Digital Logic Troubleshooting

Troubleshooting

• Basic Equipment
  – Circuit diagram
  – Data book (for IC pin outs)
  – Logic probe
  – Voltmeter
  – Oscilloscope

• Advanced
  – Logic analyzer
Basic ideas

- Troubleshooting is systemic
- Don’t start troubleshooting without a circuit diagram
- Troubleshooting requires **time** and **patience**. Don’t give up if the fault can’t be found after a short time
- Most faults are caused by incorrect wiring. Wiring a circuit neatly will minimize the time spent debugging.
- When constructing a circuit on a breadboard put the components of an IC near the particular IC. Thus you know how the circuit was built.
- Mark each wire on the schematic. This way you won’t miss out any wire.

**Build a circuit one module at a time. Try to test each module independently. A good design is based around this concept**

Troubleshooting - Breadboard

- Check for shorted circuited power supplies
- Loose wires
- Components incorrectly inserted
- Spaghetti wiring!
- Bent leads
Troubleshooting steps

• Check that power is supplied to each IC at the correct pins!
• If no power is present, check for continuity along the +5V and GND respectively
• Short circuit conditions
  – If the voltage across the supply is less than 4.5V, consider a possibility of a short circuit
  – If any LED’s are not as bright as they are supposed to be consider a short circuit
  – If the regulator is hot there is a short circuit, switch off immediately

Troubleshooting – Combinational Logic

• Use a logic probe to check the logic level
• For combinational logic circuits
  – Check the expected and actual logic values are the same
• For switched inputs check that the switch is wired correctly (no floating inputs)
• Check if the output changes when the logic state of the switch changes.
• If the output is stuck at a particular logic level try changing the IC
  – In the case of combinational logic functions check if the Enable/Chip Enable inputs are set to the correct logic value for the IC to operate properly

Always consult the manufacturer’s data sheet for the correct pin out and operating modes
Troubleshooting – Combinational Logic

- Use a logic probe to check the logic level
- If the logic level is undefined, check with a voltmeter.
- If the output voltage is in the undefined region, replace the IC
- If different logic families are being used, make sure that they are correctly interfaced. Signals that are correct on a logic family, may be too short on another logic family to trigger correctly.

*Make sure there is a series driving resistor (150Ω – 820Ω) when driving an LED*

Troubleshooting-Clock signals

- Use the logic probe to check that the clock signal is applied to all clock inputs.
- Slow down the clock frequency if necessary
- Advanced Check – Use an oscilloscope to check the clock waveform.
- If the clock signal amplitude is less than the supply voltage than the signal is loaded
  - Resistance to ground
  - Burnt IC

*Use a 100nF decoupling capacitor between IC and the ground to reduce noise on the clock signal*
Troubleshooting – Sequential logic

• Sequential logic functions are composed of two sections
  – Asynchronous inputs
  – Synchronous inputs
• Make sure that the IC is configured in the correct mode.
• Consult the datasheet to get a list of the operating modes
• If the IC resets unexpectedly
  – There may be an incorrectly wired combinational logic gate
  – Noise on the reset line may be the cause of a reset. Decouple reset line by a 100nF capacitor to ground
• Use a function table to create a table of expected values and see where and how the output differs.

Remember that asynchronous inputs have priority over synchronous inputs, so check them first

© Joseph A. Zammit 2007

Specialized functions - Counter

• Counters are supposed to count sequentially!
• Counters that display the wrong value may be wired incorrectly
• If a counter resets at the wrong binary number, check the reset logic
• Counter starts at any value other than 0 on startup, check reset or parallel load inputs
• Make sure that the counter resets on startup

The circuit may be frozen by applying a single pulse clock instead of the normal clock

© Joseph A. Zammit 2007
Specialized functions – Shift register

- The easiest and most efficient way to see that a shift register is working as it supposed to be is to apply a test pattern
- A test pattern is a known input that will produce a predictable output
- Other than that, a suitable test pattern will allow a fault to debugged found more easily
- Examples of test patterns (8 bits)
  1. 10000001 (Hex 0x81)
  2. 10010110 (Hex 0x96)
  3. 01010101 (Hex 0x55)
  4. 10101010 (Hex 0xAA)
- Patterns 1 and 2 is a unique pattern, Patterns 3 and 4 apply an alternating pattern.

*Sometimes it helps by attaching an LED with a high value (1kΩ) series resistor to monitor a particular output*