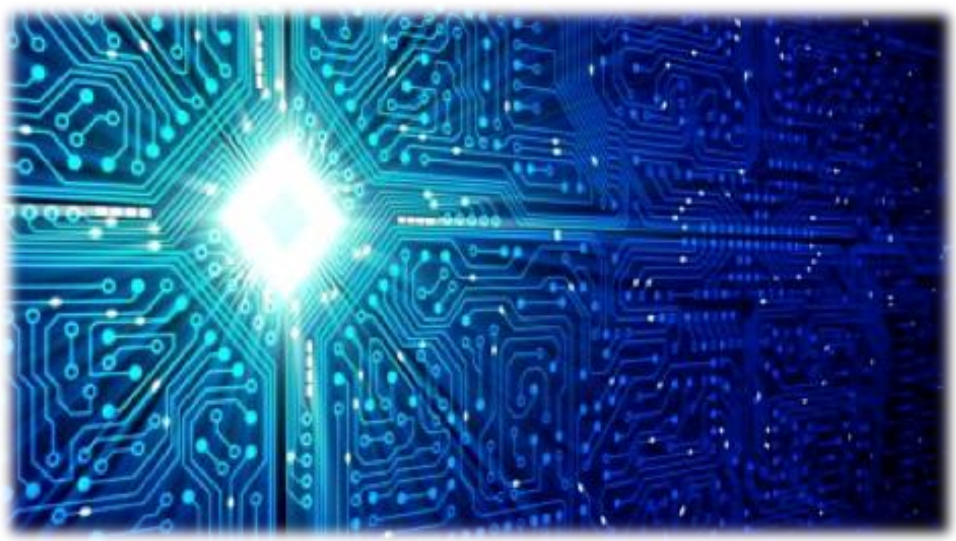


The Microwave Project

A Digital Electronics and Microcontrollers Project

Arody Deleon | Suhail Prasathong

Table of Contents

Project Description

Executive Description

The Microwave Project is an application of the compilation of digital electronics skills acquired through the semester under the supervision of Professor Bertam Gamory, Monroe Community College. Through intricate circuitry, digital logic and enhanced wiring techniques we were required to create a generic over-the-counter microwave oven control system. The project required the inclusion of a power system, timer aspects, a microwave generation integrated system, a carousel plate, housed lighting and door sensors. We went through the process of trial and error development while attempting to wire various aspects of the project. We faced quite a few hurdles while converting our schematic from Multisim to Altera. Then we went on to wire the different outputs and link that to the schematics written up on the Altera software system. Our hard-wire outputs consist of servo motor, speaker, power motor, lights and vent fans. After we wired up the Quartus system, we connected the Altera board with our breadboard through the ribbon cable connecting all our systems in unison. Although implementing FPGA is a fairly new concept to the 157 Design aspect, we felt that the challenge is rather vital to prove that we have actually learned and are capable of implementing all the knowledge garnered from the FPGA labs throughout the semester.

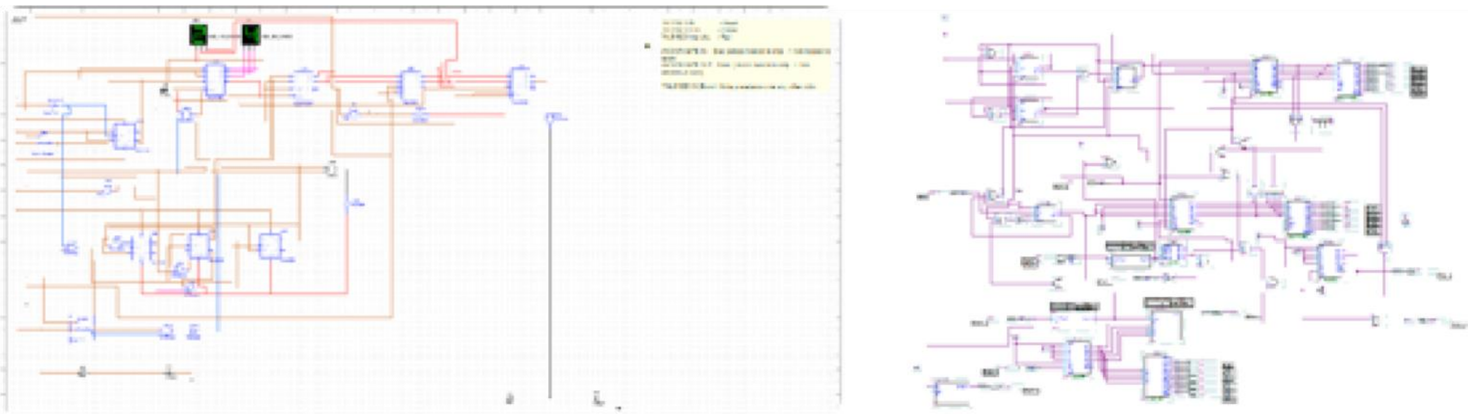
Technical Summary

Wiring the system on Multisim involved a lot of unnecessary probes and LED's for testing purposes. The inclusion of those components complicated the design further. We implemented the

74LS192D chips to successfully initiate the countdown and increment timers as per the specifications given. Then we moved on to work on the vent system and power control using VHDL programming structures provided by Professor Albert Knebel, Monroe Community College. Next, we managed the microwave door through a switch system implemented into the Altera board. Our lighting simply consists of an LED piece connected to the breadboard that lights up when the door of the system opens and also when the start switch is triggered. Essentially, we acknowledge that by working with Altera, we are stepping into a new dimension of digital logic implementation. However, with strong support from the faculty and an enormous interest in working with Computer-Aided Wiring, we feel that working with Multisim and Altera is a venture and experiment that turned out to be rather worthwhile.

