For Starters

• Basic information always fair game
  – Basic stuff about computers – ISA (what are instruction types, etc)
  – Basic stuff about number systems
    • Unsigned binary, Two’s complement, Excess codes, Floating Point
  – Basic stuff about MIPS CPU
    • What instructions do
    • Instruction encoding/decoding
    • Register(s) and their function(s)

And, Remember Interrupts

• Interrupts covered some on first test, BUT since they are so pervasive, will be used on second test as well
• Interrupts have information that is important in both the initialization and the steady state
• Interrupts break the Fetch-Decode-Execute cycle
Interrupt Initialization

• Create ISR with all smarts needed
• Create Vector Table with code, jumps
• Set up EBASE to point at table
• Set up functional units to allow interrupts and to set to known state
• Set up Interrupt Controller to allow interrupts
• Enable Interrupts (EI)

Steady State Interrupt

• ISR responds as appropriate (this different for different functional units)
• ISR polls to determine interruptor (when?)
• ISR handles work as determined by software design
• ISR clears (somehow) the bits that caused the interrupt activity
• Return to normal activity with ‘eret’
Functional Units

- First test really only dealt with UART
  - But, UART always fair game, since we use it all the time
- Other functional units include:
  - Timers
  - External interrupts/GPIO pins
  - PWM system

For the Functional Units

- Know what the basic function is and how it is accomplished
- Know how to use with/without interrupt
- Know how to enable interrupt, reset interrupt flags
- Know how to access (limitations, as appropriate)
- Know how to get to pins with Programmed I/O
Instruction Activities

- Normal stuff (RISC vs CISC) – what can instruction do
- Instructions for data movement/program control
  - Subroutine linkage – how to get back
  - Subroutine parameter passing
- How to...
  - Stacks and other data structures
  - Control the chip, system

Software Techniques

- How to do data structures with assembly language (array, list, … )
- How to utilize loops
- How to test for conditions