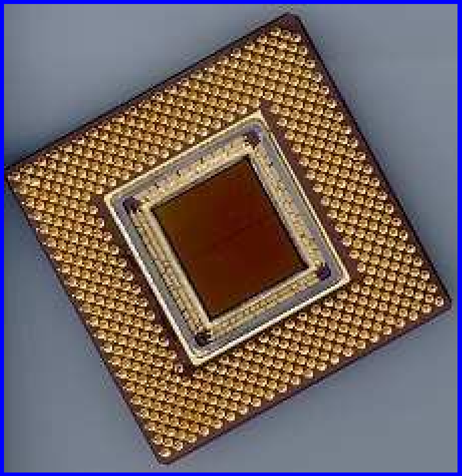
© 2014 by D. G. Meyer



ECE 477 Digital Systems Senior Design Project

Module 10

Now What??? (Systematic Debugging)

Outline

- "Avoid Really Stupid Tricks (RST) when debugging"
- "My power supply doesn't work. What do I do now???"
- "My microcontroller doesn't work. What do I do now???"
- "My microcontroller kinda works, but 'strange things' are happening. What do I do now???

“Avoid Really Stupid Tricks (RST)” - 1

- Do NOT solder parts, attach wires, connect probes, etc. while your board is powered up
  - Temporary short circuits can totally fry numerous components on your board
  - Remember that the soldering iron tip is GROUNDED, and that random points tied to ground on your board while it is in operation could be a very bad thing!

“Avoid Really Stupid Tricks (RST)” - 2

- Do NOT attempt to “probe” the pads of surface mount parts
  - Few have been known to do this successfully
  - Shorts between/among microcontroller pins, regardless of how temporary, can be disastrous
  - This is why you put headers on your board!

“Avoid Really Stupid Tricks (RST)” - 3

- Do NOT connect port pins directly to power supply rails to obtain a “1” or “0” when debugging (or, *in general*)
  - Pin might be programmed as an output, at which point its pad driver will be instant toast
  - If pin programmed as an input, might be over-biased and destroy buffer
  - ALWAYS use a resistor (10K is a good value) as a pull-up or pull-down – ALWAYS
  - To test ADC (analog) inputs, ALWAYS use a potentiometer (10K is a good value) – ALWAYS

“Avoid Really Stupid Tricks (RST)” - 4

- Do NOT attempt to power different parts of your circuit with different (external) power supplies
  - Think about what will happen to your board/parts if two essentially infinite current sources are trying to drive (*even slightly*) different voltages on the same net/trace…
  - Then, look for “burn marks” on ECE 362 students who tried this on their Mini-Project

“My power supply doesn’t work” - 1

- Quantify symptoms
  - no voltage output whatsoever
  - voltages over/under specifications
  - voltage drops significantly when connected to the rest of your circuit
  - excessive heat generated by components
  - excessive ripple/noise on regulated output

“My power supply doesn’t work” - 2

- Consider possible causes of each symptom
  - no voltage output whatsoever – check input voltage; if that’s OK, check output impedance; if that’s OK, replace regulator IC
  - voltages over/under specifications – check to make sure regulator GND pin is hooked up (also look for cold solder joint); otherwise, replace regulator IC
  - voltage drops significantly when connected to the rest of your circuit – *may* be bad regulator, but also could be load impedance too low (partial short/malfunction)
  - excessive heat generated by components – unregulated input voltage may be too high, load may be partially shorted, or insufficient heat sink used
  - excessive ripple/noise on regulated output – check size of filter caps and/or amount of input ripple

“My microcontroller doesn’t work” - 1

- Perform basic checks
  - what’s the first basic thing you should check?

    look at power/ground pins under microscope
  - what’s the second basic thing you should check?

    oscillator circuit *(but be careful, since connecting scope probes to oscillator circuit pins can detune it enough to disable it)*
  - what’s the third basic thing you should check?

    reset circuit – is reset being held low “long enough” (and not stuck low)?
  - what’s the fourth basic thing you should check?

    silicon errata sheet for your microcontroller

“My microcontroller doesn’t work” - 2

- More advanced checks
  - “heartbeat” program – just toggle a port pin and look at on scope (make sure heartbeat works on your Eval Board)
  - utility routines to exercise ports and on-chip peripherals
    - Digital I/O – distinctive pattern (Johnson counter)
    - ADC – connect potentiometer to inputs / vary
    - PWM – increment duty cycle in loop
    - Timers – generate periodic interrupts
    - SPI – transmit/receive distinctive patterns
    - SCI – transmit/receive distinctive patterns

“My microcontroller kinda works” - 1

- Possible “random” behaviors
  - digital input values read from a port are “random” – check pull-up (internal/external), most likely inputs are floating (also check for cold solder joints)
  - digital input values read from a port have a consistent (wrong) pattern – check programming of port pins – make sure they are configured as inputs – input may be damaged due to ESD/over-voltage
  - the same value is read from adjacent port pins – check for solder bridges, cold solder joints, or floating adjacent pins

“My microcontroller kinda works” - 2

- Possible “random” behaviors…
  - digital values sent to an output port don’t appear on the pins (port pins are not driven) – check to make sure port pins are programmed to be outputs, and that the “drive” register (if present) is set correctly (some pins can be configured to be open-drain)
  - some output pins are always driven high or always driven low – this is most likely due to a “stuck at” fault resulting from a failure in the pad driver circuit (if there are spare port pins available, fly wire around the problem: otherwise, it’s time to replace the microcontroller)
  - the same value is output on adjacent port pins – check for solder bridges (and possible damage to μC)

“My microcontroller kinda works” - 3

- Possible “random” behaviors…
  - analog input values read from the ADC are always zero (or, always the same value) – check reference voltage, ADC programming/device driver (make sure you are waiting for the “conversion complete” flag to be set)
  - analog input values read from the ADC are “random” (or, are inconsistent) – check ADC reference voltages and input pin solder joints (cold joint can act like a capacitor); also, check driver routine to make sure it is handshaking on the “conversion complete” flag
  - the lower two bits (or so) of values read from the ADC are “random” – this is “normal” (typically all the lower two bits are good for is rounding the result!)

“My microcontroller kinda works” - 4

- Possible “random” behaviors…
  - the SCI/SPI is not receiving or transmitting data – most likely a configuration/driver problem (look at Tx line on logic analyzer; when Tx works, loop back to Rx line and check to make sure the same characters your driver is transmitting are being received)
  - the SCI is “alive” (can see waveforms on oscilloscope), but board will not communicate with a PC “com” port or a serial LCD – check baud rate and character frame format of sending/receiving end; check to make sure RS232 level translators generating correct polarity and voltage swing (nominally ± 9 V); check to make sure side “A” Tx connected to side “B” Rx, and vice-versa

“My microcontroller kinda works” - 5

- Possible “random” behaviors…
  - processor “crashes” when board is moved or “flexed” – bad (“cold”) solder joint somewhere (retouch all power/ground solder joints)
  - processor runs for a few milliseconds and resets…processor runs for a few milliseconds and resets… – make sure WDT (watchdog timer) is disabled; also, check reset circuit

“My microcontroller kinda works” - 6

- Possible “random” behaviors…
  - applications run on the processor for a few seconds, and then “crash” – may have random interrupts that are not implemented or handled correctly, stack may be overflowing (“creep” due to software error)
  - microcontroller gets too warm for comfort – pins programmed as outputs are “fighting”, input signals not at supply rail potential (weak 1’s), or may be drawing too much current from port pin (source/sink)