

Final Design Project
“Interval Trainer”
Austin Stockwell
ECET 16400
Professor Kishan Ranasinghe
12-15-2014

Objectives:

The purpose of this project was to utilize all the knowledge gained in ECET 109 and ECET 155 in order to create a musical “interval/scale trainer”. The user is to guess the next “note” in the “scale”—any wrong answers were displayed in the 7-segment displays, and the amount of wrong answers were added and displayed on a separate 7-segment display. The design utilized three separate MOD counters, one decoder circuit, an active HIGH sub-circuit, and active LOW sub-circuit, one frequency divider, a comparator, a latching circuit, an adder, five LEDs, four 7-segment displays, eight switches, and one push button.

Equipment:

- Altera DE-2 FPGA
- Quartus 13.0 Design Software

Procedure:

The first task was to create transition tables for all three “scales” (each scale is represented by a separate MOD counter). The chromatic MOD counter is a MOD 13 in which counts from 0000 to 1100. The major MOD counters states were as follows: 0000, 0010, 0100, 0101, 0111, 1001, 1011, and 1100. The minor MOD counters states are as follows: 0000, 0010, 0011, 0101, 0111, 1000, 1010, and 1100.

After the three separate “scale” MOD counters were designed, they were implemented in Quartus (Fig 12-14). Notice each MOD counter has a separate “sense and set” circuit” that sends a pulse out of an output pin when each counter reaches 1100 (decimal 12). This pulse in turn clocks a separate MOD 10 (binary 0-9). The output of the second MOD counter is sent to a 7447 7-Segment Display

(HEX7) that is used to count how many times each “scale” has been played (fig 15). For example, if a scale MOD counter has progressed through its entire sequence once, “1” will be displayed on the 7-Segment Display.

Next, a four output decoder circuit was implemented that was used to select which of the three separate “scale” MOD counters is to be clocked (fig 16). In order to do this, two switches (SW1 AND SW0) are connected to an array of AND gates. In position 00, an ACTIVE LOW sub-circuit is utilized that clears each scale and 7-segment display (no clock will be passed through the decoder when this option is selected). In position 01, the “chromatic scale” MOD counter will be clocked. In position 10, the “major” MOD counter will be clocked. In position 11, the “minor” MOD counter will be clocked.

Once a “scale” selection is chosen, the user has the option to choose whether or not the chosen MOD counter is to be clocked automatically via PIN N2 (the internal clock), or manually via pin W26 (KEY 3). This is achieved by using a two input OR gate that routes the signal according to what position switch S6 is set to (fig 18).

If SW6 is active HIGH, then the clock signal coming out of the decoder circuit (PIN N2) is sent into a frequency divider (fig 17) in order to slow the counter down to 1 HZ. To achieve this, the original 50Megahertz signal coming from PIN N2 was cut down to 10 Megahertz using a MOD 5 (fig 19) counter with a sense and set circuit that sends an active high when binary 100 is read on outputs Q3, Q2, Q1 (once every 5 clock signals). Thus, the 50Megahertz signal was successfully cut into a 10Megahertz signal. To further reduce the signal to 1 Hertz, 7 separate MOD 10s

(fig 20) (also with a sense and set circuit)) were used. As seen in fig17, the output signal of each mod counter is sent into the CLK of the next MOD counter. This effectively slowed the rate of the internal clock to sequence the counters at a rate that can easily be seen and interpreted by the user. The clock coming from the frequency divider is then sent to the selected scale's CLK.

If the user decides to clock the MOD counters manually, SW6 is set to ACTIVE LOW. Thus, the OR gate in fig 18 allows the clock to be passed via push-button (fig 18).

After the user decides what CLK source is to be used, he/she must progress through each of the counter's states while keeping LEDG0 (PIN AE22) lit the entire time. In order to do this, the user must enter the correct NEXT state of the selected counter using SW5-SW2 (with SW2 being the LSB). The users inputs are compared with the outputs of the selected MOD counter (Q3-Q0) (fig 21)) with a 7485 IC. If the answers are equal, LEDG0 will light—and remain lit unless the user inputs the wrong “next” state. If a wrong answer is given, a separate MOD 10 (0-9) will be clocked—and the number of wrong answers (Q3-Q0 of the MOD 10) will display on either HEX5 (less than) or HEX4 (greater than).

As seen in figure 21, eight separate “D-Latches” are utilized to act as a memory circuit—used to ensure that the user's “answers” are only compared to the current counter state when he/she desires (using ENABLE/SW7). The process is explained in the following sentences.

The user must first stop the automatic clock by setting SW6 to active LOW. The user must then set binary 0000 on SW5-SW2. The user then selects a pattern.

Once the pattern is selected, and the switches and Q's are all equal, LEDG0 (PIN AE22) will light. The user must then ENABLE the latch by setting SW7 to active LOW (latch ENABLE). This will lock in the "last" state (0000 on the switches and 0000 on the selected MOD/"scale") and will allow the user to change the pins to the "next" state without causing the LED0 (equal) to turn off. Once the user enters the next count (for example, 0001 in the chromatic scale), the user will CLK the desired MOD counter using KEY3 (advancing the counter from 0000 to 0001—matching the users switches inputs). Thus, both the counter and the switches should be equal once the ENABLE (SW7) is returned back to logic level HIGH.

If the outputs are equal, LEDG0 (PIN AE22) will stay lit. If the user's inputs don't match the current state of the selected MOD counter, then the comparator will output either "greater than" or "less than". Each separate GT/LT pin of the 7485 comparator is sent to a separate MOD 10 counter (decimal 0-9) that is in turn sent to a separate 7447 7-segment display. If the user gets the answer "wrong", an active HIGH will be sent along either the "greater than" or "less than" pins of the 7485 comparator IC—clocking the next mod counter, and displaying both the number of instances "higher than" and the number of instances "lower than" on separate 7-segment displays. The number of "less than" is displayed on HEX 5 display, while the number "greater than" is displayed on the HEX 4 display.

