Overview

The project consists of designing a microcontroller in VLSI, using the software package LEDIT. The system creates delays based upon user inputs through the accumulator.

Figure 1: System Block Diagram

System Block Diagram

The user enters an 8-bit input into the accumulator. Through the register controller, the user can also enter a 2-bit value indicating which register the accumulator value will be stored in. The system will have four registers controlled by clock inputs, as shown above in Figure 1. The 8-bit value stored in each register becomes the input to a
comparator. The value will be compared to the lower 8 bits of a 16-bit incremental timer. The comparator will output a single bit.

System Inputs and Outputs

Figure 2 shows the user inputs and outputs to the system.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bits into accumulator</td>
<td>1 bit out of each comparator</td>
</tr>
<tr>
<td>2 bits into the register controller</td>
<td>16 bit timer value</td>
</tr>
<tr>
<td>clock pulse to control timer and registers</td>
<td></td>
</tr>
<tr>
<td>1 bit reset to timer</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: List of Inputs and Outputs

Timer

The main subsystem of the project is the 16-bit timer shown in Figure 3 below. The timer will drive each of the other subsystems and create any necessary delays. Although the timer is only being compared to 8-bit registers, the project will use a 16-bit time to create longer delays. It will increment from 0000h to FFFFh, return to 0000h, and repeat the process. Whenever the reset input value is set high, the timer can be made to restart at 0000h.

Figure 3: Timer Subsystem

Registers

The registers can be updated according to user’s specifications at any time during the program. The user inputs an 8-bit value into the accumulator, as shown in Figure 4. It is through the register controller that the user selects which register to change to the
value that is in the accumulator. The register values are continually compared to the incremented timer through the comparator subsystem.

Comparators

There will be one comparator for each register, as shown in Figure 5. The inputs to the comparator will be the 8 bits from the register and the lower 8 bits from the timer. If the two values are the same, the comparator will output the binary bit ‘1’, otherwise it will output the binary bit ‘0.’ Each of the four comparators will output a single bit. These values can be used as external outputs or can be used to trigger various interrupts.