Digital System Design Lab

Lab #11 | Coffee Machine Statemachine Development

Author: Carlo Cortez

Lab Partner: Alex Nicolazzo

Instructor: Richard Maes

Date: November 13, 2025

Course: CPEN 430L 01

Electrical and Computer Engineering Department

Gonzaga University

Abstract

In this modern day and age, consumer demands for machines that produce food and beverages request that they be robust and reliable. This project aims to simulate a coffee maker machine of that caliber on a **field programmable gate array (FPGA)** board.The essence of the machine is based on a **finite state machine (FSM)**, in which the coffee selection, warnings, and progress of coffee production are all handled. Coffee selections are based on given recipes which include the coffee bin, type, and size and to be made through a start button. Other features implemented on the machine with sensors to track warnings are as follows: coffee bean bin level detection, water pressure and temperature sensing, non-dairy cream and chocolate powder tracking, and paper filter level tracking. All these features interface with the LCD on the board which include the selection menu UI, progress bars, and warning and error indications.

This system was developed using **System Verilog**, which is a **Hardware Description Language (HDL).** It has gone through extensive simulation to ensure proper signal timing and state transitions. The design was then developed and synthesized on a target FPGA board in which switches on the development board were utilized to simulate sensors, and the main progression and status of the machine were shown through the LCD on the development board. The results prove that this FPGA-based solution has provided a prominent foundation to this coffee machine, ensuring that sequential logic has been implemented correctly to avoid any metastability errors and that state machine controls work as intended and efficiently. This lab validates the process of FPGA utilization of developing complex state machine control systems for typical embedded applications showing the strong suits of high flexibility, timing control, and sequential logic.

Table of Contents

1. Introduction

1.1 Objectives

- Design and develop a proof of concept commercial coffee machine through an FPGA
- Demonstrate understanding of timing analysis, metastability, sequential logic implementation towards avoiding hazards and glitches, and finite state machine integration
- Verification through FPGA design flow through system architecture planning, simulation, synthesis, and optimization
- Analysis of efficiency, timing results, and responsiveness

1.2 Background

Many applications in a modern kitchen include embedded systems, where microcontrollers and dedicated digital logic dictate the operation of appliances ranging from microwaves to dishwashers. What makes these systems most notable for what they do are characterized for their reliable real-time control of physical processes with user input. In the case of applications needing high reliability, precise timing, and parallel thread execution, Field-Programmable Gate Arrays (FPGAs) provide an excellent alternative.

An FPGA is a semiconductor device that is enveloped in a matrix of configurable logic blocks (CLBs) and interconnects. These CLBs and interconnects can be programmed to synthesize and produce digital circuits. FPGAs present themselves as being ideal for control algorithms characterized and developed through state based applications, while also being able to execute tasks in parallel.

The essence of these control systems lie in Finite State Machines (FSMs), in which a FSM is a control system that specifies a finite amount of states that can also transition between one another. There are two primary types of FSMs, Moore and Mealy ***(Finish)***. States transition between each other through discrete events like switch flipping, timer expiration, and button presses.

This project dives into the viability of modeling digital control systems and logic through an FPGA for a commercial appliance, which in our case is our coffee maker. By implementing things through an FSM and utilizing the parallel task execution and management by an FPGA, we are able to successfully simulate, model, and synthesize a responsive, reliable, and testable digital control system to represent the core principles of digital system design that can be applied towards other embedded systems past this specific application.

