Lab # 7

An Alarm Clock – Phase 1

Objectives

- More practice on macros.
- First lab to build an alarm clock, which displays minutes and hours.

Overview

This lab is the first of a series of four labs, in which you will eventually build a digital alarm clock. The clock should display minutes and hours in the seven-segment displays available on the Digilab board. Furthermore, the clock is to have two modes of operation, controlled by an input *MODE* signal.

MODE-0 is the normal mode where the clock displays the current time. In this mode, time can be adjusted by individually adjusting either the minutes or the hours. Two buttons are needed for this purpose. One button is for advancing-minutes (AM-button) and the other for advancing-hours (AH-button).

MODE-1 is the alarm mode, where the clock displays the time at which the alarm is set. In this mode, the AM & AH buttons are used to adjust alarm time. For example, pressing button AH in this mode (MODE=1), advances the hour entry of the alarm time and the clock display shows the alarm time as the hours are advanced.

Table 7.1 summarizes the operation of the alarm clock.

MODE	Alarm Clock Function
0	Displays current time. Use buttons *AH* & *AM* to advance the hours & minutes of current time.
1	Displays alarm time. Use buttons *AH* & *AM* to advance the hours & minutes of alarm time.

Table 7.1: Operation of the alarm clock.

We need to run our clock at 1 Hz during normal operation. However, when setting the time, we would like to run our clock at twice the speed, i.e. 2 Hz.

Design Specifications

The digital alarm clock to be designed requires use of five counters

For Normal Clock Operation

- 2 Mod-60 counters for counting seconds and minutes,
- 1 Mod-24 counter for counting hours.

For Alarm Operation

- 1 Mod-60 counter for storing minutes for alarm.
- 1 Mod-24 counter for storing hours for alarm.

Modulo-24 Counter

In previous lab, you designed modulo-10, modulo-6 and eventually a modulo-60 counter. Similar to the counters you designed previously, a modulo-24 counter counts from 0 to 23 and then roll backs to 0. In this lab we require modulo-24 counter for keeping track of hours in our digital-clock.

The modulo-24 counter is very similar to a modulo-60. The difference with modulo-60 counter is in the way the asynchronous clear signal is generated. In mod-24 counter,

- We need to reset the mod-24 counter to all zeros if we reach the value of 24, i.e. 24 hours.
- Apart from above, we also need to clear the LSB digit of hours (0-9) if it counts to a value 10.

Based on the above logic, you need to design a macro for mod-24 counter.

Normal Clock (Mode-0) Operation Logic

The normal clock operational logic is shown in Figure 7.1. The left mod-60 counter is used for counting *seconds* and the right one to count *minutes*. The mod-24 counter counts *hours*. As visible from the diagram, the *chip enable* signal ripples to successive stages as the previous stage reaches its maximum value. Recall that a mod-60 counter generates its *chip enable output* (CEO) signal once it has counted sixty values. Thus, once the seconds counter has counted sixty seconds, it generates chip enable output (CEO) signal which lets the minutes counter to increment. Similarly, on every sixty values for minutes, CEO of the minutes counter causes an increment in an hour counter.