Technical Discussion

Figure 1.0



Overall System

Our overall system was first laid out on Multisim then converted to Altera's Quartus II software for implementation. The system consists of timers, displays, a housed lighting logic, power systems, fans and logic for buttons and switches. All the major logic is housed in the Altera software system which is then implemented to the board. Figure 1.0 above describes the Multisim system on the left hand side and the Altera system on the right. Essentially, that is an image of various chips with different preset functions that we used to make the logic flow which then allows the different functions to operate. This logic is loaded into the board, where assigned switches and buttons perform various functions such as power increment, timer on, timer pause, clear button, servo motor control and so on.

The Altera board, in addition to housing the logic and code also works as an output base. It outputs the timer functionalities through the housed display system and the switches and buttons are operable by users preferences. The other forms of output are wired onto a traditional breadboard. Although the logic is loaded into the Altera board, the breadboard is connected to the Altera with a ribbon cable. The breadboard is also linked, through logic, with different chips and wires that go on to output the operation of the DC motor, servo motor and the LED. Conclusively, that summarizes the application of our entire system which is a smooth union of Multisim, Altera's Quartus II and traditional wiring techniques.

Subsystems

The subsystems are essentially the hands and legs of the entire project. Our approach to integrating the project was rather conventional. We worked on the subsystems individually on both Multisim and Altera Quartus II. Then we integrated them all together in Quartus II where we found a significant amount of bugs. The respective breakdown of each subsystems functionality is discussed below.

Timers

The timer is responsible for clocking the entire system. When the clock is set to a certain value, it will count down. At time value 0, the timer shuts the entire system down. The timer is controlled by a *74LS192D* chip which has the functionality of loading a value and counting down from there. For instance, if we input 10 seconds, the chip will load to 10 and countdown by 1 once the start switch is triggered. The Q outputs on this chip are connected to a 7 Sigma Display on the Altera board which portrays the clock countdown in ones and tens respectively. In our system, we used two *74LS192D* chips. This is simply because we are using the "borrow" function of the chip to display the ones and tens digit countdown in a respective manner.

Another major component that runs the timer is the 50 MHz internal signal clock which is divided down to 1 Hz for a one second countdown. As far as the loading of start time is concerned, we are implementing three J-K Flip-flops to synchronously flash from 0 to 6 in the tenths place. Additionally, we are using a J-K Flip-flop in toggle mode to flash the ones place from 0 to 5.

The last touch to the functionality of the timer is the VHDL software which divides our 50 MHz clock 24 times resulting in a clock output of 3 Hz. We then implement the *74LS92* chip to further divide the clock to 1Hz.

Display

The display operates on a rather straight-forward logic. Whatever is outputted from the *74LS192D* chip essentially goes into the display and shows the values that are dictated by the logic. However, before it can display the values, we used a *74LS47* chip to convert the values that came out of the *74LS192D* from BCD to 7-Sigma. After the conversion, the values are efficiently showcased on the display panel which is on the Altera board.

Lighting

Our system consists of two major lighting components. The first light is a representation of the counter light. We used a basic red LED component which was connected to the output of the Altera Board. On the software front, the Quartus contains a simple input/output connection which is linked to switch7 on the Altera Board.

The second lighting component is the oven light. This light was represented by a green LED component. The oven turns on either when the door is open or when the microwave is running. The light will be off when the microwave is not running, however, by opening the door even when the microwave is not functioning, the light will be triggered. This also uses a basic input/output format in terms of the software. However, we included a few AND gates and other logic supplements to allow the light to carry out it's respective functions.

Power Systems

The power systems consisted of a 12V motor, MOSFETS, a 74LS06 inverter chip, a comparator, a diode and a pull-down voltage of 9V. These components are wired on the breadboard. The input signal from the FPGA is from the power generator (PWM GEN) chip.

On the software front, the PWM GEN converts BCD code to control the duty cycle of the motor. A 12.2 KHz clock is going into the PWM GEN while the BCD contains 4 inputs that allow us to dictate the speed of the motor.

Motors (Fan and Servo)

Our project had two major motor components, *(power motor has already been discussed)* the fan and the servo motors. The fan or vent fan motor has a similar setup to the power motor. Basically, the vent motor is connected to an inverter chip, a comparator, a diode and a pull-down voltage of 9V. The Quartus wiring consists of a PWM GEN where the first two bits of BCD input are connected to Vcc and switch 8 is connected to the third input, while the last input is connected to Vdd ground. What this allows us to do is, when switch 8 is flipped to high the vent fan activates at 70% power to demonstrate a "high" setting. When switch 8 is flipped to low, it outputs 30% power, representing a "low" setting for the fan.

The servo motor is a simple representation of the carousel. The red and black wires coming out of the servo are respectively connected to power and ground on the breadboard. The white wire is connected to the ribbon cable. On the Quartus front, a 50 MHz clock comes into the chip and the enable is set to high, while the output goes into the ribbon cable. A vast majority of the software was designed off the provided VHDL code.