2. Design Methodology

2.1 System Architecture

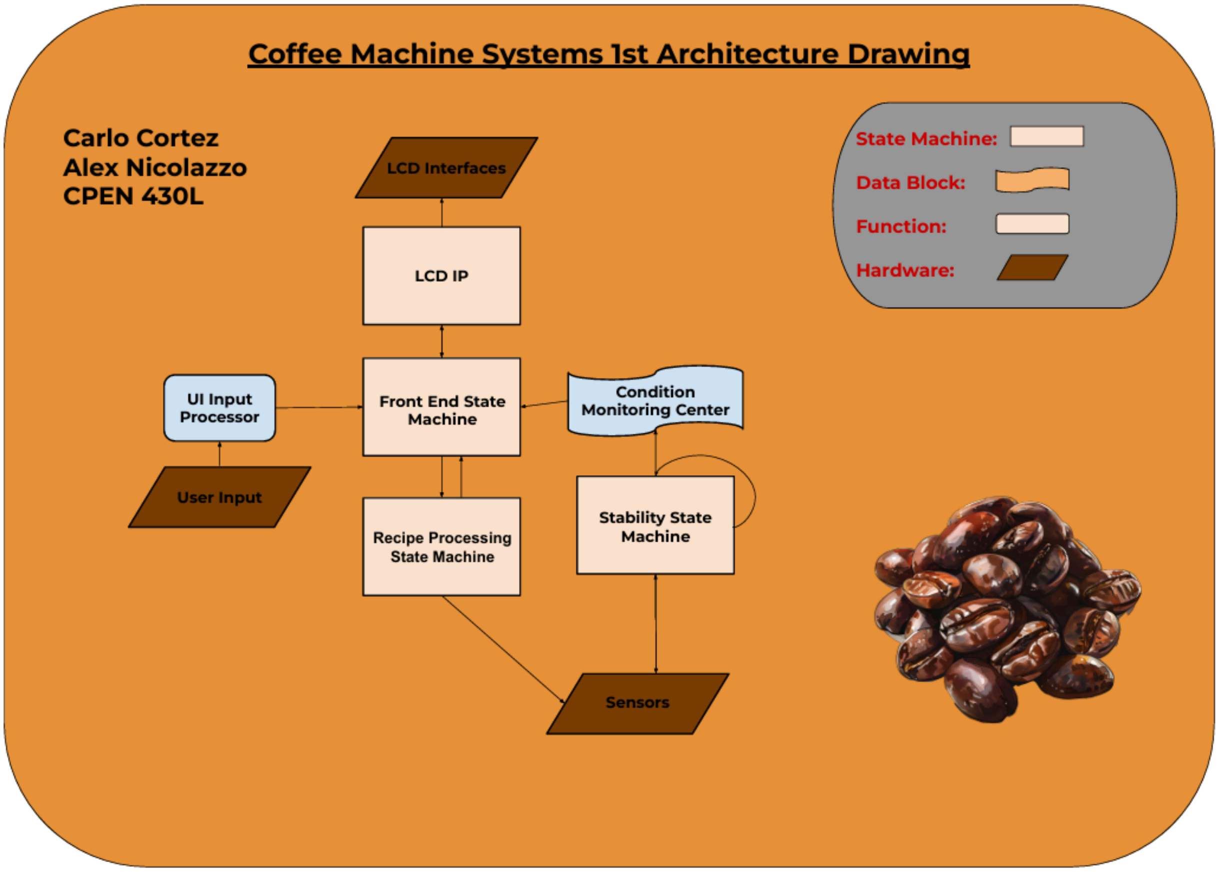
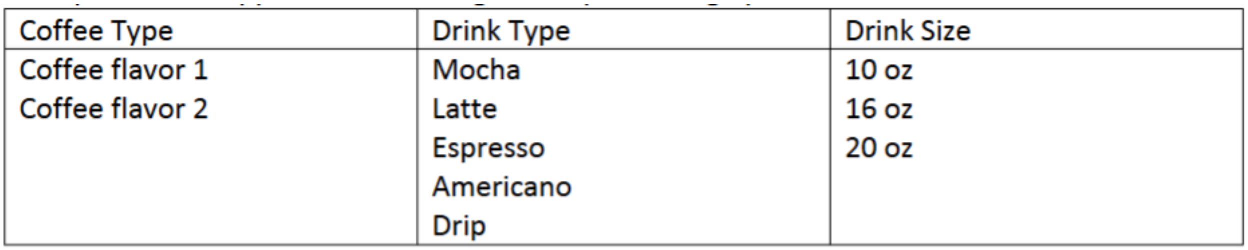


Figure 2.1a) Architecture drawing of the coffee machine generated by me and my lab partner during the first phase of the project.