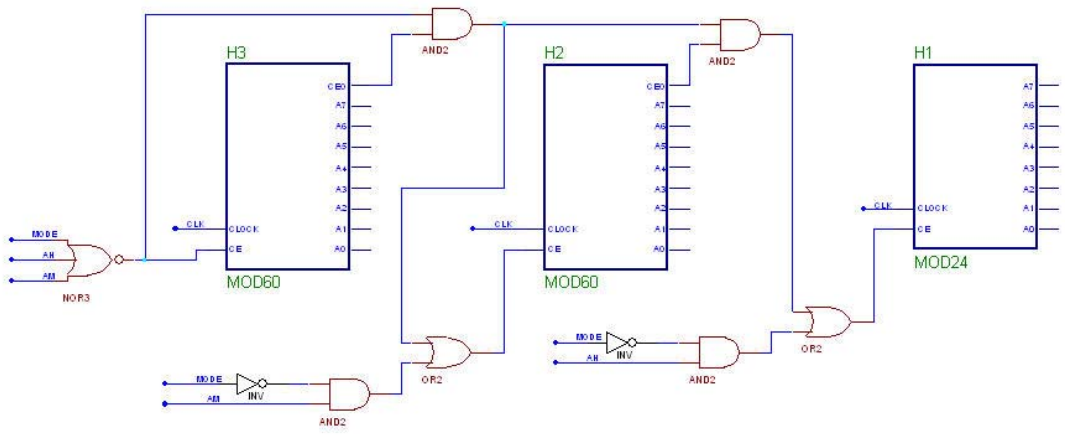

Figure 7.2: Normal clock operational logic

We will next investigate each stage closely.

The Seconds Stage

The 'Seconds' stage is zoomed-into in Figure 7.2. Note the chip enable logic for this stage. We need to disable this counter if either we are in alarm-setting mode (mode-1) or are trying to advance minutes or hours.

The chip enable output for next stage, counter-minutes, is generated once the counter reaches its maximum value (59) and the seconds counter is still enabled, i.e. the counter will be rolling-back to value-0 on the next clock cycle.

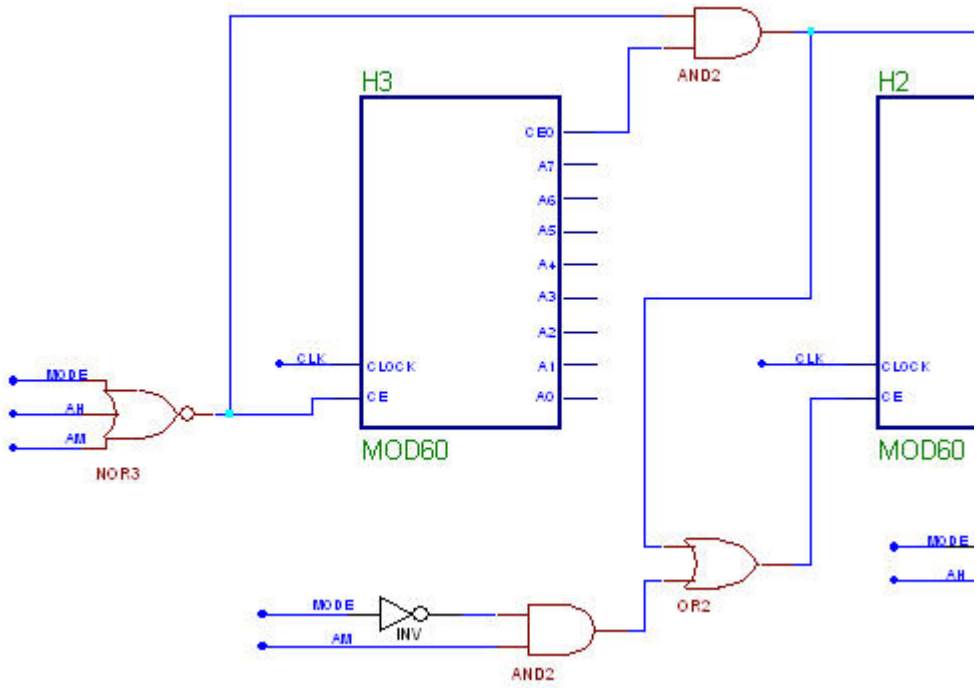

Figure 7.2: Seconds Stage

The Minutes Stage

The 'Minutes' stage is zoomed-into in Figure 7.3. The minutes counter is to increment either after sixty seconds or if the user is trying to increment minutes. Verify if the chip enable logic shown in Figure 7-4 can achieves this.