While the user is advancing through the game, the number of wrong answers are added together using a 7483 4-bit adder IC, and their sum is displayed on HEX3 (fig 22). This is achieved by sending the outputs of each of the two MOD 10 counters (which hold the number of "greater than" or "less than" instances) to the inputs of

the 7483 adder IC. A1-3 of the 7483 adder are connected to Q3-Q1 of the “less than” MOD 10, while B1-B3 of the 7483 adder are connected to Q3-Q1 of the “greater than” MOD 10. The outputs of the 7483 adder were then wired to a fourth 7447 7-Segment Display.

Data:

Fig 12 (Chromatic Scale w/sense & set)

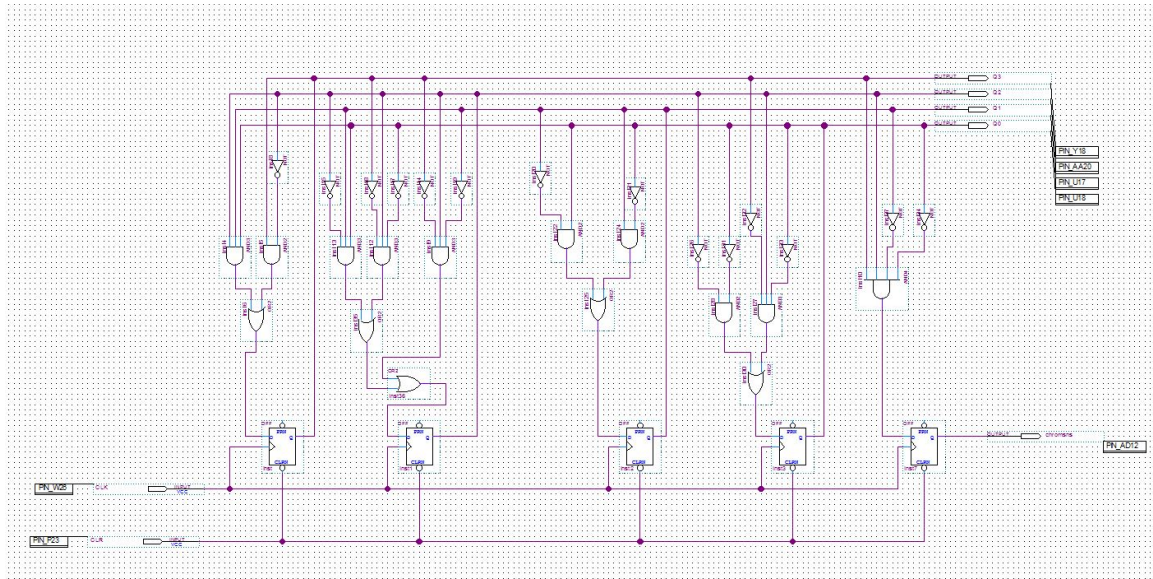


Fig 13 (Major Scale w/sense & set)

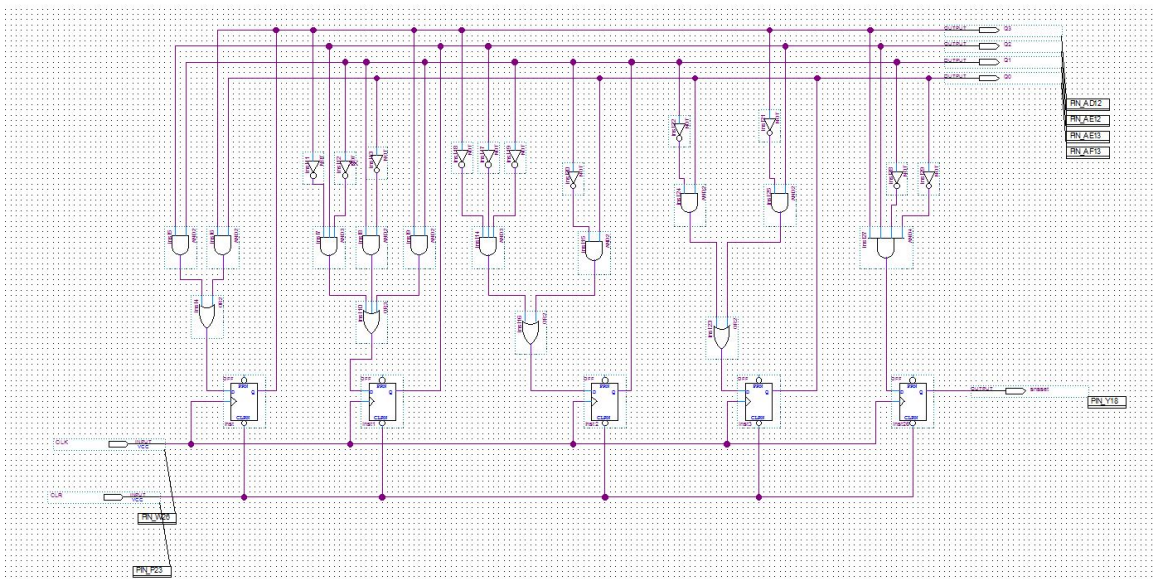


Fig 14 (Minor Scale w/ sense & set)