The project first started out with basic system architecture planning and preparation. With the knowledge that we'd mainly use an LCD interface on the FPGA development board, we partitioned everything up into parts. Starting from the heart of the architecture, the front end state machine serves as the core to every other section of the design in which it processes the coffee orders and status updates through an FSM.

On the left of the front end state machine we have our user input, which includes any sensor inputs to simulate errors along with coffee order selection. To the right of the front end state machine, lies warning and error handling. The condition monitoring center contains the data for any warnings or errors that are present which are indicated and given by the stability state machine which constantly checks to see errors present in the system. The stability state machine is dependent on the sensors which are also hooked to the recipe processing machine, which actually goes through the process of making an order for the customer. The progress of the recipe and any warnings and errors thrown are fed back to the front end state machine for processing. Lastly, the LCD IP is the main component that interfaces with the front end state machine to handle these errors and warnings, UI interface for coffee selection, and recipe progress for the physical LCD display. This ties together all the other layers and provides a solution to give real-time response to the user about the coffee maker during their use of it.

2.2 Component Specification



Coffee Type	Drink Type	Drink Size
Coffee flavor 1 Coffee flavor 2	Mocha Latte Espresso Americano Drip	10 oz 16 oz 20 oz

Figure 2.2a) Coffee drink specification that includes types of coffee, drink type, and drink size.

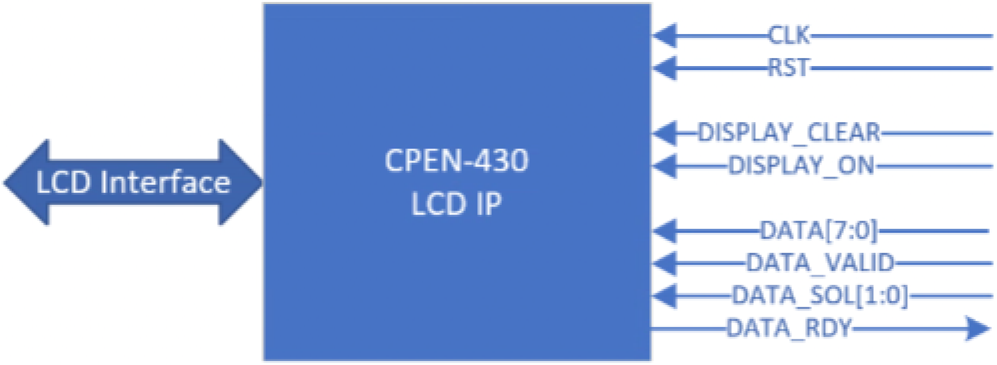
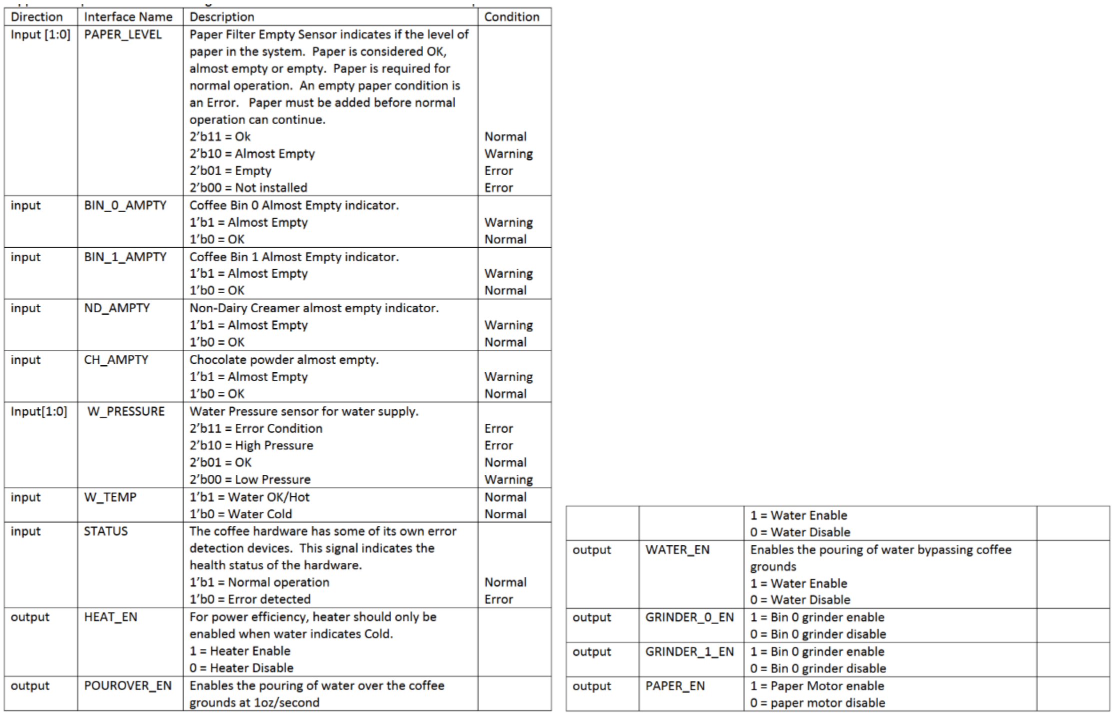