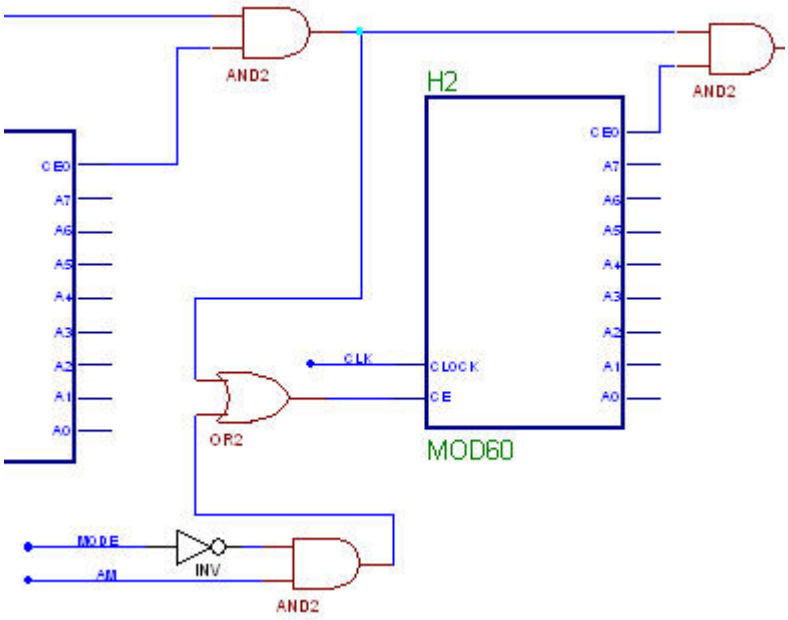

Figure 7.3: Minutes Stage

The Hours Stage

The 'Hours' stage is shown in Figure 7.4. The functionality of the stage is very similar to minutes stage. The chip enable signal provided by the top AND gate is provided when both the CEO of seconds and minutes counters are high and the clock is running in normal mode. That is, when the hours counter increments its value, the other two counters roll-back to zeros.

The lower AND gate in Figure 7-5 enables the hours counter for incrementing hours using 'AH' signal.

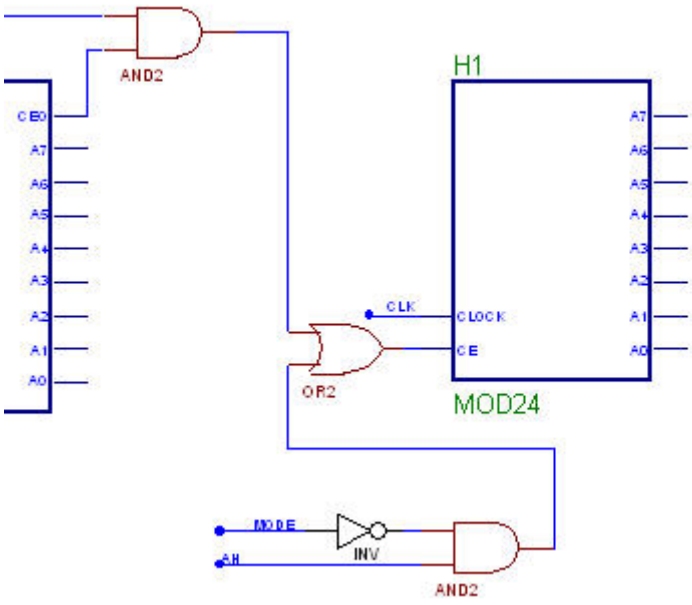

Figure 7.4: Hours Stage

Alarm Mode (Mode-1) Operation Logic

Figure 7-5 shows the logic required for storing minutes and hours for alarm time. The chip is enabled whenever *mode* signal is high and user tries to set time using AM and AH signals.

