EE 121
Digital Design Laboratory

October 29, 2002
Handout #19

Laboratory Assignment #5

Sequential Circuit: Clock-Calendar

Prelab due date: Tuesday, November 5, at 9:30am

Design a clock-calendar circuit using Xilinx Foundation software. Enter a hierarchical logic design of the circuit using the Xilinx schematic editor. Test the circuit using the Xilinx logic simulator. In laboratory section, download the circuit into an XC2S100 FPGA and demonstrate its operation using a VGA monitor to display the date and time values.

The block diagram of the clock-calendar is shown in Figure 1.

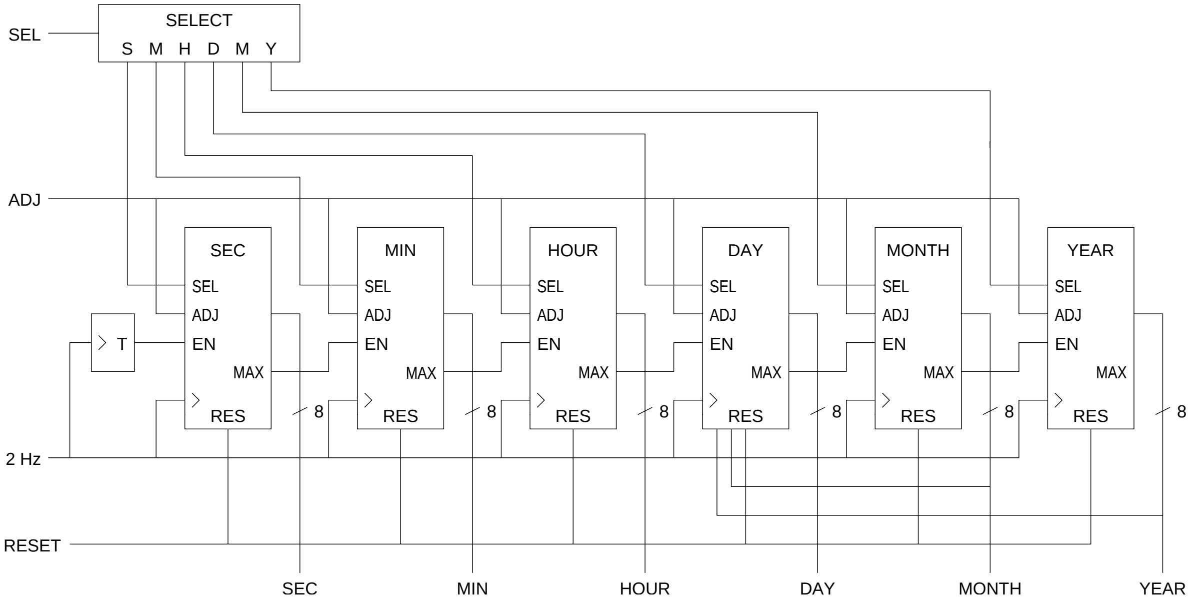

Figure 1: Clock-calendar block diagram

The outputs of the clock-calendar are the binary-coded decimal digits that represent the date and time in 24-hour ("military") format. Each digit is encoded using four bits, so eight bits are used for each of seconds, minutes, hours, day, month, and year. For example, the date/time 7:34:25pm on March 17, 20021, is represented by

02/03/17 19:58:25 = 0000 0010 0000 0011 0001 0111 0001 1001 0101 1000 0010 0101

Since the Y2K problem is in the past, this clock-calendar stores the year in only two digits.

In normal operation, the seconds counter increments each second. Every 60 seconds, the seconds counter reaches its maximum value of 59 and enables the minutes counter, which increments on the next rising clock edge. Similarly, every 60 minutes, the minutes counter reaches 59 and enables the hours counter to increment. All six counters are clocked by the same 2 Hz signal, so this clock-calendar is an example of a clocked synchronous state machine. Note that the toggle flip-flop that enables the seconds counter every other clock tick, that is, once per second; the output of the T flip-flop is *not* used as a clock.

The basic building block of the clock-calendar is the decade (modulo 10) counter. The Xilinx library part for this counter is the CD4CLE, which is a 4-bit loadable cascadable BCD counter with clock enable. (Other decimal counters in the library could also be used.) The logic block of the CD4CLE is shown in Figure 2. This counter is similar to the 74x163 binary counter except that its maximum value is $9_{10} = 1001_2$ rather than $15_{10} = 1111_2$. Two decade counters can be cascaded to create a modulo 100 year counter. With a small amount of additional logic, two decade counters can be combined to count with other periods, such as 60 (for seconds and minutes) and 24 (for hours).

