CS2261 Media Device Architecture

Announcements:

* Quiz next Monday (Interrupts, Sound (concepts), Timers)
* Project help moved to Wednesday, OO won’t be here

Timing/Timers

* Why are they needed?
  + For precise DMA transfers involving sound, timers are used
  + Clock/timer on screen
  + Creating an interesting background (lightning flashing every now and then)
  + Profiling studies (for optimization)

GBA Timers:

* Trigger an event after a duration
* There are 4 timers, each with a control register and data register
  + Each control register (REG\_TM0CNT, REG\_TM1CNT, REG\_TM2CNT, REG\_TM3CNT) has timer data associated with it (REG\_TM0D, REG\_TM1D, REG\_TM2D, REG\_TM3D) that is 16-bits
  + 4 different speeds/frequencies
    - When ticks should happen
    - Based on processor clock (more later)
    - Each timer can use any frequency
  + Control (8-bit):
    - Frequency, overflow, turn timers on and off, interrupts
    - Each bit does something
  + Data (16-bit):
    - Holds the time data in terms of ticks
    - # of elapsed ticks, range from 0 – 65535 (216 – 1)
    - Timers count up like a stopwatch, starting at all 0’s and ending at all 1’s before overflowing at 0
      * At overflow, timer usually resets to initial value
    - The max number of ticks per timer data is 216 – 1 = 65535 (16 bits)
      * And an extra one for overflow
    - 16 bits is the limit for ticks in a timer, but if we want to go over that amount, we can use **timer chaining**
      * Have one timer’s overflow trigger the second timer’s time to start
    - Only 4 timers, so the max you can achieve with this method if 64 bit timer

Frequency:

* Inner-workings of a processor are synchronized by an internal oscillator (or clock).
* What’s an oscillator?
  + Inner-workings of a processor are synchronized by an internal oscillator – circuitry that generates pulses as a sine or square wave
* A clock cycle is the time between 2 pulses
  + The more pulses per second indicates faster processing
  + Speed is usually represented by clock cycles per second -- in megahertz (MHz) or gigahertz (GHz) depending on magnitude.
    - A 1 MHz process performs 1,000,000 clock cycles per second.
    - A 5 GHz processor performs 5,000,000,000
* There are 4 frequencies (chart on slides)

Using Timers for Polling and Interrupts:

* Polling - start a timer, then check the the time (i.e. the value in the timer register after doing some stuff)
  + Find out how long a function takes to execute
    - check time before and after the function
  + Useful for development and optimization
* Interrupts - you are alerted by interrupt event whenever a certain amount of time has passed
  + Good for doing tasks at a precise time
  + or repeating a task at even intervals

Setting Up Timers:

* If you want a timer to go off after one tick, you should set it’s value to all 1’s (65535) and the transition to overflow (resetting) counts as one tick
* Otherwise, if you want a timer to go off after, say 60 ticks, set it to 65535 – 60
  + When the timer overflows, it’ll reset to its initial value of 65535 – 60
    - But you can only initialize the value of a disabled timer, so disable it, set the value and reenable it
* Frequency of the timer also matters