Figure 2.2b) Basic LCD IP configuration providing a generated clock, reset, display command signals, and data line signals.



Direction	Interface Name	Description	Condition
Input [1:0]	PAPER_LEVEL	Paper Filter Empty Sensor indicates if the level of paper in the system. Paper is considered OK, almost empty or empty. Paper is required for normal operation. An empty paper condition is an Error. Paper must be added before normal operation can continue. 2'b11 = Ok 2'b10 = Almost Empty 2'b01 = Empty 2'b00 = Not installed	Normal Warning Error Error
input	BIN_0_AMPTY	Coffee Bin 0 Almost Empty indicator. 1'b1 = Almost Empty 1'b0 = OK	Warning Normal
input	BIN_1_AMPTY	Coffee Bin 1 Almost Empty indicator. 1'b1 = Almost Empty 1'b0 = OK	Warning Normal
input	ND_AMPTY	Non-Dairy Creamer almost empty indicator. 1'b1 = Almost Empty 1'b0 = OK	Warning Normal
input	CH_AMPTY	Chocolate powder almost empty. 1'b1 = Almost Empty 1'b0 = OK	Warning Normal
Input[1:0]	W_PRESSURE	Water Pressure sensor for water supply. 2'b11 = Error Condition 2'b10 = High Pressure 2'b01 = OK 2'b00 = Low Pressure	Error Error Normal Warning
input	W_TEMP	1'b1 = Water OK/Hot 1'b0 = Water Cold	Normal Normal
input	STATUS	The coffee hardware has some of its own error detection devices. This signal indicates the health status of the hardware. 1'b1 = Normal operation 1'b0 = Error detected	Normal Error
output	HEAT_EN	For power efficiency, heater should only be enabled when water indicates Cold. 1 = Heater Enable 0 = Heater Disable	
output	POUROVER_EN	Enables the pouring of water over the coffee grounds at 1oz/second 1 = Water Enable 0 = Water Disable	
output	WATER_EN	Enables the pouring of water bypassing coffee grounds 1 = Water Enable 0 = Water Disable	
output	GRINDER_0_EN	1 = Bin 0 grinder enable 0 = Bin 0 grinder disable	
output	GRINDER_1_EN	1 = Bin 0 grinder enable 0 = Bin 0 grinder disable	
output	PAPER_EN	1 = Paper Motor enable 0 = paper motor disable	

Figure 2.2c) Table of sensor inputs and outputs. Designation of normal, warning, and error status.

Any errors or warnings present are collected into a warning mask which is periodically displayed to the LCD at 1-second when no blocking errors are present (errors preventing recipe progression or making such as lack of ingredients, and so forth; any errors that are listed below). Multiple errors are handled with priority, with errors taking main priority, and then the type of errors that the system throws from hardware status errors, water pressure, ingredient inventory, and paper filter level status respectively.

2.3 State Machine Selection and Processing Architecture

2.3.1 Recipe Table and Lookup Integration

Recipes are given through **"cmach_recipes.svh"** with the respective **"cmach_recp.sv"** file containing all the given ingredients, timing phases, and information regarding the coffee recipes. Each entry is packed (**"coffee_recipe_t"**) containing this information which gets unpacked in the actual system into its individual parts.

