CE1911

Spring 2016

Week 9-10 Lab

Overview

This lab replaces the special purpose algebra processor lab for weeks 9 and 10. In electrical and computer engineering, state machines are often used to create waveforms to communicate information between devices. The state machine is designed to create a waveform that meets the timing specifications of the communication partner device.

Your DEO-Nano-SoC board has an analog to digital converter (ADC). The ADC senses a voltage on one of eight input pins, and converts the analog voltage to a 12-bit number. This 12-bit number is sent back to the FPGA using serial communication. The FPGA sends configuration information to the ADC via serial communication as well.

For this lab, you will use the ADC and your LT24 display to create a digital voltmeter. Your circuit will communicate to the ADC that you want it to convert an analog voltage on pin 7 to a 12-bit value. Your circuit will then sample the serial communication line to gather all 12 bits. Your circuit will use a read-only memory (which we will create in class together) to convert the 12 bit binary number to decimal and then display the voltage on LT24 seven segment displays SEG3 through SEG0.

Block Diagram

The block diagram on the following page lists the components that you will need to make for this project.

- State Machine

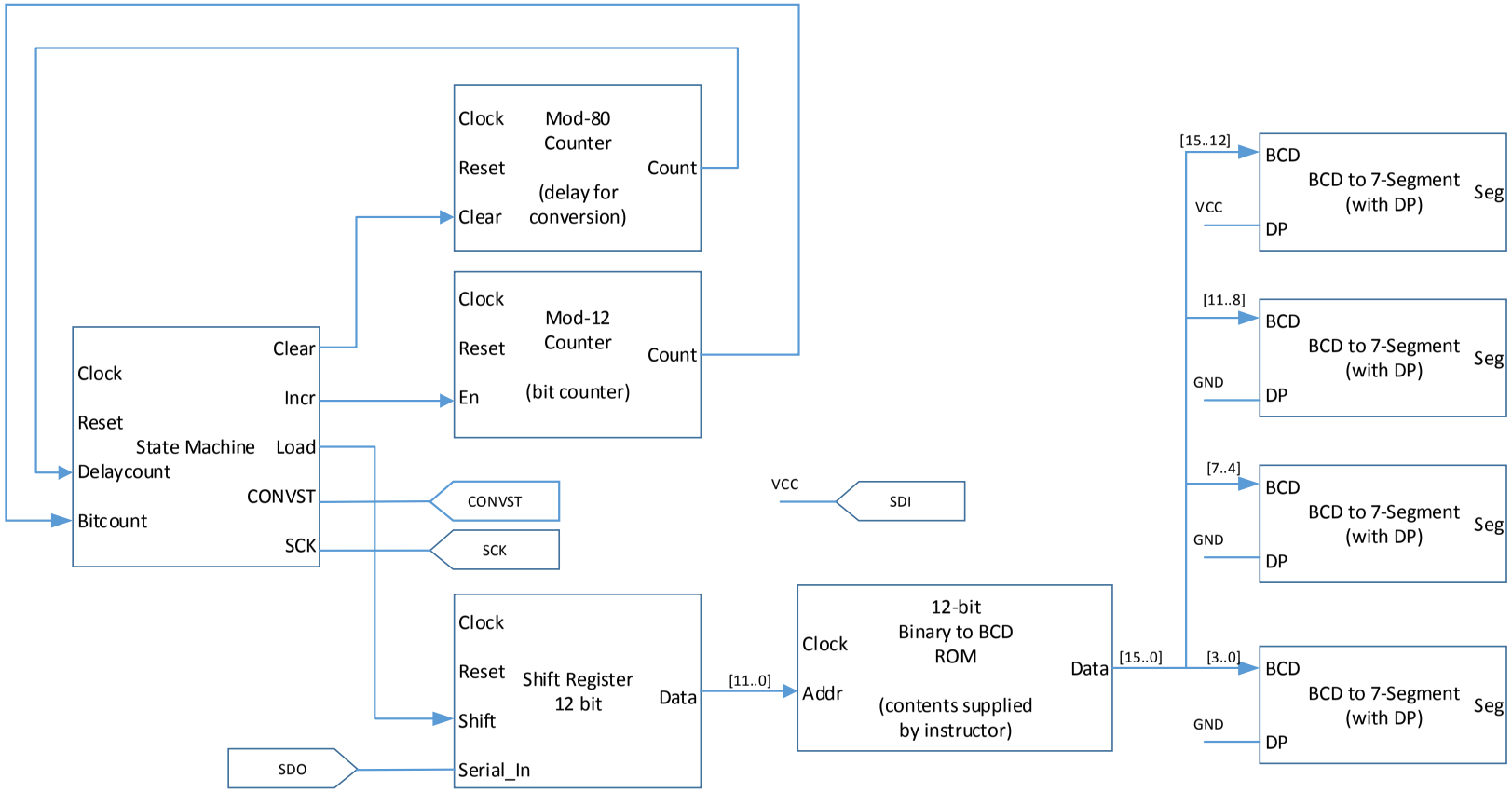
  You will be creating a state machine that acts as a controller, running the various other blocks in the system that will create the waveform and assemble/convert/display the 12 bit voltage value. Your state machine will read and clear a mod-80 counter, which is used to time the waveform that gives the ADC the signal to start conversion. (Your instructor will discuss timing in greater detail in class.) Your state machine will also use a mod-12 counter, to count the bits coming in. Your state machine will load each bit of the voltage value into a shift register. All of these registers need control signals to increment and clear them for example. It is the job of your state machine to provide these signals, and also change the voltage on the ADC pins to create the communication waveform.