Sound Systems

Our project consists of one simple speaker that outputs a 2-seconds tone. The input comes from the Altera Board which connects to the trainer. The trainer then sounds the 2seconds tone. The tone was set at 30 KHz.

The Quartus wiring consists of a clock and a 74LS192 chip. We took a 1 Hz clock and put it through the 74LS192 counter so it would sound a 30 KHz frequency until the counter hits 2 counts. Once that happens, it sends a reset instruction the 74LS192 chip. Obviously, this only happens when the timer hits 0:00.

Buttons and Switches

Our Altera Board consisted of 3 buttons and 6 switches with their respective assignments. The first button, *BUTTON 0* is our 30-seconds preset button. Basically, when the button is triggered, it sets the clock to 30 seconds. This happens by using JK Flip-flop in toggle mode. When the button is pressed, it sends a HIGH to the Q of the flip-flop. The counter is then set to 0011.

BUTTON 1 is the power cycle button for the power motor. The button is used for power increment from 10%-100% gradually incrementing in 10’s by each click. This button’s functionality is based off two counters which cycle through 1-10 in BCD which is inputted to the PWM GEN for the power motor.

BUTTON 2 is the clock set. This button is meant to set the timer which cycles from 5-60 seconds with increments of 5 seconds per click. The clock functions on three J-K Flipflops running in synchronous fashion. There is also a fourth J-K Flip-flop that toggles between 0 and 5.

SWITCH 0 is meant for the door open/close. This controls the oven light in the sense that when the door opens up, the light inside is triggered and when the door is closed the light goes off. Based off Quartus, we the switch being inputted with logic which dictates the functions of the switch. There is no involvement of a chip or any such component for this particular function.

SWITCH 1 is the pause switch. This switch when turned to HIGH will stop the timer where it is and when switched to LOW will continue the timer from where it left off.

SWITCH 2 is the counter light. Based off simple logic circuitry, the light comes on when switch is triggered to HIGH and turns the light off when the switch goes LOW.

SWITCH 7 is the vent fan on off switch. Essentially, when the switch goes HIGH, the fan turns on and when the switch goes LOW it turns the vent fan off.

SWITCH 8 is the vent fan high low switch. Switch 7 simply turns the vent fan on or off. Switch 8, however, controls the power of the vent fan. If the switch is HIGH the power will go to 70% and when the switch is LOW power will go to 30%.

SWITCH 9 is the start/clear switch. When the switch goes high, the clock starts to countdown and the microwave starts. When the switch goes low, it clears the entire board out.

Overall, we feel that the technical applications of the project were well versed by our group. We managed to gather all the required data and logic flow we needed to go forth and proceeded in due fashion. The various subsystems integrated via the Altera Board, breadboards and Quartus gave us the opportunity to generate a fine integrated system that fulfilled the requirements of the final project in a just manner.