- r_load_filter (filter / paper load phase)
- r_grinder time (seconds)
- r_cocoa_time (seconds)
- r_pour_time (seconds)
- r_hot_water_time (seconds)

- r_add_creamer (addition of creamer during brew if toggled)
- _unused_HP (additional packed bit that generally is as stated unused)

With the info provided through the packed record, timing and ingredient information were tested as followed to meet the lab requirements and constraints.

2.3.2 Top-Level State Machine

The coffee production engine consists of three main states as shown:

- SELECT (S_SELECT): Users are able to cycle through their options of coffee flavor, type, and size which are shown in real-time through the LCD.
- WAIT (S_WAIT): After the start, the engine prepares the temperature and waits for it to be completed (W_TEMP)
- BREW (S_BREW): This is a phase based cycle that gets executed using the timing and ingredients provided of the recipe selected.

With the given processes and sensors in **2.3.1**, LEDs on the FPGA board are lit up to indicate which procedure is being executed in the current time.

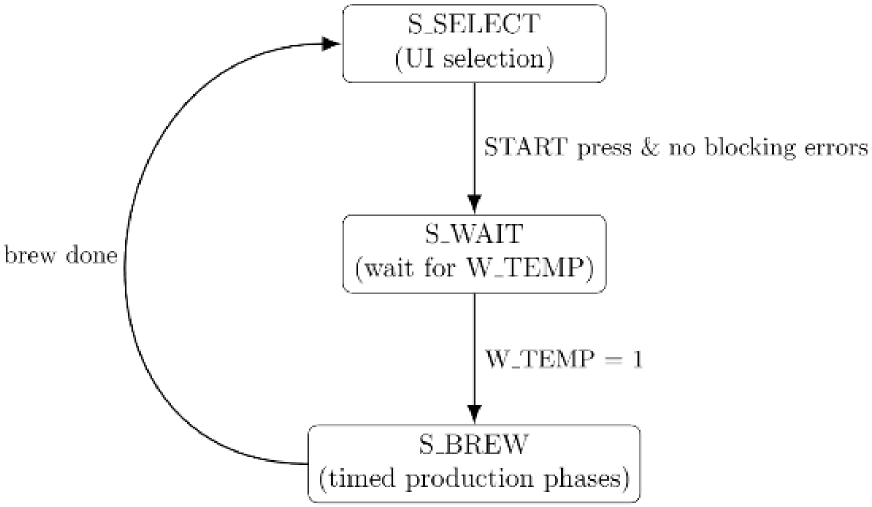
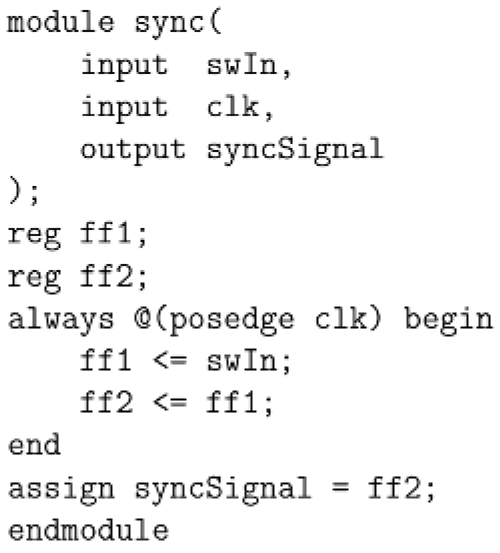


Figure 2.3.1 a) Top level state machine handling the coffee making process.

3. Implementation Results

3.1 Metastability Factorization and Debouncing



```
module sync(
    input  swIn,
    input  clk,
    output syncSignal
);
reg ff1;
reg ff2;
always @(posedge clk) begin
    ff1 <= swIn;
    ff2 <= ff1;
end
assign syncSignal = ff2;
endmodule
```

Figure 3.1a) Sync module to avoid and prevent metastability issues.

