ELE302

Spring 2010

Project 1*
LED Digital Clock

Due 3/1/10 – 4:00 PM

Project Description:

Your assignment is to construct a digital clock that exactly meets the physical and functional specifications included here. The circuit must exactly match the schematics provided. The software must be written in 56800 series assembly language, but otherwise you have complete freedom as to the structure of the software and the algorithms and methods you use. Schematics, a bill of materials (parts list) and kit assembly instructions are provided. Data sheets for all key components are included as well.

The project is a 24-hour digital clock that displays time in hours, minutes, and seconds as well as having AM and PM indicators. The clock display uses a 12-hour format. A run button and a stop button determine whether or not the clock counts or holds the current time. Six other buttons are used to set the hour, minutes, and seconds.

Physical Specifications:

Your project should look like the prototype that was shown in class. Follow the instructions to assemble the kit. The 6 LED modules should be at the top showing hours, minutes, and seconds. The AM and PM indicators will be to the right of the second digits with the AM indicator on top. The behaviors of the buttons are defined as follows:

Button Name	Function
S0	Stop Time
S1	Start Time
S2	Decrement Hours
S3	Increment Hours
S4	Decrement Minutes
S5	Increment Minutes
S6	Decrement Seconds
S7	Increment Seconds

Functional Specifications:

The primary function of the system is that is must keep the correct time. The display must normally show the current clock setting. When the Stop button is pushed, the display stops counting and remains unchanged. Optional features may employ special display modes. When the Start button is pushed, the clock increments its display by one second at one second intervals. **The clock should be accurate to within 10 seconds per day** (in operation on your lab bench).

The remaining 6 control buttons are used to set the time. They operate whether or not the clock is running. The Increment button can be pushed to advance the time display or the Decrement button can be pushed to reverse the time display. Each time a button is actuated, it increments or decrements the time by one unit. This is one second for S6 and S7, one minute for S4 and S5, and one hour for S2 and S3. This means that if the clock is set at 12:00:00 AM and S6 is pushed then the display should read 11:59:59 PM.

If a button is pushed for less than .5 seconds, it only changes the time by one unit. However, if the button is pushed and held for a longer period of time, then it changes the time more rapidly. **When a button is held, the time display should change by one unit as soon as the button is pushed, stay unchanged for .5s and then begin to change continuously at a rate of 1 unit every .2-.25 seconds.** If more than one time setting button is pushed, they should operate simultaneously. The display should be bright enough to be comfortably read and should not visibly flicker.

Optional Features:

A working clock is the absolute requirement; the project must be completed before work is started on the car projects. Addition of optional features will not be taken into account in evaluating the project, but everyone is encouraged to demonstrate ingenuity and creativity in the completed clock. So, for example, it is possible to add special display modes triggered on the hour, quarter-hour, and half-hour, either using the clock board display itself or using additional custom hardware. Alternately, an alarm or timer feature might be implemented.

Documentation & Demonstration Requirements:

Each group will make a design review presentation during the week of February 22. You must demonstrate a fully operational system by **4:00 PM on Mar. 1. 2010**. Although you will use the JTAG port on your processor to download and debug code for development, the final system must be fully independent. The only external connections will be the two wires to the power supply.

You documentation must include the assembly language listing of the program as well as a write-up describing the operation of your system. This must include definitions of each variable, a functional description of each major section of code, and an overall description of the mechanisms used to instantiate each important function of the system. Some of this can be included as comments in the original code listing. You should also include a measure of your minimum RAM and FLASH requirements and a derivation showing that you have met the accuracy requirement. **This is due by 1:30 PM on Mar. 5.** Your lab notebook should also be kept current throughout the project and made available for review on Mar. 5.

Theory of Operation:

The system consists of two parts, the microprocessor demo board and the clock kit board. The microprocessor demo board is manufactured by Motorola as a way for embedded system developers to quickly evaluate the microprocessor and prototype applications. The clock kit board is a custom design for this course and will interface with the microprocessor demo board by means of a 10-wire ribbon cable.

To begin, look at the schematic for the 56F800 demonstration board. It is on page 24 of the 56F800 Demonstration Board User's Manual, which is available on the course webpage. If you are familiar with microprocessors, one thing you may notice is that this processor does not have external data or program busses. In this version of the processor these have been left out to save pins and reduce cost. Instead, the program code is stored on FLASH memory inside the chip and data is stored in SRAM that is also on-chip. This results in a very inexpensive (less than $3) and easy to use microprocessor. If you look carefully at the schematic, nearly all of the components on the demo board could be left out and the microprocessor would still function. So, if a designer wants to add this processor to a system, it can essentially just be dropped in, and without any external support hardware it will function. You see why embedded processors like this one have made their way into nearly every electrical or electronic product made today.

An important feature of this demo board is the HOST JTAG port and supporting hardware that can be found towards the right side of the schematic. This connector is driven by the parallel port of a PC to allow a developer to download program code to FLASH memory and to do real-time debugging of the code by setting breakpoints and viewing the contents of system registers and memory. In a real commercial system, the logic portions of this port would be left off and only a few wires would be accessible by some external programming device that would load the program code onto the microprocessor before it was packaged.

Now, let’s turn our attention to the clock board. Take a look at the schematic and block diagram of the clock board. Both are available on the course website. You will connect your clock board to the 56F800 demonstration board by a 10-wire ribbon cable, but you will only use 8 wires from this cable. 7 of the wires will be configured as outputs of the processor; 3 of these outputs form an address that determines which display is written to and which switch is read. The other 4 outputs are a binary coded decimal that is displayed onto the LED displays. The CD4511 BCD-to-7-Segment LED decoder driver takes the BCD input, decodes it, latches it, and then drives high current outputs that can provide enough current to illuminate the LEDs.

Since we have 3-bits of address, 8 possible devices can be selected, 6 of which are used for the LED displays, one of which is unused, and the last one is used to select a J-K flip-flop that drives the AM/PM indicators. By selecting the J-K flip-flop and providing a certain pattern of bits to the inputs you will be able to switch the AM/PM indicators. See the 74F112 datasheet on the course website to review how a J-K flip-flop works.

Lastly, the same 3 bits of address are used to multiplex the 8 switches onto one input line that is fed into the processor. This means that you will need to scan through all 8 addresses to read all 8 buttons. This must be done sufficiently fast to provide a good response time when a button is pushed. So, although the displays can latch a value indefinitely, you will need to refresh them frequently in order to scan all the buttons.

*** Rev. 1 of project description, Feb. 1, 2010. There may be some minor revisions.**