EECS 388
Computer Systems and Assembly Language
System Addressing and Data

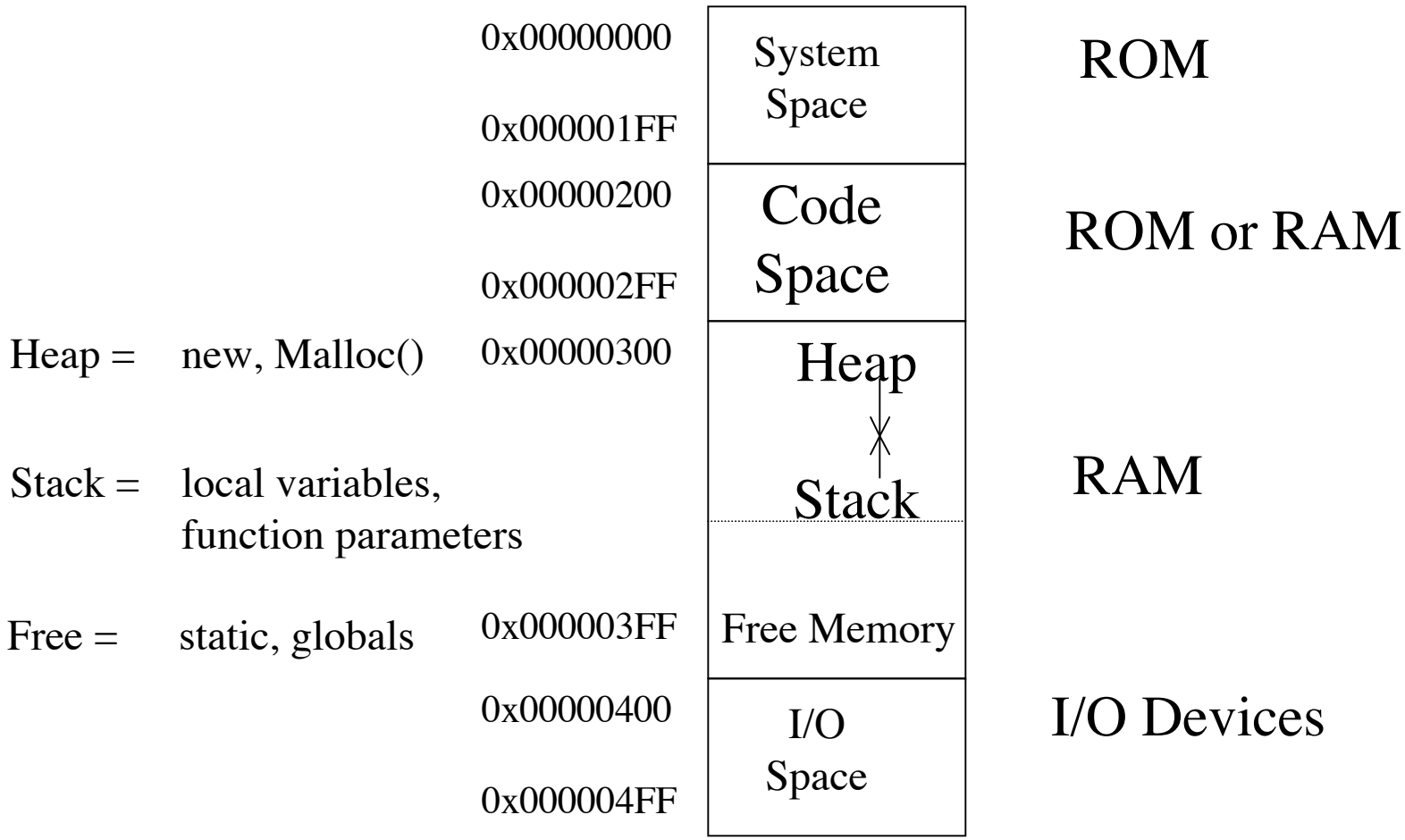
David Andrews

dandrews@ittc.ku.edu

All the worlds a stage…

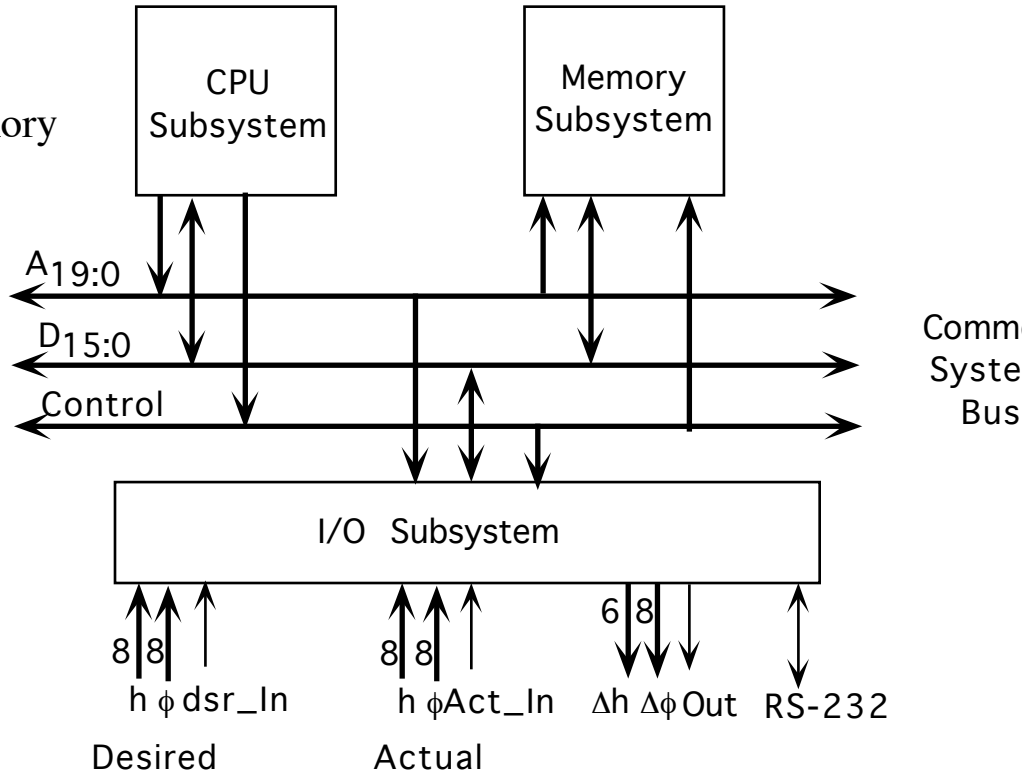
- We look at the “Computer” from the CPU’s perspective
  - The “system” is a contiguous set of addresses
    - We map specific devices into that address range
      - ROM
      - RAM
      - I/O Devices
    - Lets see….