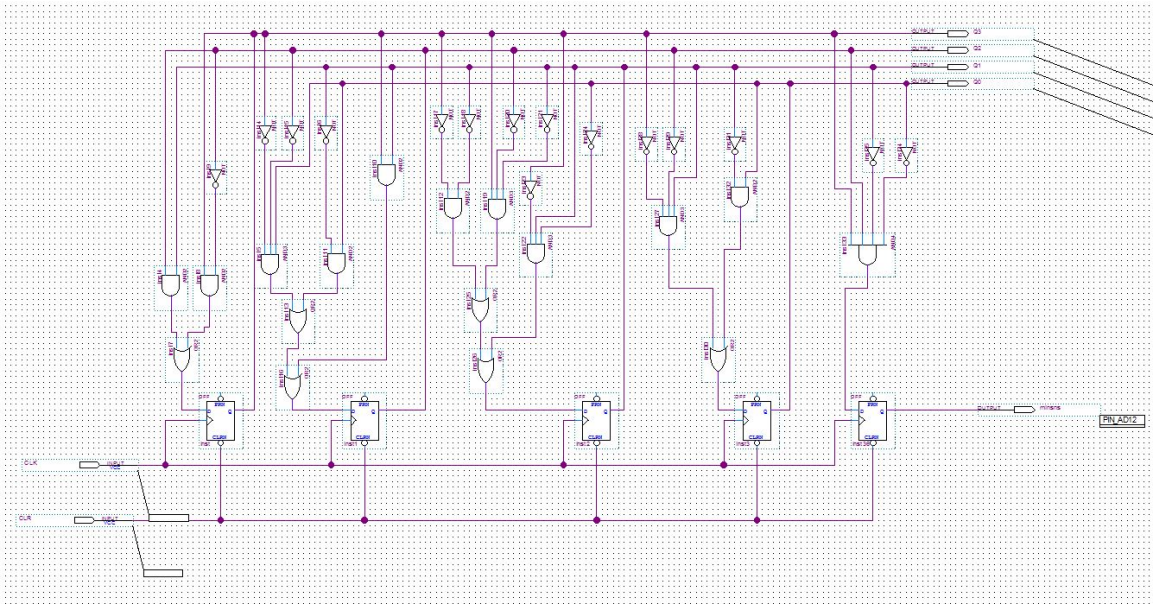


Fig 15 (MODS to separate counter to “7447” (HEX7))

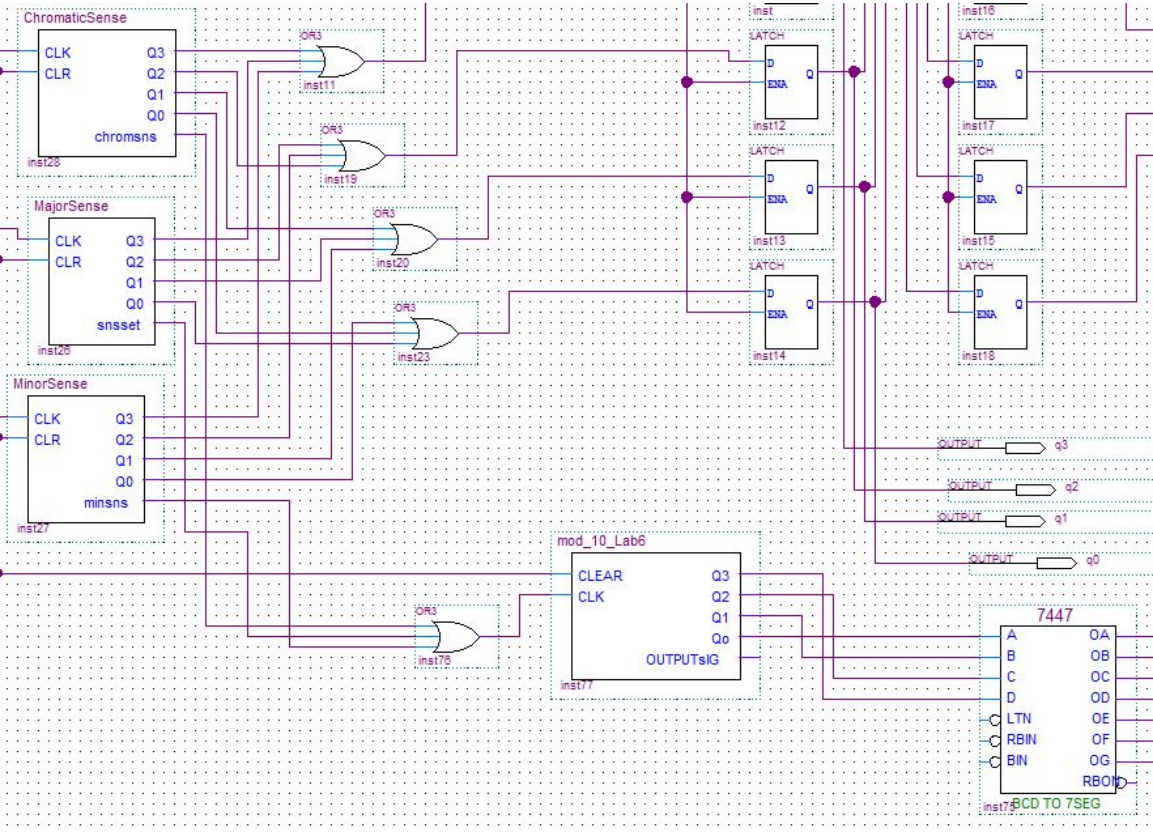


Fig 16 (Decoder circuit)