Project Development

Team Dynamics/Flow Chart

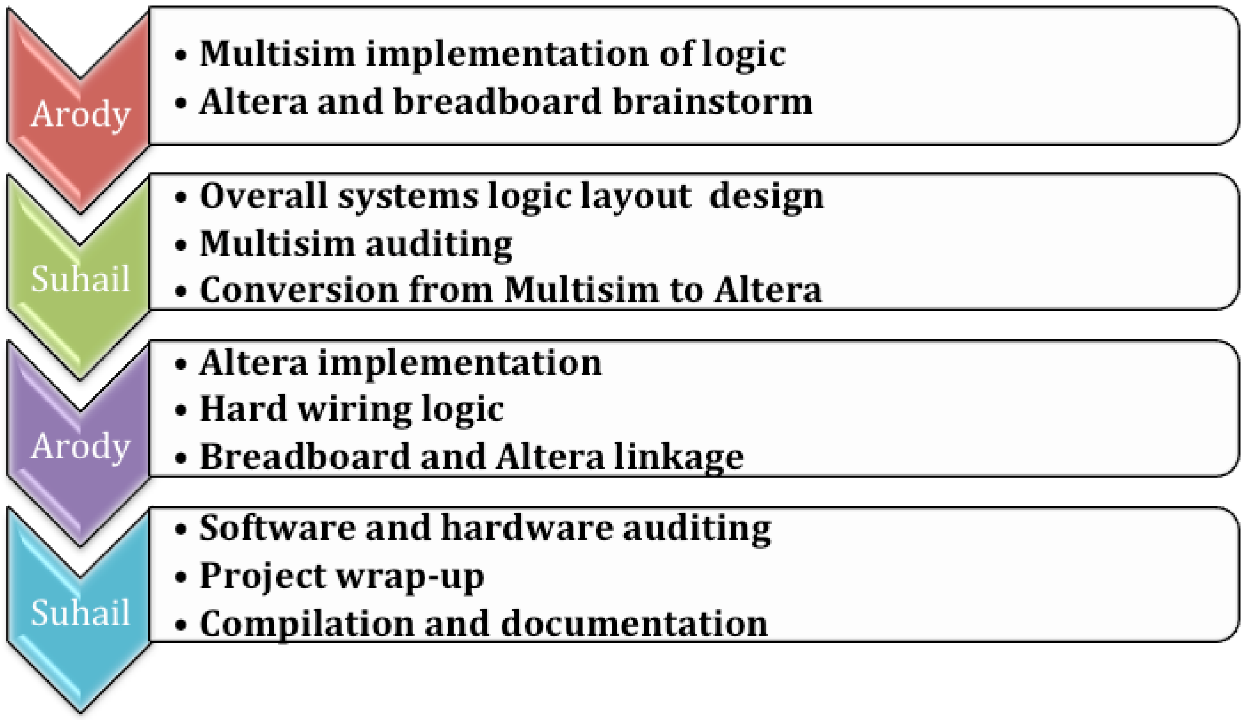
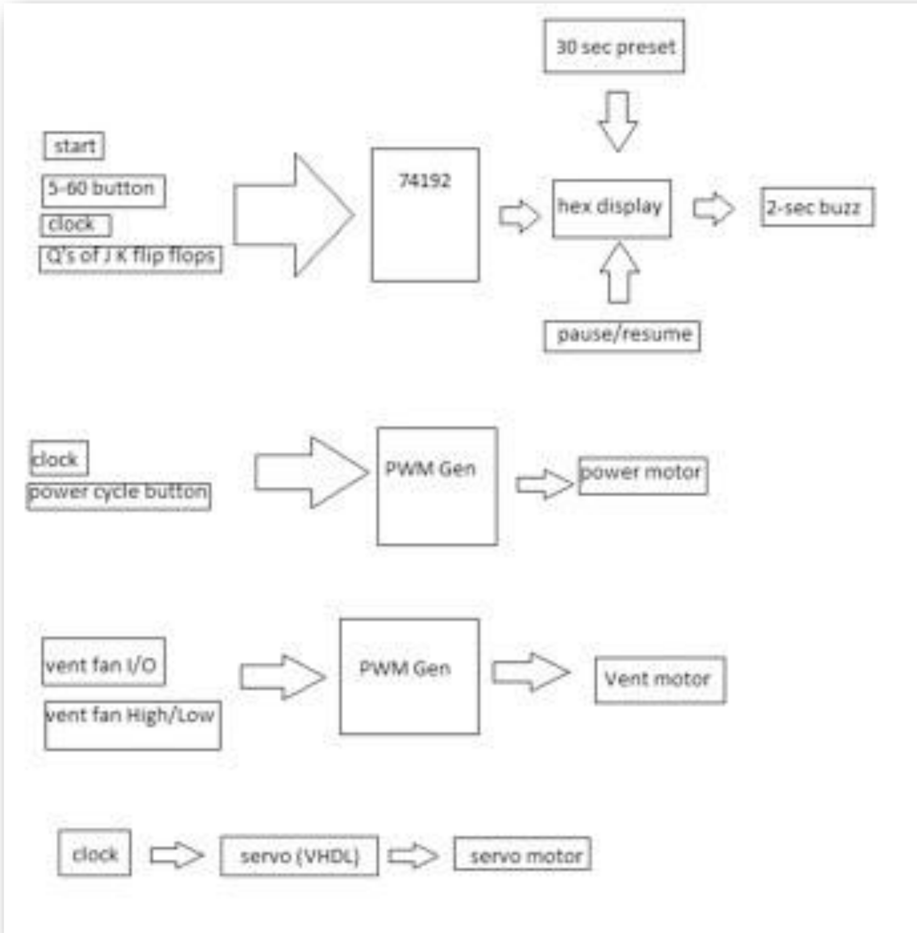


Figure 2.0

Block Diagrams and Logic Analysis

The image below is a simple block diagram describing all the major functions of our project in the Quartus software system. Basically, the block diagram breaks down and simplifies all the logic and different inputs/outputs so the viewer can easily comprehend what is going on in the circuit system. A detailed description with its respective functionalities can be found in the FPGA Analysis section of the report.

Figure 2.1



Cost Analysis

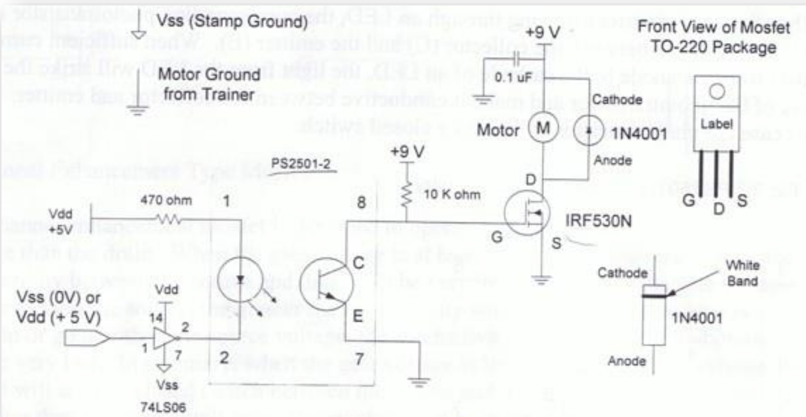
The cost analysis as follows is a compilation of parts we used to build our microwave system. The data table includes information such as the part name, it's description, the supplier, the quantity used, it's individual unit price and the total cost of each product. Our data table consists of highly intricate decimal places in terms of unit and total costs, that is because some of our suppliers are in China and currency conversion results in two place decimal values on most products.

