Dryer Microcontroller Project

Phase 2

Benjamin Yeffeth bxy110230

3/1/2022

Introduction

This report contains the progress on the clothes dryer microcontroller since the Phase 1 report, delivered on 2/15 of this year.

This report will contain truncated truth tables, and waveforms from ModelSim sufficient to determine that the intended functionality is being produced by both the behavioral and synthesized Verilog code. However, in the interest of brevity, this report will not directly contain the complete code used to simulate these designs. The behavioral code is well-commented, and all code (including the test bench and synthesized RTL code) will be made available for review—packaged along with this report.

What follows is a brief summary of the previous instantiation of the design.

Review of Previous Iteration

The first iteration was purely instantiated as a block diagram and behavioral Verilog code. The device then had 3 primary inputs: a dial with four selections and “Start” and “Cancel” buttons. The “Start” and “Cancel” buttons are referred to by the terms RST, and OFF respectively. The dial, having four selections, was binary encoded to two bits. In total, there were four bits of input. There were no timekeeping mechanisms.

In the Phase 1 report, there were only four possible states. These states were termed _OFF, _STD, _TWL, and _DEL in the behavioral Verilog.

The microcontroller had only two outputs. A single bit motor output which gates voltage to the motor of the device, and a 2-bit heater output, which determined whether the heater is off, low, or high.

A quote from the Phase 1 report: “this microcontroller has been designed [with] economy and simplicity in mind, and in its current form has very few features and settings. As such, future iterations of this project present many opportunities for improvement and addition of features.”

Modified FSM Description

In its current iteration, the capabilities of the device have been substantially expanded.

There are 5 primary inputs: 3 buttons labeled “Start”, “Cancel”, and “Reset”, along with 2 dials, “Main” and “Heat”. The main dial, which is now a 4-bit input, has 10 valid input combinations, which are labeled “Standard”, “Towels”, “Delicates”, “Touch Up”, “90”, “75”, “60”, “45”, “30”, and “15”. The named (automatic) cycles are ‘one-hot’ encoded, and the numbered (manual) cycles are assigned to hexadecimal values ‘A’ through ‘F’, like so:

4’h1	–	Standard
4’h2	–	Towels
4’h4	–	Delicates
4’h8	–	Touch Up
4’hA	–	15
4’hB	–	30
4’hC	–	45

4’hD – 60
4’hE – 75
4’hF – 90

The heat dial is a 2-bit input with 4 possible input combinations, “OFF”, “LOW”, “MED”, “HIGH”.

2’b00 – OFF
2’b01 – LOW
2’b10 – MED
2’b11 – HIGH

The state register is a single 8-bit byte, and is organized in the following fashion, along with each state’s associated outputs (where uppercase ‘X’ refers to a ‘don’t care’):

Hex Value		State Title		MTR	HTR
8’h1X	–	Standard	→	1	01
8’h2X	–	Towels	→	1	11
8’h4X	–	Delicates	→	1	00
8’h8X	–	Touch Up	→	1	01
8’hAX	–	Manual (high heat)	→	1	11
8’hBX	–	Manual (med heat)	→	1	10
8’hCX	–	Manual (low heat)	→	1	01
8’hDX	–	Manual (heat off)	→	1	00

For example, the states 0001’0000 – 0001’1111 represent the standard cycle states, the states 1010’XXXX represent the manual high heat cycle states, and so on.

The outputs of the machine remain largely unchanged, with a 1-bit motor and 2-bit heater output. However, the heater output has an additional legal state ‘10’ which represents medium heat.

The most influential change implemented in this phase was the addition of a timekeeping mechanism. The testbench timescale is set to 10us/10us, with a clock toggle every time unit. This implies a 20us period between clock rising edges, corresponding to a 50kHz clock frequency. Every clock cycle, a 16-bit wide register is incremented. When this counter reaches 50000, we should estimate that one second has passed. At this point, another counter is incremented and the original counter reset. We will refer to this other counter as the “sec_counter”.

When the sec_counter gets to a *certain value*, the state of the dryer is decremented to indicate that a single ‘period’ has occurred. This *certain value* is the minimum cycle length, and we will call it a “cycle period”.

Intended Behavior

When the main dial is turned to select an option and the 'ON' button is pressed, that cycle activates. Each possible cycle has a certain number of periods it will run for, after which the state will return to IDLE automatically.

To describe the timescale system: in a traditional dryer, a reasonable cycle period would be around 15 minutes, and this cycle would repeat several times to dry a single load of laundry. In the testing of this design, the cycle period was set to 3 seconds, to facilitate adequate testing while avoiding cumbersome timescales in our simulation software.