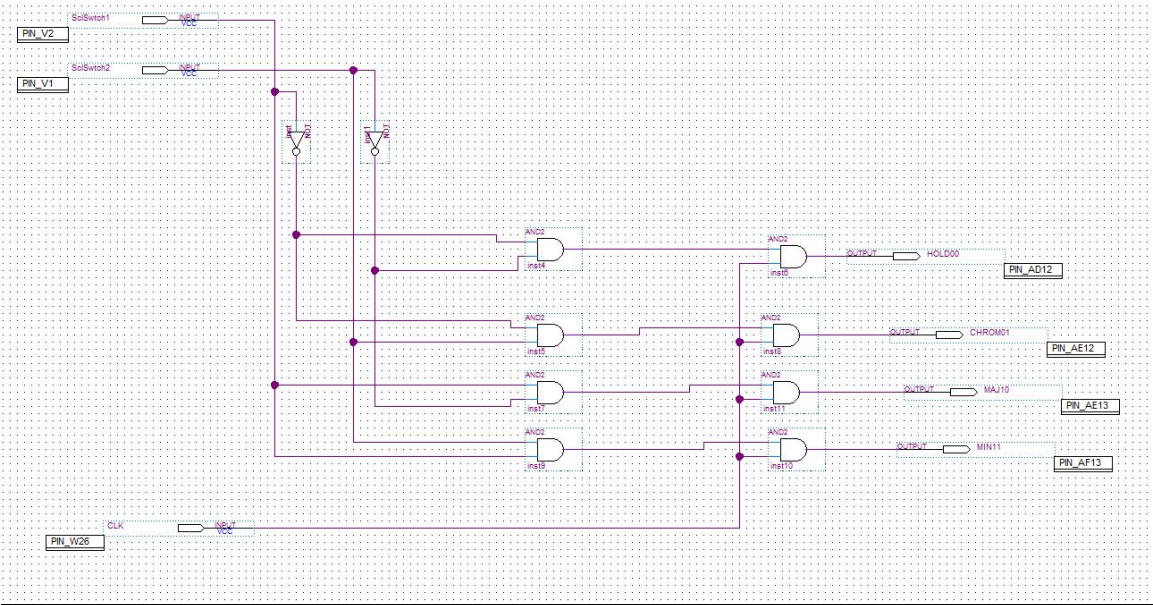


Fig 17 (Freq Divider)

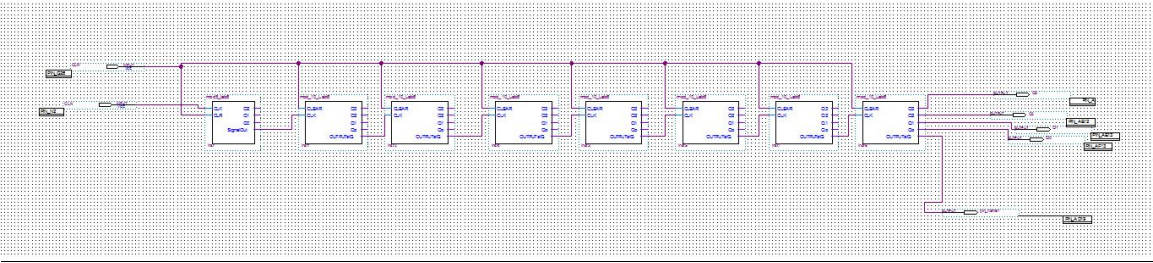


Fig 18 (Clock Choice)

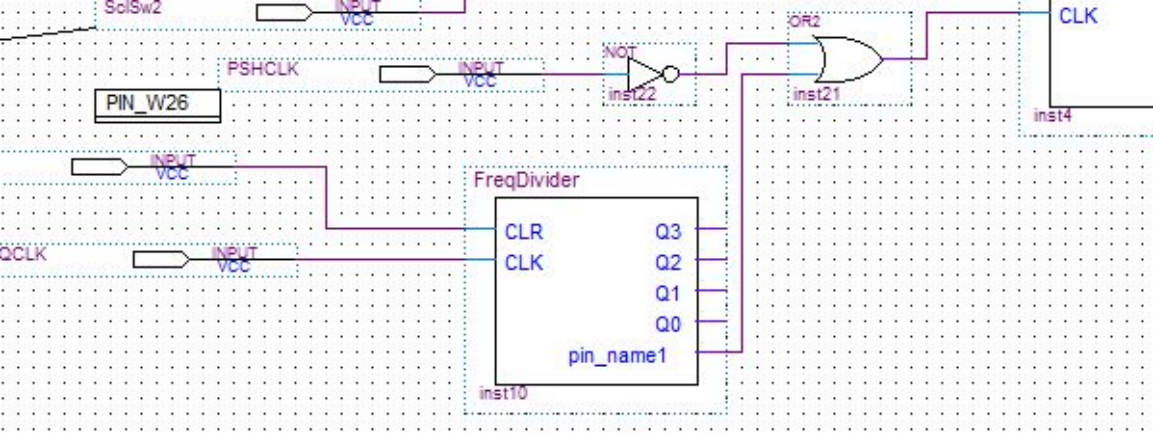


Fig 19 (MOD 5 w/sense & set used in FREQ Divider)

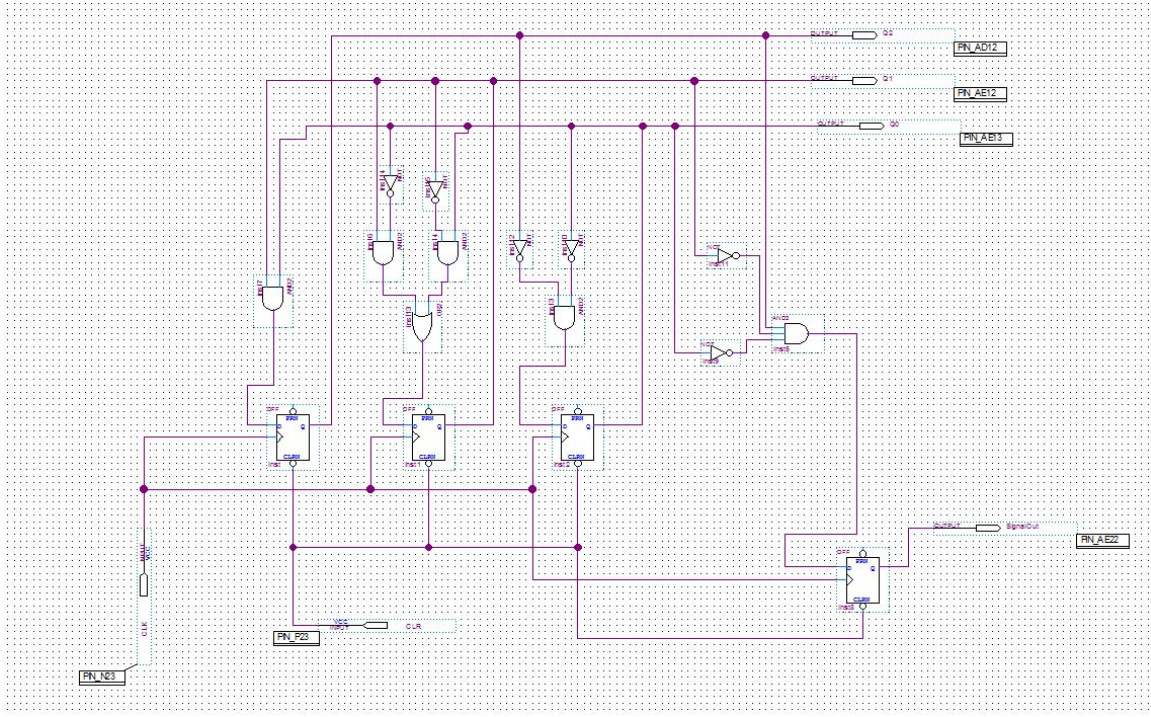


Fig 20 (MOD 10 w/sense & set used in FREQ Divider)