Part Name	Part Description	Supplier	Quantity Used	Unit Price ($)	Total Cost ($)
Altera Board	**Main control board, heart of the project.**	**Altera**	1	US$129.99	US$129.99
Male A-B USB Cable	**Generic USB cable, generally used with printers**	**Amazon**	1	US$2.69	US$2.69

Ribbon Cable	Link cable between the Altera and Bread boards	BoHobby Shop	1	US$0.15	US$0.15
Breadboards	Board for making experimental model of a circuit	BoHobby Shop	2	US$8.99	US$17.98
Wire kit	General kit with various wire leg lengths	MCC Bookstore	1	US$14.75	US$14.75
Chips	Clip on systems with varied logic functions	BoHobby Shop	8	$1.00 (average)	US$8.00
LED's	Light emitting probes, usually used on breadboards	BoHobby Shop	4	US$0.50	US$2.00
Motors	9V DC Motor, Servo Motor, Lamp	BoHobby Shop	1 each	US$48.00	US$48.00
			Unit Total Cost:		Total Cost:
			$206.07		US$223.56

Sensor System Schematics and Logic Development

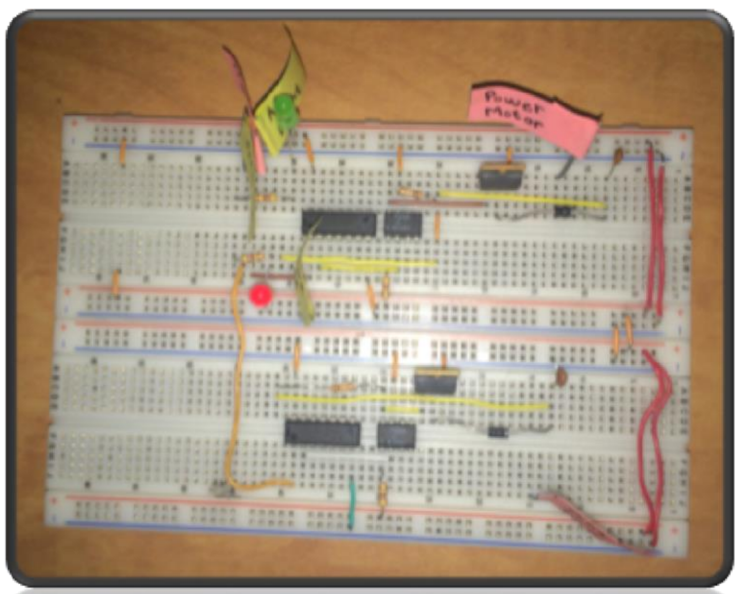
Figure 3.0



The two motor controls, (vent motor and power motor), are wired by the schematics below. The output of the PWM motor goes the input of the inverter.

Figure 3.1

The counter light (RED LED) is wired to the output of the ribbon cable same thing for the (GREEN LED) which is the oven light. The other wiring prospects include motors, speakers and the vent fan.

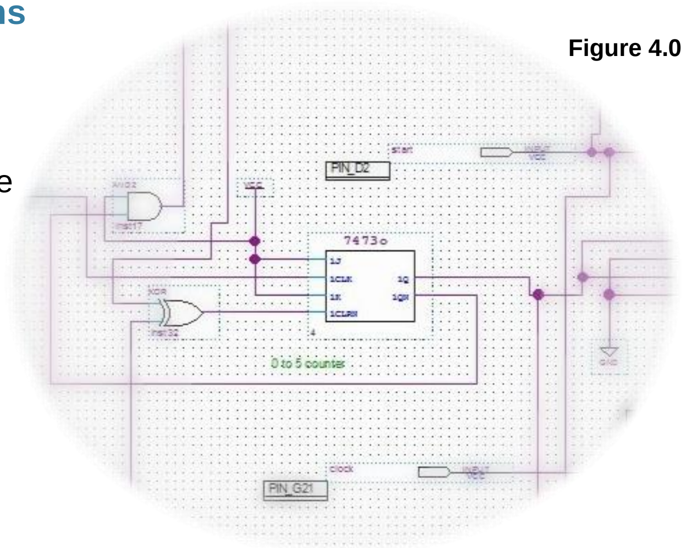


FPGA Analysis

The software aspect of the project was probably the most exciting aspect of the project considering all the members in the group are computer engineering majors. The next few pages will entail our ride through various nitty-gritty aspects of our FPGA wire-up showing specific components being wired up and their logic being rightfully justified. Additionally, we have a full array of our comments, schematic summaries and close-ups and port assignments.

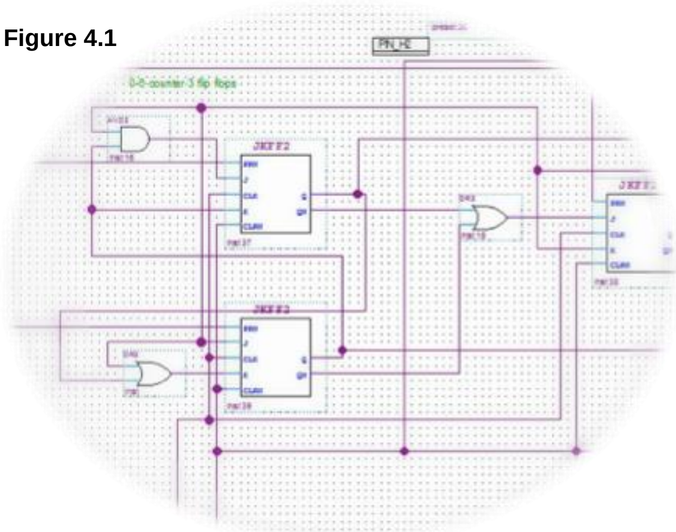
Schematics for Respective Subsystems

Figure 4.0



The 74LS73o chip described in *figure 4.0* represents the 0 to 5 counter. Essentially, the clock is a push button to enable the Q1 HIGH/LOW which allows the output values to flip from 5 to 0 and vice versa.