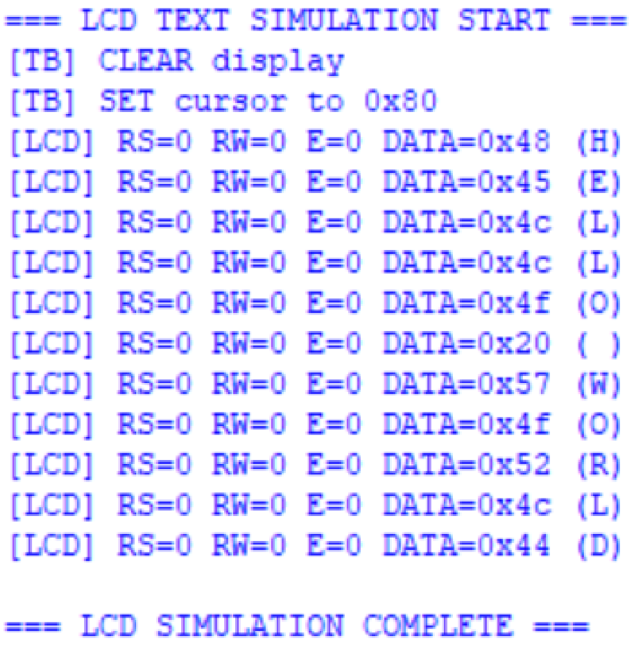
Any asynchronous UI inputs through the board's switches and buttons are synchronized using this module to avoid any metastability issues for clear and functional data output. Debounce for UI inputs as well are also implemented to ensure clean and accurately represented UI inputs for the user.

3.3 Hardware Specifications

FPGA Development Board: Altera DE2-115
FPGA Programmable Chip: EP4CE115F29C7
LCD: HD44780

4. Results and Analysis

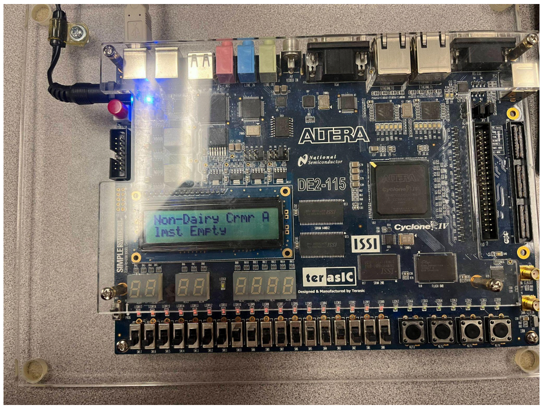
4.1 Simulation Results

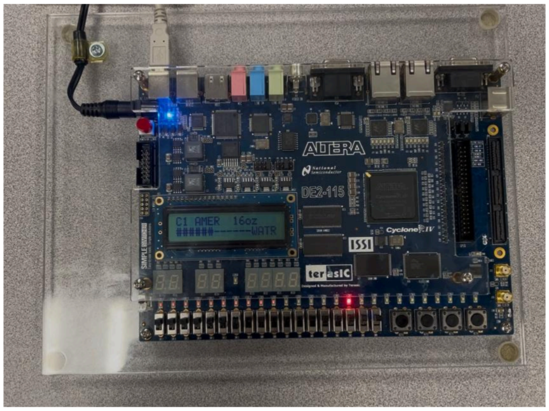


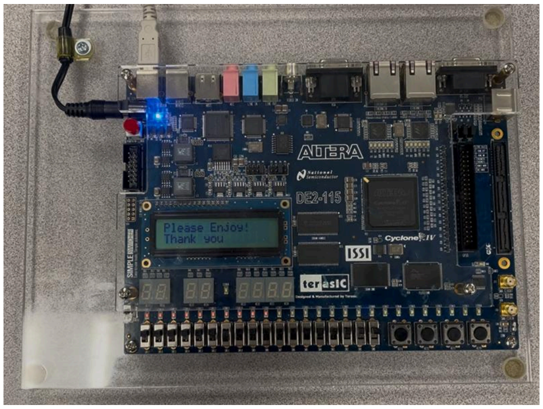
```
=== LCD TEXT SIMULATION START ===
[TB] CLEAR display
[TB] SET cursor to 0x80
[LCD] RS=0 RW=0 E=0 DATA=0x48 (H)
[LCD] RS=0 RW=0 E=0 DATA=0x45 (E)
[LCD] RS=0 RW=0 E=0 DATA=0x4c (L)
[LCD] RS=0 RW=0 E=0 DATA=0x4c (L)
[LCD] RS=0 RW=0 E=0 DATA=0x4f (O)
[LCD] RS=0 RW=0 E=0 DATA=0x20 ( )
[LCD] RS=0 RW=0 E=0 DATA=0x57 (W)
[LCD] RS=0 RW=0 E=0 DATA=0x4f (O)
[LCD] RS=0 RW=0 E=0 DATA=0x52 (R)
[LCD] RS=0 RW=0 E=0 DATA=0x4c (L)
[LCD] RS=0 RW=0 E=0 DATA=0x44 (D)

=== LCD SIMULATION COMPLETE ===
```

