenheit? Hint: The general conversion rule is derived from the observation that two different linear measures measure the same physical quantity must be proportional to each other. For example, 10% in range_1 map to 10% in range_2, and 70% in the range_1 map to ______% in range_2.
- Class Quiz 1: How to convert 60c to Fahrenheit:
- Class Quiz 2: Generalize your result in Quiz and find the general formula to convert from measure 1 to measure 2.
- Ans 1: (60/(100 - 0))*(212 - 32) + 32
  = 60 (180/100) + 32
  = 60 * 9/5 + 32 F
- Ans 2: ((measure_1 - Measure_1_Lower_Limit)/range_1)*range_2 + Measure_2_lower_limit.

A/D and D/A Mapping

A/D: ((_______ - _______)/(________))*(__________) + __________

((V - V_lower_limit)/(V_range))*(digital_range) (+ digital_lower_limit = 0)

D/A: ((________ - _______)/(________))*(__________) + __________

((digital_level)/digital_range)*analog_range + analog_lower_limit

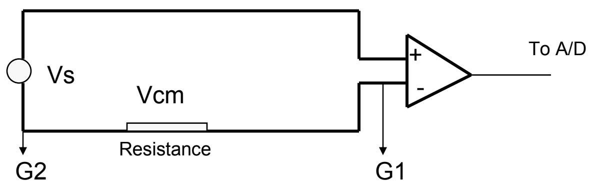
Example: analog voltages range is -1 to +2v and the digital_levels are from 0 to $2^n - 1 = 3$.
The digital range is 3 - 0 = 3.

A/D: V = 1: ((1 - -1)/3)*3 + 0 = 2. That is, 10B.

D/A: 10B (2/3)*3 + (-1) = 1

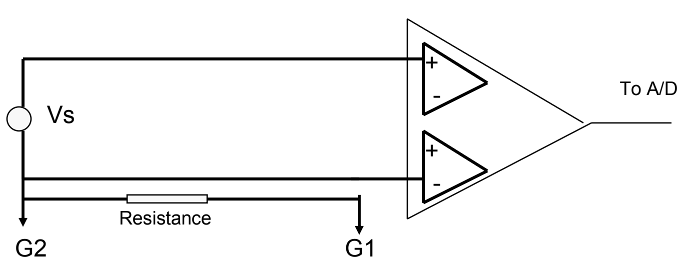
Single Ended Input

- External voltage can be connected with A/D cards via single ended inputs or differential inputs.
- Single ended input measures the difference between ground and the input signal. It is susceptible to EMI interference and voltage difference between grounds of the A/D card and the ground of the signal source. They are fine in a low noise lab environment.



Differential Inputs

- Differential connections are insensitive (e.g. up to 10 v) to ground differences and EMI.
- However, electronically differential connections requires twice input lines. For example, DAS 1600 may have 16 single ended A/D inputs or 8 differential A/D inputs.



Summary

- In this lecture you have learned
  - I/O instructions
  - Inline Assembly
  - The basics of a typical A/D - D/A card
- You are now in a position to do Lab 1.