Figure 4.1



The 74LS73o chip described in *figure 4.1* represents the 0 to 6 counter. This clock is represented by a push button that allows count up/down on the tens digit display section.

Figure 4.2

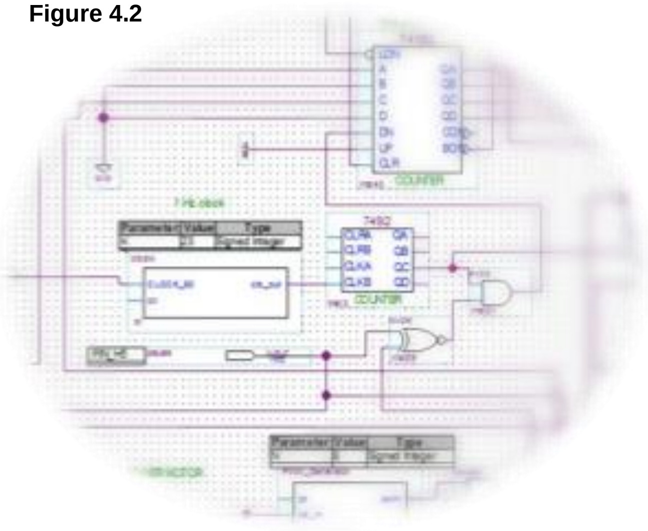
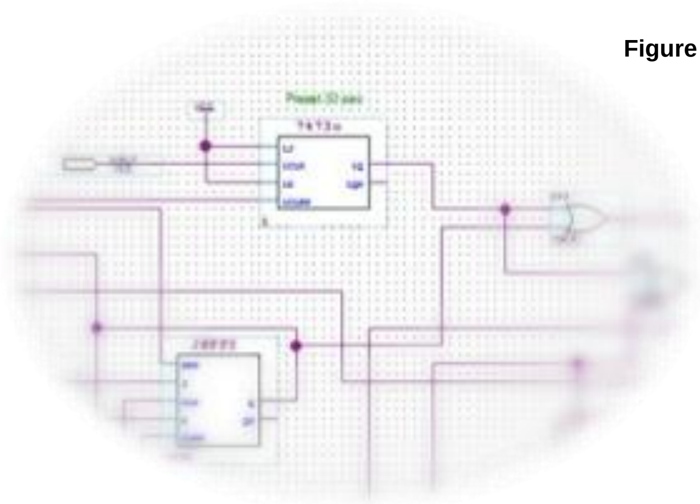


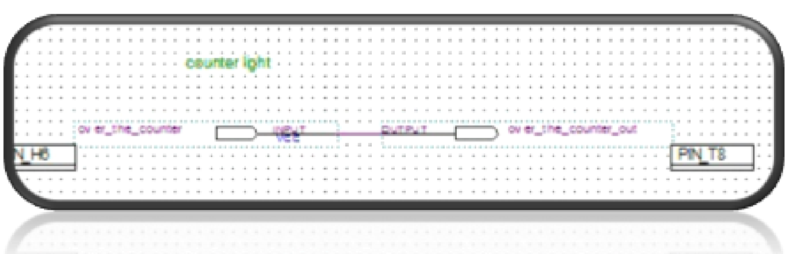
Figure 4.2 shows an input from clock divider which is 50MHz, then we used VHDL to divide the clock to 3 Hz. We then used MOD3 counter to divide to 1Hz which we also used for our speaker.

Figure 4.3



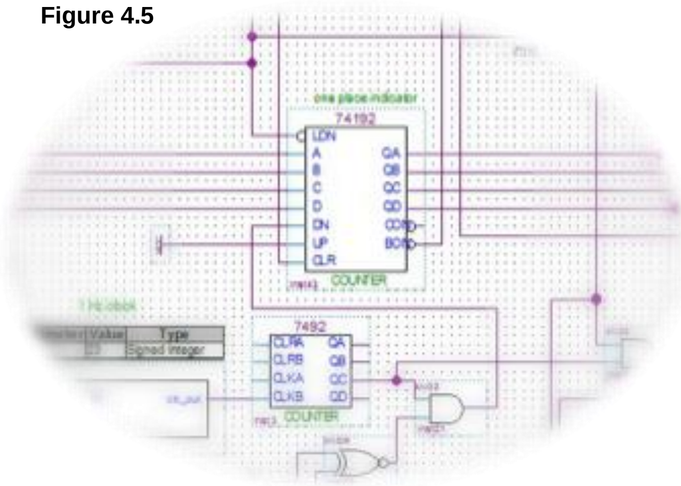
The 30 seconds preset button described in the figure on the right is rather similar to the 0-5 counter. As for the button to JK flipflop, when pressed goes to 1, when let go, goes to 0. These pulses are then sent to BCD input 0011 to display 30 seconds at the 10's place indicator.