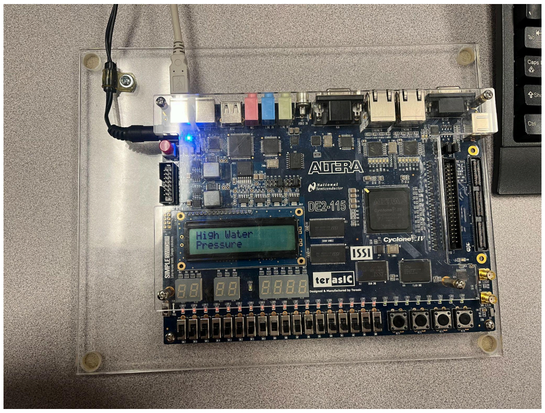
Figure 4.1a) Simulation transcript results showing that data is correctly being cleared and sent on the LCD according to the datasheet.

4.2 Sample LCD Outputs

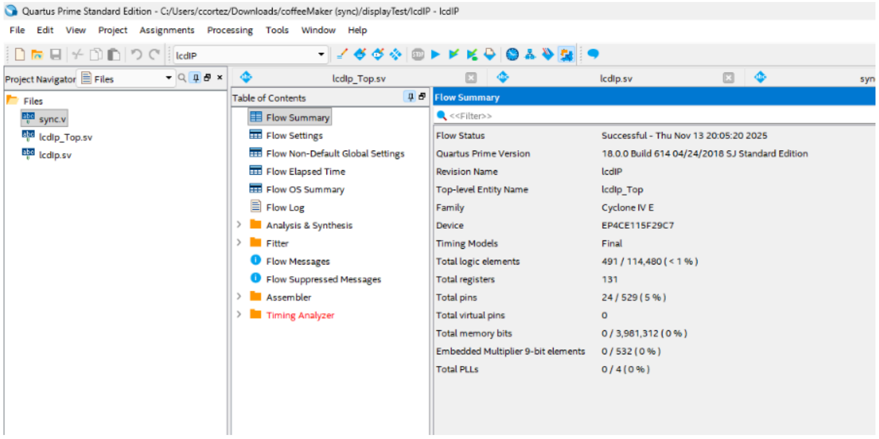








4.3 Flow Summary



Flow Status	Successful - Thu Nov 13 20:05:20 2025
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Standard Edition
Revision Name	lcdIP
Top-level Entity Name	lcdlp_Top
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	491 / 114,480 (< 1 %)
Total registers	131
Total pins	24 / 529 (5 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

5. Conclusion

Throughout the design process we have successfully achieved a proper functioning model of our coffee machine through FPGA development on the Altera DE2-115 board. Through extensive simulation testing, verification, optimization, and synthesis we have constructed so far a fully functional representation of a coffee machine with a selective menu UI, interfaceable sensors to simulate warnings or errors, and state machine implementation to process coffee recipes . The results and theory show so far that it is quite viable to use FPGA development to construct and model an embedded application or digital control system, and this model of the coffee maker shows proof of that.

6. References

Hitachi, Ltd. “HD47780U (LCD-II) Dot Matrix Liquid Crystal Display Controller/Driver”. Datasheet, Hitachi, Ltd. 1999

Terasic Technologies. “DE2-115 User Manual”. Terasic Technologies Inc.