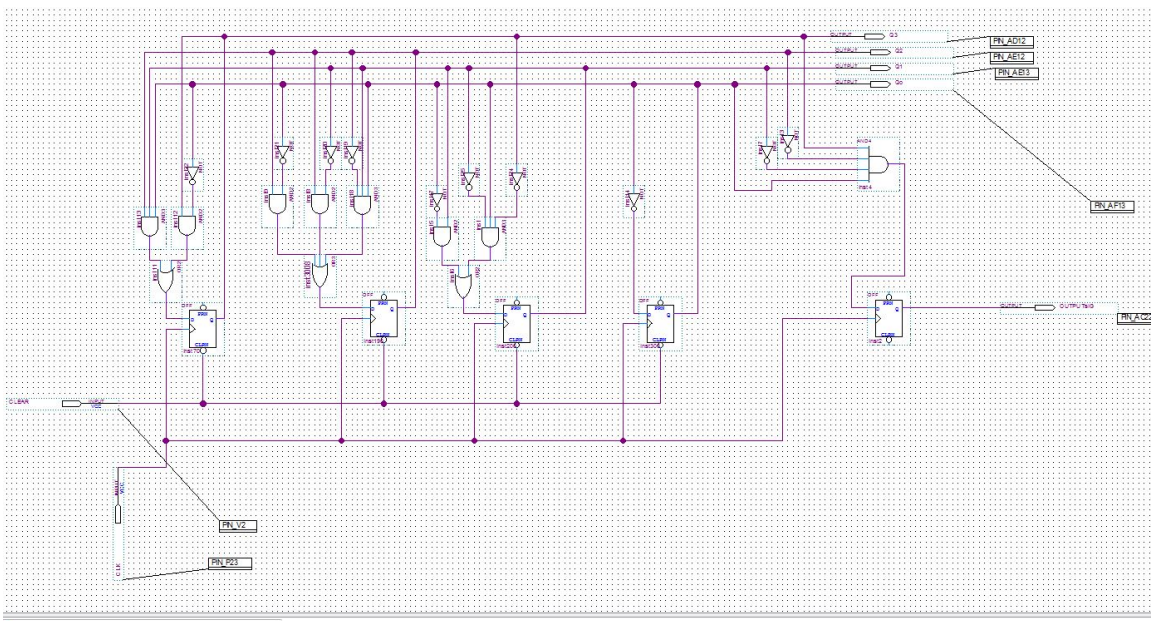


Fig 21 (Comparator Circuit)

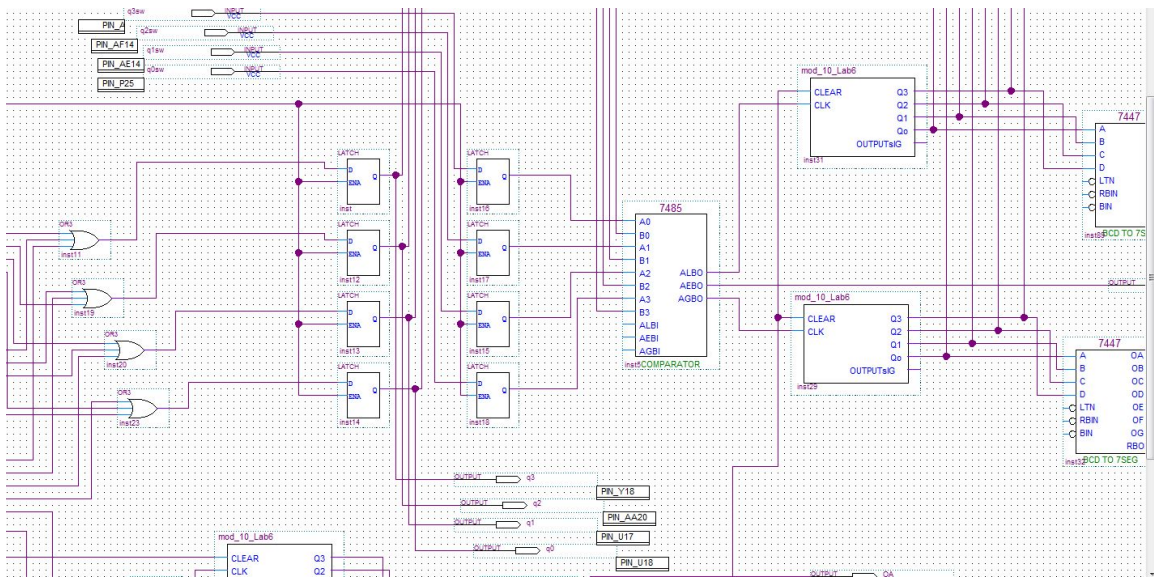


Fig 22 (Adder of wrong answers)