Figure 4.4



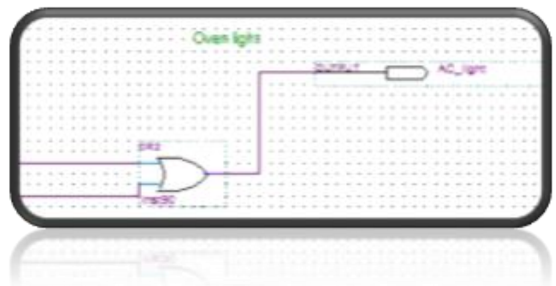
The light counter in the figure on the left is a simple input-output connection. The input is a switch while the output is a port on the ribbon cable to the LED.

Figure 4.5



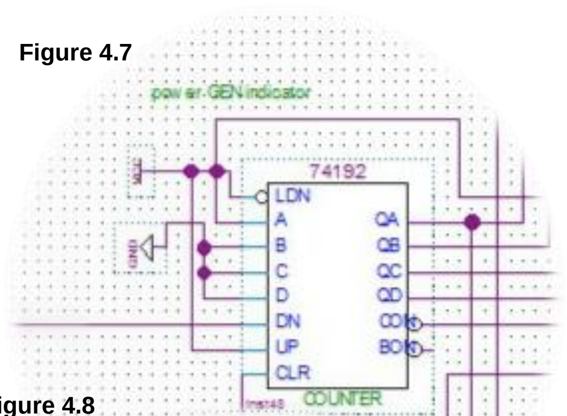
The image on the left is a ones place indicator. It is a 0-5 count that is wired up to the ones place for display of the countdown.

Figure 4.6



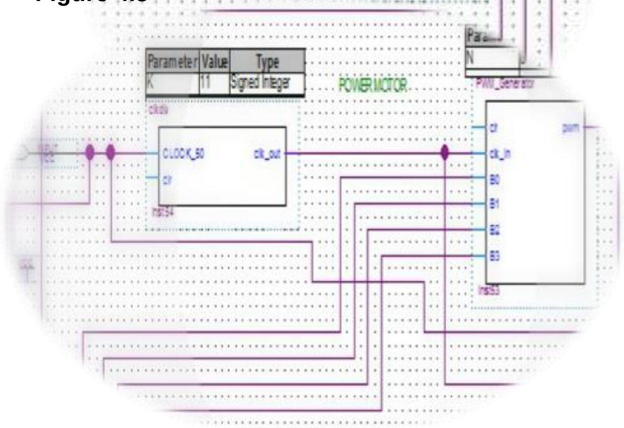
The image in *Figure 4.6* describes the oven light system. According to the wiring schematic, the light will be off when the microwave is not running. By opening the door, even when the microwave is not in motion, the light will turn on.

Figure 4.7



The figure on the left describes the PWM Generator that converts BCD code to control the duty cycle of the motor. Additionally, a 12.2 KHz clock is going into the PWM Generator while the BCD contains 4 inputs that allow us to dictate how the motor acts.

Figure 4.8



The power systems in the image on the left consists of a 12V motor, MOSFETS, a 74LS06 inverter chip, a comparator, a diode and a pull-down voltage of 9V. These components are wired on the breadboard. The input signal from the FPGA is from the power generator (PWM GEN) chip.

The image on the right is of a servo motor which is a simple representation of the carousel. On the Quartus front, a 50 MHz clock comes into the chip and the enable is set to high, while the output goes into the ribbon cable. A vast majority of the software was designed off the provided VHDL code.

