ECE 271 – Microcomputer Architecture and Applications Lecture 26

Vince Weaver

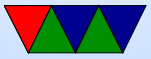
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

26 Aril 2022

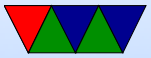
Announcements

- Don't forget the Final!
  Thursday, May 5th, 8am, this room (Barrows 119)
- Turn in your labs!



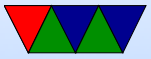
Hand Back and go Over Midterm #2

- Average was a 92



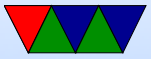
Final Exam Review

- You're allowed on page of notes (8.5" x 11")
- You're allowed a calculator



Final Review – Midterm #1 Topics

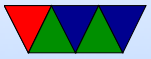
- Know basic C
  - Setting/clearing bits
  - Knowing what volatile is and why you use it
  - Knowing that local variables go on the stack, difference from global variables.
- Assembly language
  - I will provide a table of THUMB2 instructions so no need to memorize.
  - Things like, what does this code do? Or, add



comments to this code, or, what is wrong with this code.

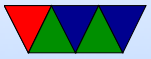
- Know basic things, like what a loop looks like, and how to load into registers.
- I *won't* ask detailed flag question, but remember how CMP works, and how to make ALU instructions set the flags (ADD vs ADDS).
- Also be able to work through a simple loop that uses CMP followed by BEQ or BNE or similar

- ABI, know why we have one.
  - ARGS go in r0-r3



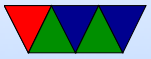
- return value in r0
- Some register are callee vs caller
- Return value in LR, what happens in leaf vs other function

- Lab topics
  - GPIO, LCD, Scanning, Stepper
  - Most important is GPIO as we used those the most, no low-level details about the others
  - No need to memorize all of the MODER register fields, etc.
  - Questions may be similar to those on pre/post-lab



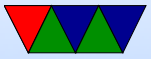
Final Review – Midterm #2 Topics

- Interrupts
  - Vectored interrupts – there is a table low in memory (the interrupt vector) which has a lookup table of pointers. When an interrupt happens, the CPU looks up the address from the appropriate pointer and jumps to it
  - Interrupt handlers on Cortex-M are just normal functions (the CPU does "stacking", saving some registers on the stack automatically for you. A magic



value is put in the Link register so the CPU knows you are returning from a handler).

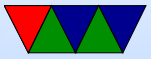
- Once an interrupt happens, you enter the handler. Often you will need to ACK (acknowledge) the interrupt so the CPU knows you are done handling it. How this happens can vary with what hardware you use (it's often clearing a bit, but somehow it happens automatically).
- Enabling an interrupt is multiple steps. You have to enable it in the device (For example, in the timer registers). Then you have to tell the NVIC (interrupt



controller) to enable it. Then you have to enable it on the CPU globally with `__asm("CPIE i")`. Finally you have to enable whatever hardware will be triggering the interrupt.

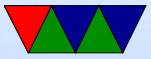
  - Some devices (such as TIM4) can have multiple causes that trigger the same interrupt (i.e., overflow and capture). You can figure out which one was the cause by checking certain bits to see the source of the interrupt.

- Timers
  - Lots of registers to set.



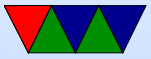
- Set what clock source to use system wide.
- You can then use the prescalar to divide the frequency down.
- You can set to count up (from 0 to the ARR register value), count down (from ARR to 0) or center count (count up and down).

- PWM
  - pulse width modulation: you set a CCR value that when the counter gets bigger it triggers an output to go high. This way you can generate a regular pulse with an overall frequency based on ARR and a duty

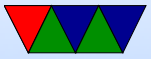


cycle based on CCR.

- Input Capture
  - You can use the Timer to measure the length of signals.
  - When the incoming signal has a transition (high to low or low to high) the timer will grab the current timer count and store it in a register. (optionally also an interrupt can be triggered).
  - By saving the previous value, and subtracting from the current value, you can calculate how many clock ticks have happened
  - If your signal is wider than the maximum value of the

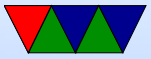


counter, an overflow can happen. In order to measure signals that long you can catch the overflows and count them, and then add that time in to the measured time.

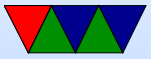


More Recent Topics

- ADC – analog to digital conversion
  - Measure analog signal, convert to digital values
  - Can read analog sensors
- DAC – digital to analog conversion
  - Convert digital values to analog signal
  - Can be used to generate wave forms, music
- Floating Point on Cortex-M4
  - Be aware of fixed point vs floating point tradeoffs
  - Do a simple floating point to decimal conversion (I'd



give you the formulas)
  - Know the FPU is a separate unit. Cortex-M4 only has single precision (32-bit) support. Separate register set, completely separate instructions starting with V
- DMA
  - Special unit that can do peripheral to memory copies without CPU interaction
  - Can run in background, send interrupt when done (or nearing done)
  - Frees up the CPU for processing
- Recent material detailing busses on the STM board



- Be aware of it, but probably no detailed questions