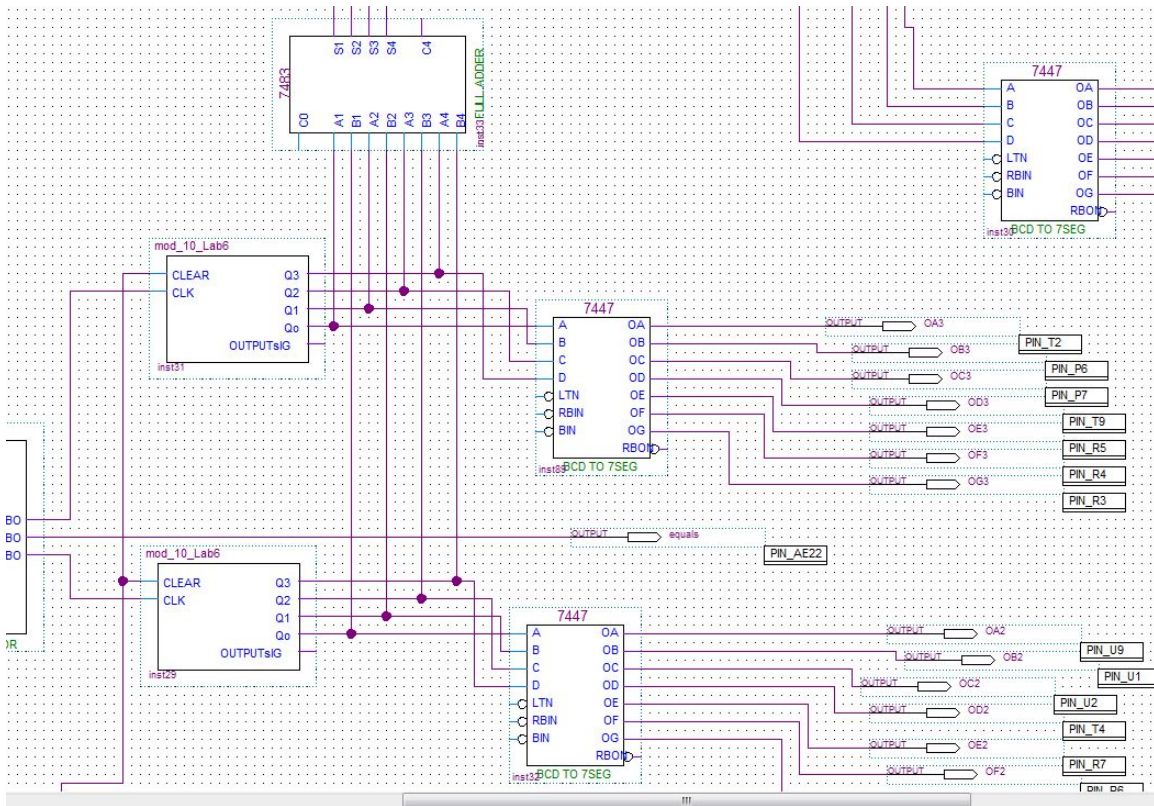


Fig 23 (Latching Circuit)

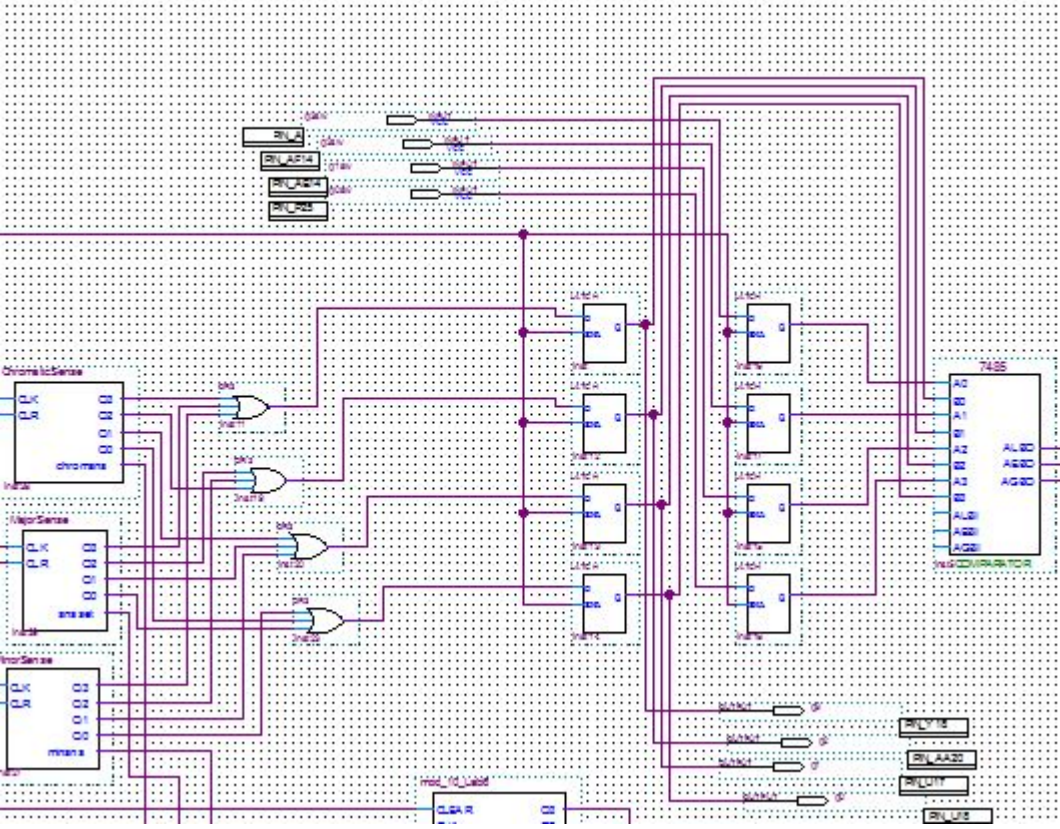


Fig 24 (Final Design)

