EECE 276
Embedded Systems

Review of memory types
I/O techniques
Hardware Concepts

Memory types

● Read-only memory
  » Non-modifiable (through usual instructions), non-volatile
  » Variants: EPROM, Flash, EEPROM

● Random-access memory
  » Modifiable, volatile
  » Variants: Static, dynamic (periodic refresh!)
I/O Techniques

HW Interface: Memory mapped I/O

Microprocessor Address Space

I/O Peripheral Interface Registers Mapped to Memory Locations

Implementation:

A0-A15 16 bit Address

A2-A15 High-order bits

A0-A1 Low-order bits

Peripheral Interface Device

CS*
Direct Memory Access (DMA)

Application: high-speed transfer of large blocks of data

CPU → DMA

DMA 

MEM

DMA 

I/O

Address bus

Data bus

BUSREQ → BUSACK

DMAREQ → DMAACK
I/O Techniques

Direct Memory Access (DMA) – “Cycle-stealing”

- I/O device can directly transfer data to/from memory – bypasses CPU!
- DMA controller
  - Gets transfer request from I/O
  - Obtains bus control from CPU
  - Runs bus cycle:
    - Read from I/O write to MEM or vice versa
  - Returns bus control to CPU
- When transfer finished: IT to CPU
I/O Techniques

Interrupt:

Hardware event that signals the end of a I/O operation and triggers a special subroutine call.

Process:

1. CPU initiates operation (transfer), then goes on doing something else.
2. When external operation finishes, it triggers an interrupt event.
3. If the interrupt event is acceptable to the CPU, it finishes current instruction, then invokes an Interrupt Service Routine (ISR) to handle the end of I/O operation.
4. ISR returns, interrupted program continues.
Interrupt-driven I/O:

**CPU**
- On IT Request:
  - Finish instruction
  - Stack machine state
  - Determine IT source
  - Call ISR
  - In SW (ISR)
    - Acknowledge IT
    - Finish I/O
    - RTI
  - In HW:
    - Return to interrupted code

**IREQ**

**MEM**

**I/O**

**IT Vector Table**

*ISR*
I/O Techniques

Software issues with interrupts:

- Disable IT-s when changing HW
- (Re)route IT vectors
- Enable IT-s when IRQ-s are expected
- In ISR:
  - Do NOT mess up the stack.
  - Ascertain IT source. If spurious, ignore.
  - If needed, acknowledge/clear IRQ.
  - Should not re-enable IT-s.
I/O Techniques

Interrupts:

- Allow overlapping processing with I/O
  » Asynchronous events!
- CPU can handle multiple I/O interfaces, rapidly
- Can be used to implement multi-tasking
- Almost all real-time systems rely on interrupts.