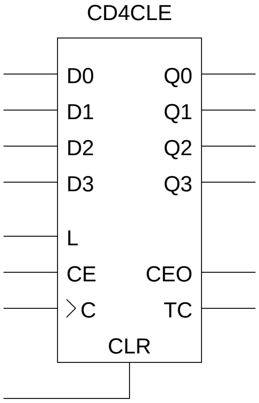

Figure 2: CD4CLE decade counter logic symbol

The day counter is much more complicated than the others because the number of days in a month depends on both the month and the year. Using the inputs from the month and year counters, the day counter wraps around to 1 after its state reaches the number of days in the current month. The day counter logic circuit should store a table of days per month. The number of days per month ranges from 28 to 31. In a leap year, February (month 2) has 29 days instead of the usual 28 days. (For the purposes of this laboratory assignment, a leap year is any year divisible by 4; that is another reason that this circuit fails in the year 2100.)

The time and date stored in the clock-calendar can be changed using the inputs SEL and ADJ. At any time, the SEL input of at most one counter is true. The SEL input is used to change which counter is selected; each pulse of SEL chooses a different counter, in the order S, M, H, D, M, Y, none. When a counter is selected, the ADJ signal enables incrementing on every clock cycle, and that counter advances at a rate of 2 Hz. When ADJ is false, the counter operates normally and increments when enabled by the next low order counter.

A simple circuit for the SELECT component is shown in Figure 3 on page 3. This circuit performs no processing on the SEL input—no debouncing or conversion of long input pulses to single clock cycle pulses—since the 2 Hz system clock is slow enough to be used directly.

The third input signal is RESET. When RESET is active, the counters are forced to the initial state 00/01/01 00:00:00. The nonzero initial values can be loaded into the counters by wiring the direct load inputs with the required values.

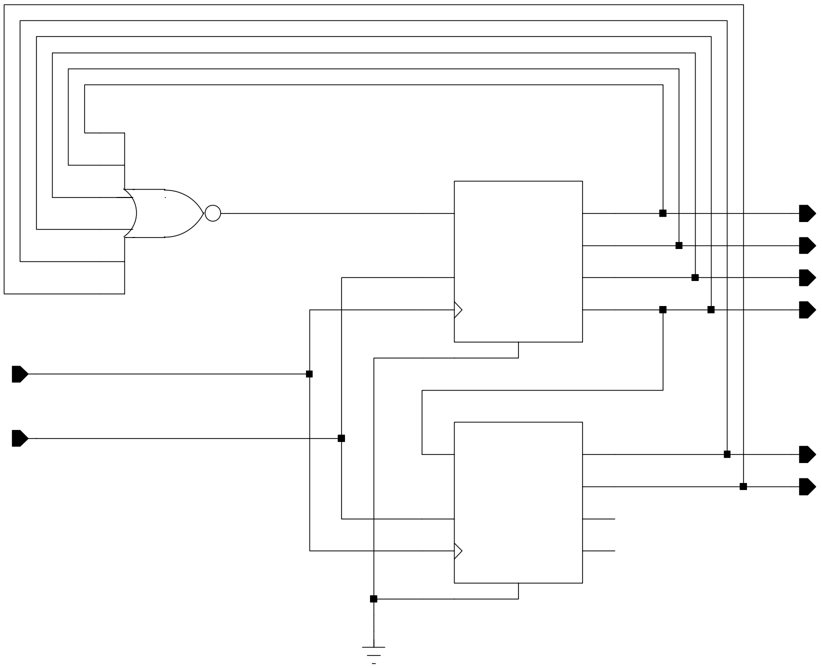

Figure 3: SELECT schematic

Laboratory Requirements

Design Xilinx macros for the six counters in the clock-calendar circuit. The counters, (except the day counter), should be built using two copies of the Xilinx CD4CLE macro and additional combinational logic. Create the clock-calendar circuit using the six counters together with the macro for SELECT. The 2 Hz clock signal that clocks all the counters must be obtained by dividing the 50 MHz master clock for the XC2S100.

The clock will be displayed on a VGA monitor through the VGA output of the XSA board. A top-level schematic will be provided so that the outputs of the counters in your design can be directly tied to a VGA-display module provided in the schematic (i.e., no knowledge of how the VGA module works is necessary).

The value of the counter currently selected by SELECT will also be shown on the two seven-segment displays on the XStend board. These six 8-bit values should be fed into a 6-to-1 multiplexer whose output is decoded by a seven-segment decoder (to be provided) that is connected to the seven-segment displays. The output of the multiplexer should be FF (all ones) when no counter is selected; this code will blank both seven-segment displays.

Prelab Requirements

1. A complete Foundation project with schematics for all logic blocks in your design (submitted electronically)

2. A test script demonstrating your design works (submitted electronically)
3. A report file that specifies the number of Slices/IOBs of your design (submitted electronically)
4. `README` file (submitted electronically)

The submission procedures are the same as for laboratory assignment #4.

For this laboratory assignment, partners should work together.

Laboratory Exercise

During lab time, you will get a chance to compile your design, download it into a Xilinx chip, and test it with real signals. There is no lab write-up for this lab. However, you do need to demo your circuit to your TA, using any test values your TA desires.