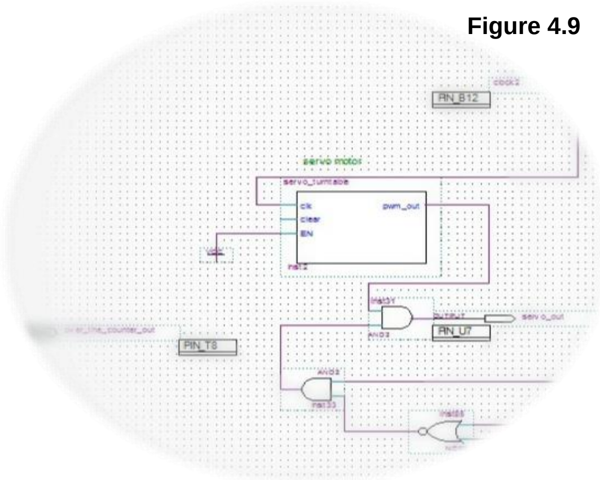
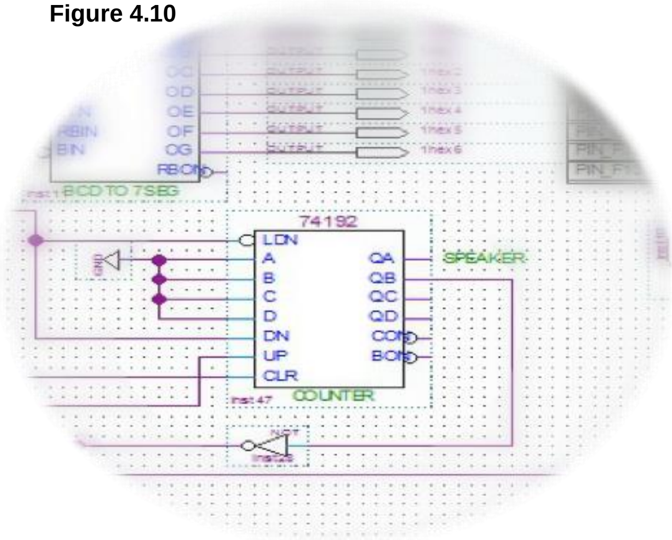
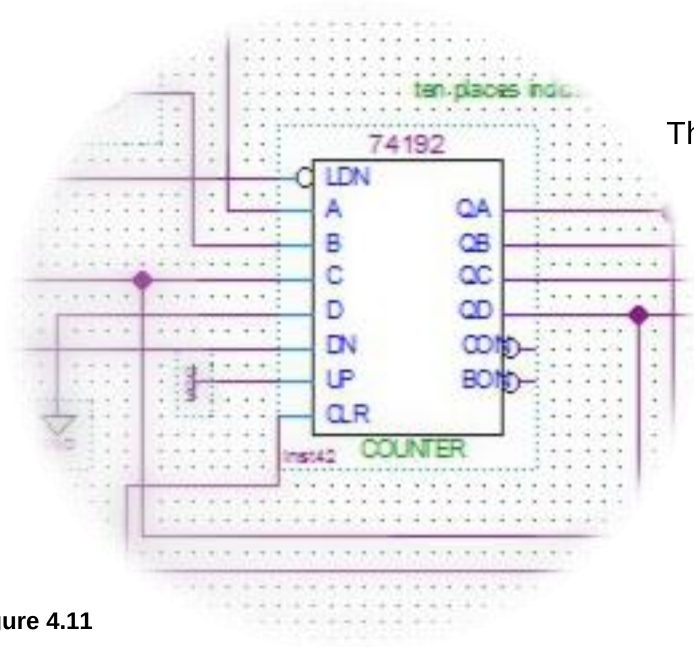


Figure 4.9

Figure 4.10



The image on the left shows a simple speaker that outputs a 2-seconds tone. The input comes from the Altera Board which connects to the trainer. The trainer then sounds the 2-seconds tone. The tone was set at 30 KHz.

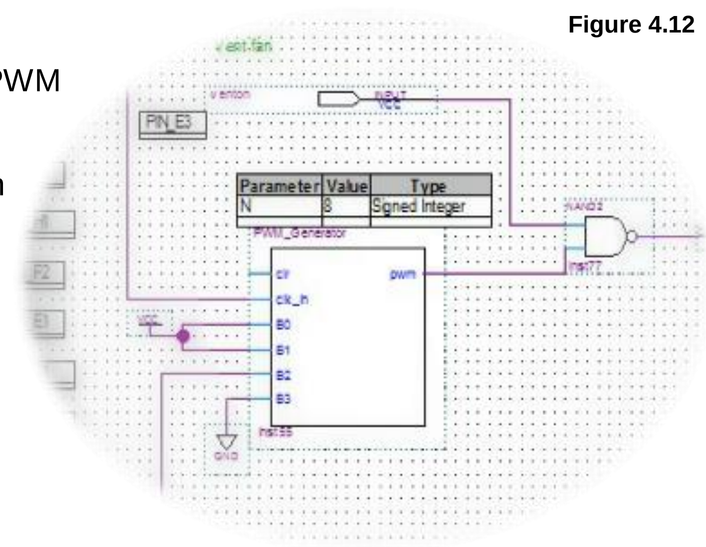


The image on the left describes a 74LS192 chip that consists of in's and out's that assist in the function of the display of the ten's place indicator.

Figure 4.11

Figure 4.12

The image on the right consists of a PWM GEN where the first two bits of BCD input are connected to Vcc and switch 8 is connected to the third input, while the last input is connected to Vdd ground. This figure represents the wiring system of the vent fan system.



Conclusively, the intricate addition of our Quartus system displays the skillfully crafted chips, gates and wires that add up to make a holistic compilation that results in the microwave system operation.

Conclusions

To wrap up, we feel that we achieved our goals through intense planning, definite dedication, limitless efforts and ultimate precision. These qualities combined, along with all the knowledge garnered over the semester allowed us to fulfill all the requirements of the project and in fact, go over the requirements. This is a great feat for both of us because this is the first time we have both actually worked with dedicated project timelines, workflow charts, task delegation and repeated auditing.

Our philosophy to succeeding in time constrained projects like this one is to use various task delegation and prioritization techniques like post-its, quick notes, assigning based on skills and prioritizing based on urgency of application to the project. These tools allowed us to produce output at a maximum efficiency which resulted in a rather successful outcome.

It is definite that we learned a whole lot from the course, not only in terms of Digital Electronics and Microcontrollers but also time management, efficient workflow and other team dynamics skills. We are both rather proud of the fact that we overcame all hurdles while working together on specific parts that gave us a lot of problem. We stuck to the idea that two brains are better than one and pulled through with it.

Lastly, both of us would like to thank both Professor Bertram Gamory and Professor Albert Knebel for setting forth a challenge that allowed us to learn and grow in ways we would never have imagined!

This page was intentionally left blank.