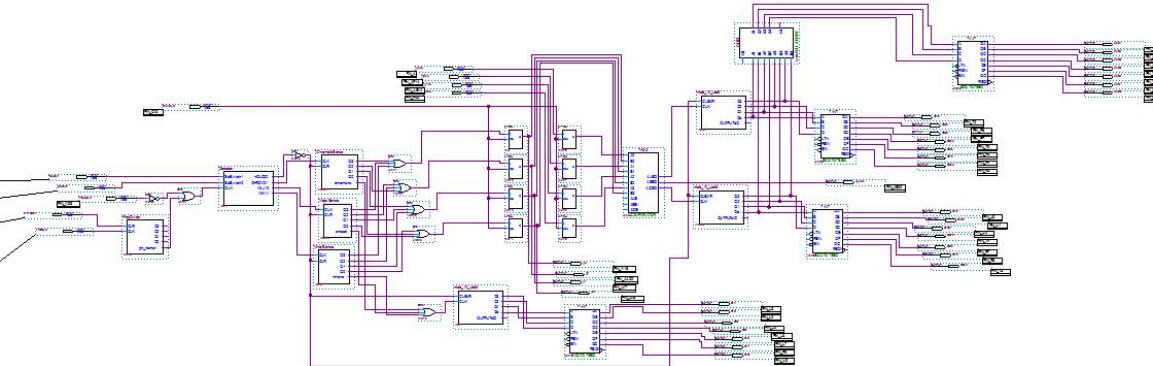


Fig 24 (Chromatic MOD Transition Tables)

Chromatic

Austin State
Final Project

Present								Next			
q_3	q_2	q_1	q_0	D_3	D_2	D_1	D_0	q_3	q_2	q_1	q_0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	1	0	0	1	1
0	0	1	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	1	0	1
0	1	0	1	0	0	1	0	0	1	1	0
0	1	1	0	0	0	1	1	0	1	1	1
0	1	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	0	1	0
1	0	1	0	0	1	0	1	1	0	1	1
1	0	1	1	0	1	1	0	1	1	0	0
1	1	0	0	0	0	0	0	0	0	0	0

$q_3 q_2$

00	0	0	0	0
01	0	0	1	0
11	0	X	X	X
10	1	1	1	1

$D_3 =$

$q_3 q_2$

00	0	0	1	0
01	1	1	0	1
11	0	X	X	X
10	0	0	1	0

$D_2 =$

$q_3 q_2$

00	0	1	0	1
01	0	1	0	1
11	0	X	X	X
10	0	1	0	1

$D_1 =$

$q_3 q_2$

00	1	0	0	1
01	1	0	0	1
11	0	X	X	X
10	1	0	0	1

$D_0 =$

Fig 25 (Chromatic Equation)

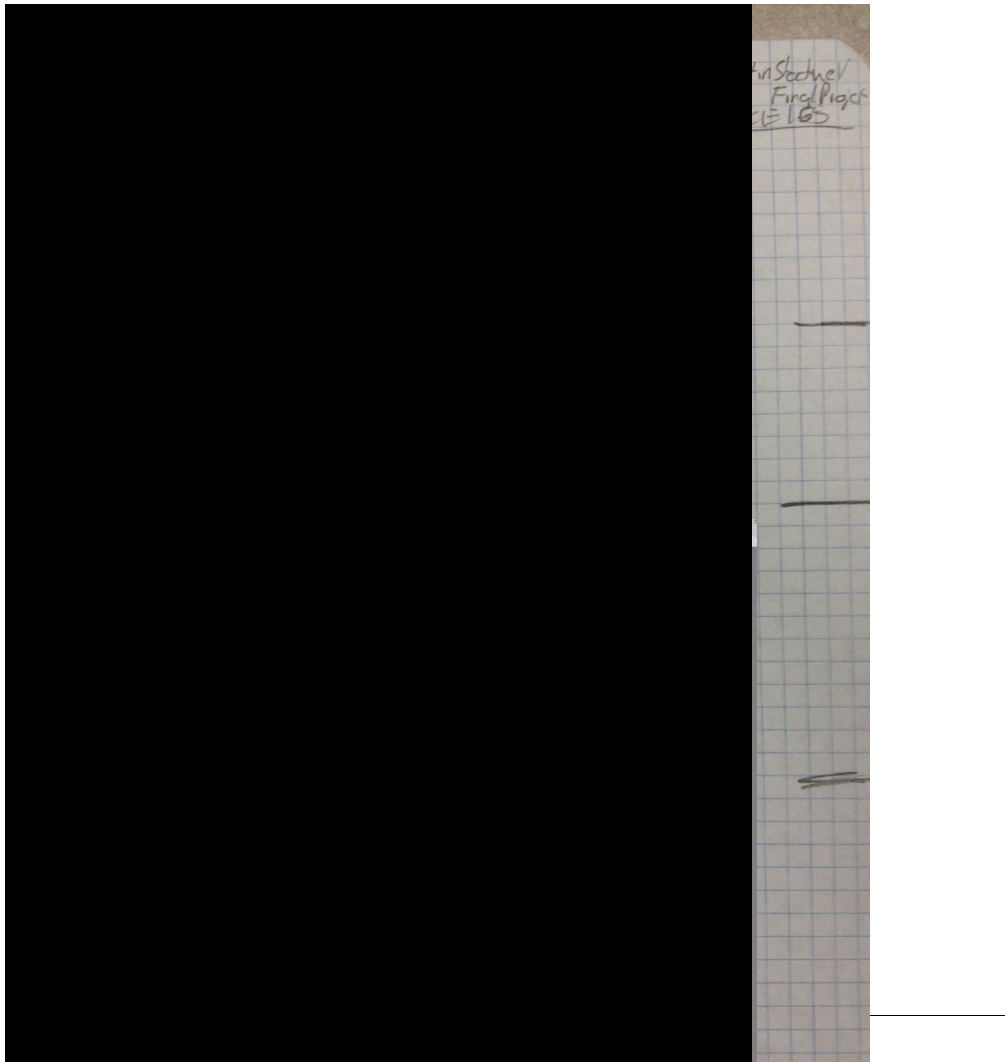


Fig 26 (Major MOD Transition Tables)

The image shows two hand-drawn MOD transition tables on graph paper. Each table is a 4x4 grid with a 2-bit input (q1, q0) and a 2-bit output (Q1, Q0). The inputs are labeled 00, 01, 11, 10 and the outputs are labeled 00, 01, 11, 10. The tables are for D1 and D0.