- Mod-80 Counter

  The ADC takes a while to complete its conversion. A mod-80 counter will help time our conversion requests. Specific timing requirements will be discussed in class.

- Mod-12 Counter

  We need to read the data line coming from the ADC 12 times, since it only sends one bit at a time. The mod-12 counter will tell us when we have assembled the entire 12-bit voltage value.


Clock
Mod-80
Counter
Reset
Count
(delay for
conversion)
Clear
Clock
Mod-12
Counter
Reset
Count
(bit counter)
En
Clear
Clock
Incr
Reset
State Machine
Load
Delaycount
CONVST
Bitcount
SCK
CONVST
SCK
VCC
SDI
Clock
Reset
Shift Register
12 bit
Data
[11..0]
Shift
SDO
Serial_In
12-bit
Binary to BCD
ROM
Clock
Data
[15..0]
Addr
(contents supplied
by instructor)
[15..12]
BCD
BCD to 7-Segment
(with DP)
Seg
VCC
DP
[11..8]
BCD
BCD to 7-Segment
(with DP)
Seg
GND
DP
[7..4]
BCD
BCD to 7-Segment
(with DP)
Seg
GND
DP
[3..0]
BCD
BCD to 7-Segment
(with DP)
Seg
GND
DP

- Shift Register

  This component saves one bit of a 12-bit number, and then shifts the bits over to make room for the next incoming bit. When it has loaded and shifted 12 times, it has assembled the number.

- Binary to BCD ROM

  A read-only memory (ROM) takes in a location number (called an address) and looks up the value at that location. The ROM will be used to take a 12-bit number and convert it into the constituent decimal digits (thousands, hundreds, tens and ones). These four decimal digits are each represented by a four bit number, 0000 through 1001. We would call these digits binary-coded-decimal digits, or BCD for short. We will create the ROM together in class.

- BCD to seven segment (with decimal point) converter

  You have already used this component in previous labs. The component takes in a 4 bit number, and a single bit representing the decimal point, and outputs an 8 bit pattern of segments that can be fed to the seven segment display on the LT24.

ADC Pins

Your circuit will access the ADC through the following pins, also shown on the block diagram:

- CONVST

  Stands for "conversion start". Your circuit will put out a pulse on this pin to tell the ADC to start a conversion. Specific timing will be discussed in class.

- SCK

  Stands for "serial clock". This is not the same clock that will be used with 'event in your VHDL! This is a clock-like waveform that your circuit will create so that the ADC sends back each of the 12 bits in a predictable manner.

- SDI

  Stands for "serial data in". This is a voltage that goes into the ADC to configure it (tell it which pin to use, whether to use signed or unsigned numbers, etc.) By keeping this pin high all the time, we can tell the ADC to use a default configuration with the voltage on pin 7 and unsigned numbers for the voltage values.

- SDO

  Stands for "serial data out". This voltage comes out of the ADC and into our circuit. The ADC brings this line up and down at specific times given by SCK, to tell us each individual bit of the 12 bit voltage value.

ADC Datasheet and Timing Diagram

The ADC has its own datasheet that specifies exactly what we should send it in order to successfully communicate. This datasheet will be available on the course website, and we will interpret it in class. It will be our main resource for crafting our state machine to make the correct communication waveform.

Due Date:

This lab should be demonstrated in the Week 10 lab period, and must be demonstrated before our final exam. If your system is not working, make an appointment with me before the final exam, so that I can take a look at the system and give you a grade for what you have completed. Demonstration:

The Analog Discovery kit will output an analog voltage between -5V and 5V on the yellow solid wire. To choose a voltage, launch the AWG (waveform generator) from the Waveforms software. The ADC on your board will be able to read voltages between 0V and 4.096 V, so test your circuit with a voltage in that range. Connect the yellow solid wire to input 7 (pin 9) on the ADC, and connect a black wire to ground (pin 10) on the ADC.

The voltage displayed on your LT24 should be close to the voltage shown on the AWG screen. I find that the output of the Discovery reads about 0.1 V lower than the Waveforms setting would indicate. If the error is greater than 10%, then there is almost certainly a problem with either your circuit or the Analog Discovery unit. Consult your instructor for help on using the Analog Discovery unit.

Deliverables:

- RTL viewer high-level diagram—should show something similar to the block diagram in this packet.
- RTL viewer showing the inside of each of the individual components, annotated to confirm that a state machine was created. Also annotate the RTL diagram to confirm that it is free of latches.
- Hand-drawn state diagram for your state machine, using the state names and signal names from your VHDL code. You may use Visio or something similar to make the diagram if you wish. The state machine viewer diagram from Quartus is **not** acceptable on its own. However, I will accept a printout of the Quartus state machine viewer diagram if you write your state names and transition conditions by hand on the diagram.
- VHDL code, printed from Quartus or pasted into a Word document. Do not use screenshots for your VHDL code.