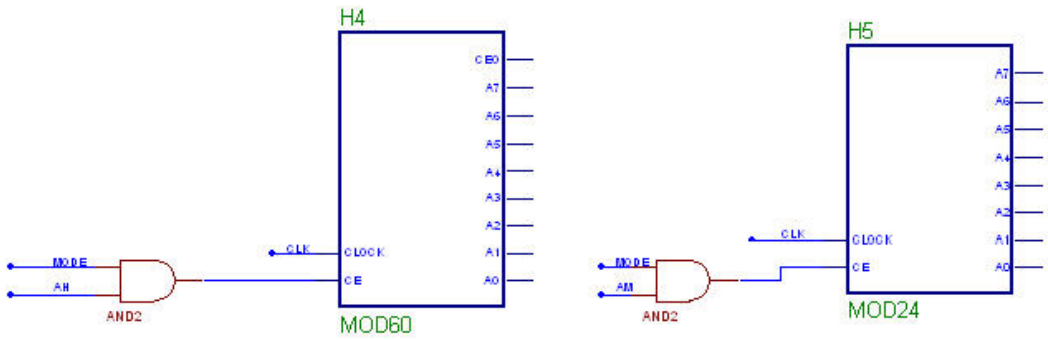

Figure 7.5: Alarm Mode Operational Logic

The Clock

Recall, that our specifications required running the clock at

- 1 Hz in normal operation
- 2 Hz (Fast mode) during time setting.

This can be achieved as shown in Figure 7-6 using a multiplexer. A *multiplexer* (also called a *selector*) is a circuit that accepts a number of inputs and outputs one of them depending on some control inputs on the selection lines. In our case, we are using a 2-1 MUX (M2_1). This component (M2_1) accepts two data inputs, D0 and D1, and depending on the value of S0 (the selection line), one of the data lines will be output through the output line O. If S0=0, then O=D0. Otherwise O=D1.

Clock1 and Clock2 are two clocks at 1 Hz and 2 Hz respectively. The output of your 1 Hz macro developed previously will feed the net Clock1. You need to modify that macro to generate a 2 Hz clock which will be connected to Clock2.

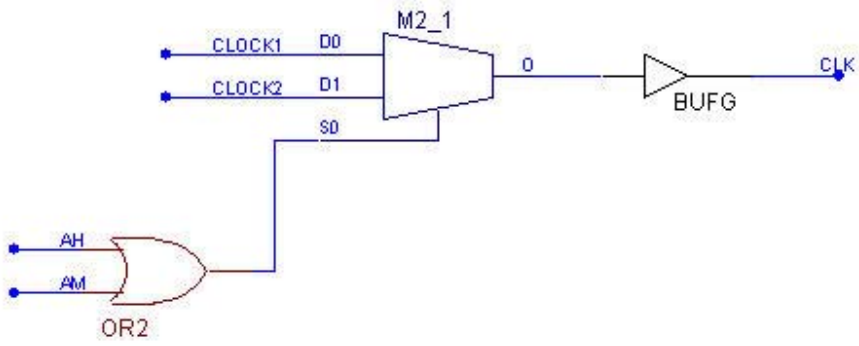

Figure 7.6: Clock Selection logic

A special global buffer, BUFG, is used to derive the clock. The global buffer ensures that the clock signal is not only buffered but it also reaches the different regions at the same time, a very important requirement for correct operation of clocked circuits.

Pre-Lab

- Bring the following from the previous labs
  - 1 Hz clock macro you developed in previous lab.
  - MOD60 counter built in the previous lab
- Develop 2 Hz macro from the 1 Hz macro.
- Design and bring, modulo-24 counter. Call it as MOD24.
- In the time setting logic for either normal running clock mode or for alarm mode, we have shown how the respective counter is enabled. Once that counter is enabled, explain how a user defined count value is going to be stored.

In-Lab

- Draw the schematic for the normal running clock and for alarm logic on one sheet. Recall from the lab-manual that two nets named identical are connected together internally, though they may both appear to be floating. Use this feature efficiently to avoid complex connections and to make your schematic look as simple as possible.
- Perform integrity test. Ignore warnings related to floating pins of the macros.
- Simulate the design.
  - You can easily add the pins of your macros as well as other input/output nets using the on-schematic simulation toolbox.
  - Join the related nets of the counters into bus in the simulation window. This can be achieved by first selecting all the desired nets then right clicking and clicking on bus\combine.
- Demonstrate your work to the instructor.