Table 1 (D1):

q1 \ q0	00	01	11	10
00	1	X	X	0
01	0	1	0	X
11	0	X	X	X
10	X	1	0	X

$D_1 =$

Table 2 (D0):

q1 \ q0	00	01	11	10
00	0	X	X	0
01	1	1	1	X
11	0	X	X	X
10	X	1	0	X

$D_0 =$

Fig 27 (Major MOD Equations)

Major Scale

Austin Slackwell
Fall 1960

$Q_3 \quad Q_2 \quad Q_1 \quad Q_0$

0	1	1	0	+	1	1	0	1
0	1	1	0		1	1	0	1
0	1	1	0		1	1	0	1

$D_3 = Q_2 Q_1 + Q_3 Q_0$

$Q_3 \quad Q_2 \quad Q_1 \quad Q_0$

0	1	0	0	+	0	0	1	0	+	1	1	1	0
0	1	0	0		0	0	1	0		1	1	1	0
0	1	0	0		0	0	1	0		1	1	1	0

$D_2 = \overline{Q_3} \overline{Q_2} \overline{Q_1} + Q_1 \overline{Q_0} + Q_3 Q_1$

$Q_3 \quad Q_2 \quad Q_1 \quad Q_0$

0	0	0	0	+	0	0	0	1
0	0	0	0		0	0	0	1
0	0	0	0		0	0	0	1

$D_1 = \overline{Q_3} \overline{Q_2} \overline{Q_1} + \overline{Q_1} \overline{Q_0}$

Fig 28 (Minor MOD Transition Table)

Austin Yokell
Final Project

Present				Minor				Next			
q_3	q_2	q_1	q_0	q_3	q_2	q_1	q_0	q_3	q_2	q_1	q_0
0	0	0	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	1	0	1	0	1	0
1	0	1	0	1	1	0	0	1	1	0	0
1	1	0	0	0	0	0	0	0	0	0	0

$D_3 =$

$q_3 q_2$	$q_1 q_0$	00	01	11	10
00	00	0	X	0	0
01	00	X	0	1	X
11	00	0	X	X	X
10	00	1	X	X	1

$D_2 =$

$q_3 q_2$	$q_1 q_0$	00	01	11	10
00	00	0	X	1	0
01	00	X	1	0	X
11	00	0	X	X	X
10	00	0	X	X	1

$D_1 =$

$q_3 q_2$	$q_1 q_0$	00	01	11	10
00	00	1	X	0	1
01	00	X	1	0	X
11	00	0	X	X	X
10	00	1	X	X	0

$D_0 =$

$q_3 q_2$	$q_1 q_0$	00	01	11	10
00	00	0	X	1	1
01	00	X	1	0	X
11	00	0	X	X	X
10	00	0	X	X	0

0 2 3 5 7 8 10 12

Fig 29 (Minor MOD Equations)

$$\begin{array}{r}
 q_3 q_2 q_1 q_0 \\
 0110 + 1000 \\
 1110 \\
 1010 \\
 \hline
 \bar{q}_3 \bar{q}_2 q_1 + q_3 \bar{q}_2
 \end{array}$$

$$D_3 = \bar{q}_3 \bar{q}_2 q_1 + q_3 \bar{q}_2$$

$$\begin{array}{r}
 q_3 q_2 q_1 q_0 \\
 0011 + 0101 + 1111 \\
 0011 \\
 0101 \\
 1111 \\
 \hline
 \bar{q}_3 \bar{q}_2 q_1 + \bar{q}_3 q_1 + q_3 q_1
 \end{array}$$

$$D_2 = \bar{q}_3 \bar{q}_2 q_1 + \bar{q}_3 q_1 + q_3 q_1$$

$$\begin{array}{r}
 0001 + 1001 + 0110 \\
 0001 \\
 1001 \\
 0110 \\
 \hline
 \bar{q}_3 \bar{q}_2 + q_3 \bar{q}_2 + \bar{q}_3 q_1 \bar{q}_0
 \end{array}$$

$$D_1 = \bar{q}_3 \bar{q}_2 + q_3 \bar{q}_2 + \bar{q}_3 q_1 \bar{q}_0$$

$$\begin{array}{r}
 q_3 q_2 q_1 q_0 \\
 0011 + 0101 \\
 0011 \\
 0101 \\
 1001 \\
 \hline
 \bar{q}_3 \bar{q}_2 q_1 + \bar{q}_3 \bar{q}_1
 \end{array}$$

$$D_0 = \bar{q}_3 \bar{q}_2 q_1 + \bar{q}_3 \bar{q}_1$$

Results:

Once the circuit was completed and placed into one final project, the circuit worked according to the design. However, there was one flaw in the design—the “greater than” or “equal to” lines (coming from the 7485 comparator) only provide one pulse. If the user was to get two “less than” or two “greater than” values in a row, the counter that was attached to the 7485 would only count once. This is due to the fact that I was unable to find a way to make the “greater than” or “less than” lines pulse. Instead, the lines would become “high” if one wrong answer was given, and would remain “high” if yet another wrong answer was given on the next clock/latch. Thus, the user would see “1” in the “wrong section” rather than “two”. This would not happen if the user was to alternate between correct and incorrect answers.

Another problem that was encountered although was eventually remedied was finding a way to compare the numbers only when the user decided to. If the user was to enter the “next state” of the mod counters on SW5-2 while the current state/pins match, the moment SW5-S2 were changed (without the latch circuit), the “equal to” LEDG0 (PIN AE22) would no longer light. To fix this, an 8-bit “latching” circuit was implemented (fig 23). Thus, the user could “latch” the previous “answer”/state—and could change the pins to match what the NEXT state would be without having the change of the pins be reflected at the inputs of the comparator. This allowed for the user to compare the states of the pins/counters when they preferred.

Conclusion:

Overall, the objectives of the final project were met. A “scale/interval” trainer was successfully implemented using all but a ripple counter in the design. The design utilized three separate MOD counters, one decoder circuit, an active HIGH sub-circuit, and active LOW sub-circuit, one frequency divider, a comparator, a latching circuit, an adder, five LEDs, four 7-segment displays, eight switches, and one push button.