Memory Address Breakdown



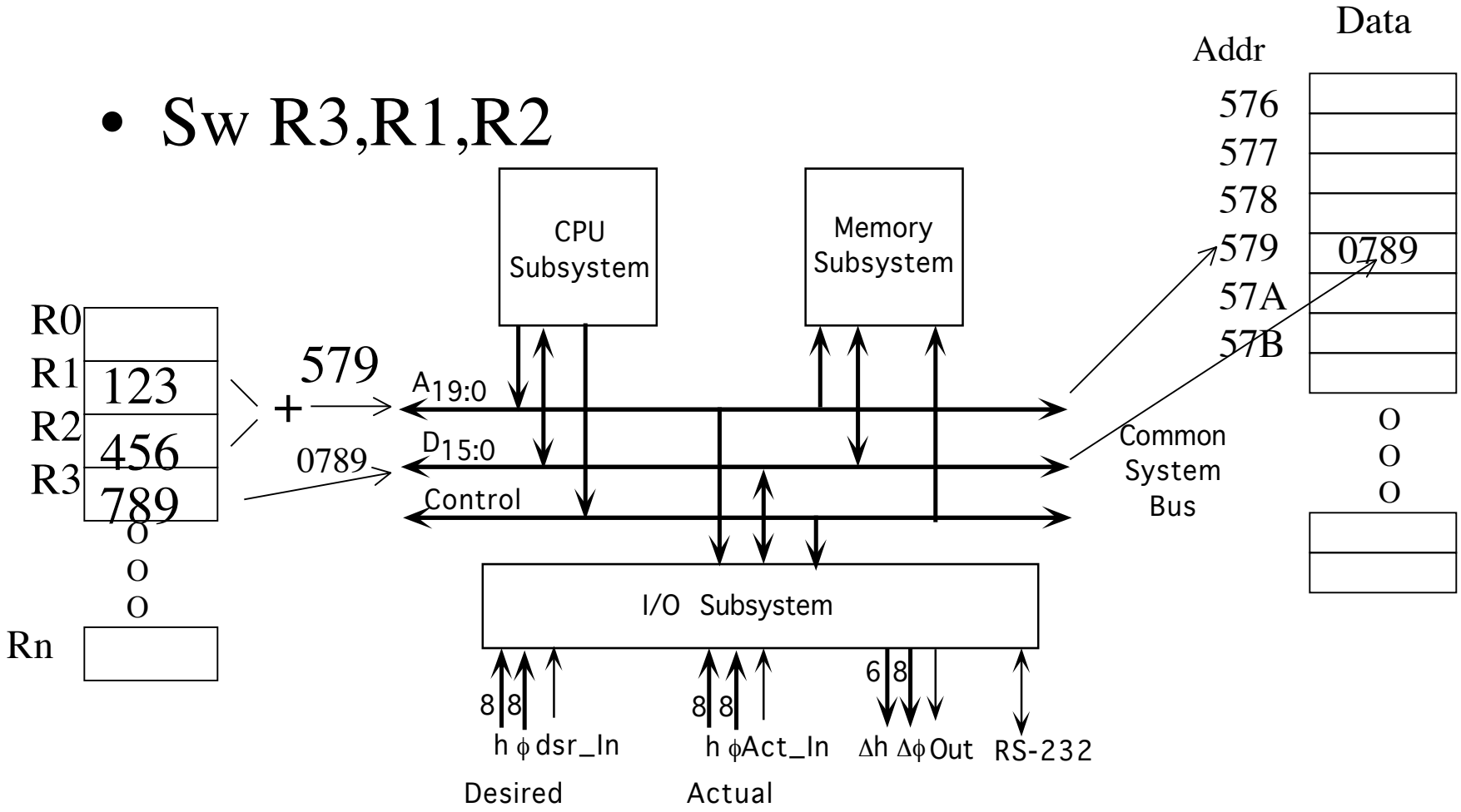
How CPU Addresses System

- 1st Dependency on "Common Bus"
  - Our system has separate bus lines for Address, Data
  - Don't worry about control lines shown
- Our CPU is a RISC load/store
  - ALU <--> Internal Registers <--> Memory
  - CPU -> System "store word"
    - Sw Ra, Rb, Rc
    - Ra -> Mem[Rb + Rc]
  - System -> CPU "load word"
    - Lw Ra, Rb, Rc
    - Ra <- Mem[Rb + Rc]



Example

- Sw R3,R1,R2



How Data Is Stored In Our System (Microblaze CPU)

- Motorola:= Big Endian
- Intel := Little Endian
- "Word" = 32 bits
  - int A;
- Half Word = 16 bits
  - Short int A;
- Byte = 8 bits
  - Char A;

Table 1-2: **Word Data Type**

Byte address	n	n+1	n+2	n+3
Byte label	0	1	2	3
Byte significance	MSByte			LSByte
Bit label	0			31
Bit significance	MSBit			LSBit

Table 1-3: **Half Word Data Type**

Byte address	n	n+1
Byte label	0	1
Byte significance	MSByte	LSByte
Bit label	0	15
Bit significance	MSBit	LSBit

Table 1-4: **Byte Data Type**

Byte address	n	
Bit label	0	7
Bit significance	MSBit	LSBit

Addressing I/O

- I/O Devices are accessed through their registers:
  - The number and meaning of registers depends on the specific device
  - Typically, there are two types of registers
    - Command/Status:
      - Command: Most devices are programmable. The CPU sets up the device by writing into the command register.. The CPU can continue to communicate and control the device during execution by writing into this register
      - Status: The CPU can read this register to see what the device is up to.
    - Data:
      - For I/O devices, this is were data is written in order to send, and read in order to receive.
  - Registers physically connect the I/O device to the bus.
  - They are simply address locations to the CPU
    - Load actually reads from a memory location
    - Store actually writes to a memory location

- We can communicate with devices directly from higher level languages such as C.
  - Lets look……