The automatic cycles have preset output settings and duration, and include "standard", "towels", "delicates", and "touch up". A "standard" cycle runs the motor for 6 periods at low heat. "Towels" turns heat to high and motor on for a duration of 6 periods. "Delicates" runs the motor for 3 periods with no heat. "Touch up" runs motor with low heat for only two periods.

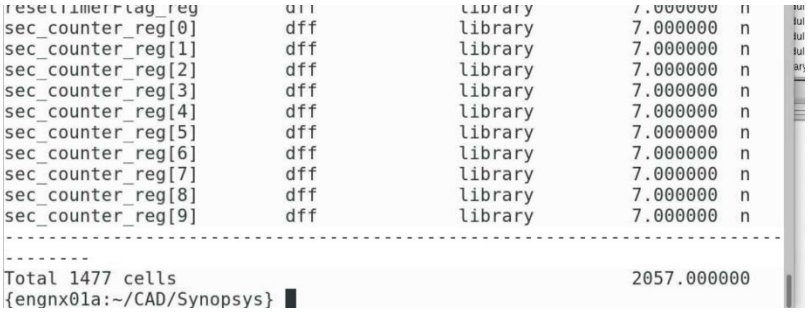
For example, turning the dial to standard and pressing 'ON' will switch the state to 8'h15. Every period, the state is decremented, until it reaches 8'h10. Once a period has elapsed during state 8'h10, the state reverts to 8'h00, the IDLE state. Using testing times, standard cycle runs for a total of 18 seconds. Using production times (with each period lasting 15 minutes), the standard cycle runs for a total of 90 minutes.

The manual cycles are reached by turning the dial to one of the numbered settings (ie: '15', '45') and setting the heat dial to the desired setting. The motor will run for $n_{periods} = {}^{n_{dial}}/_{15}$ at the selected heat setting. For example, turning the dial to '45' and the heat dial to 'HIGH' will run the motor for three periods with high heat.

Netlist Discussion

After composing the behavioral Verilog and verifying the correctness of the intended behavior, a provided library of standard cells was used to synthesize a mapped Verilog file. This was appended with another provided file, 'header.v' which contained behavioral descriptions (in the form of modules) of the mapped cells.

On the right: (Figure 1.1) The last few lines of text generated by Design Vision's Cell Report. In total, this design is composed of 1477 cells, 39 of which are D Flip-Flops.



Behavioral Analysis

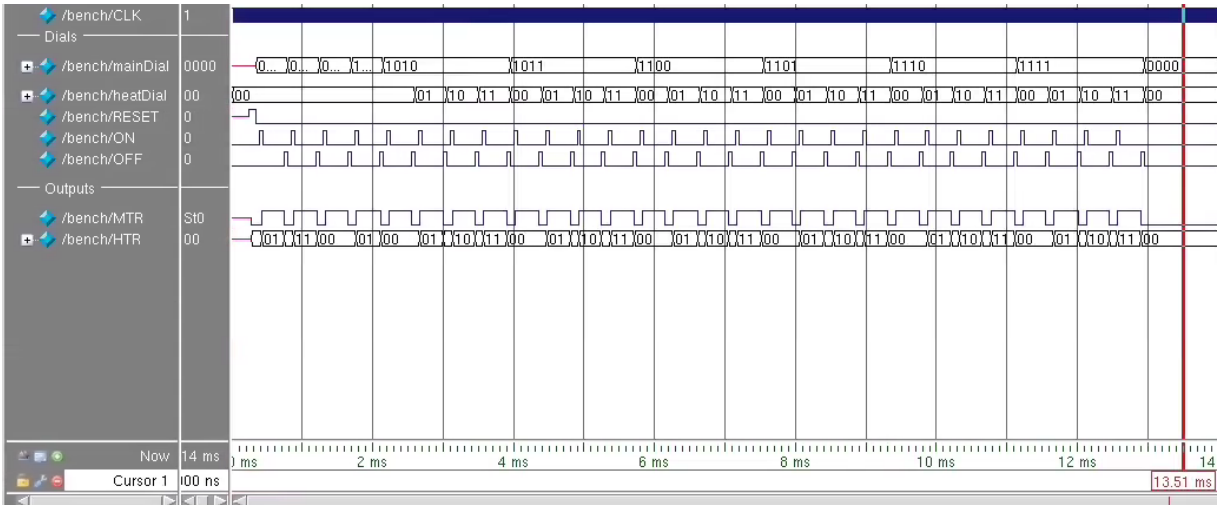
This synthesized code was simulated, to ensure that it behaved in the same way as the original behavioral code. Three tests were used: the “On/Off” test, the “Automatic Cycles Timer” test, and the “Manual Cycles Timer” test.

The On/Off Test

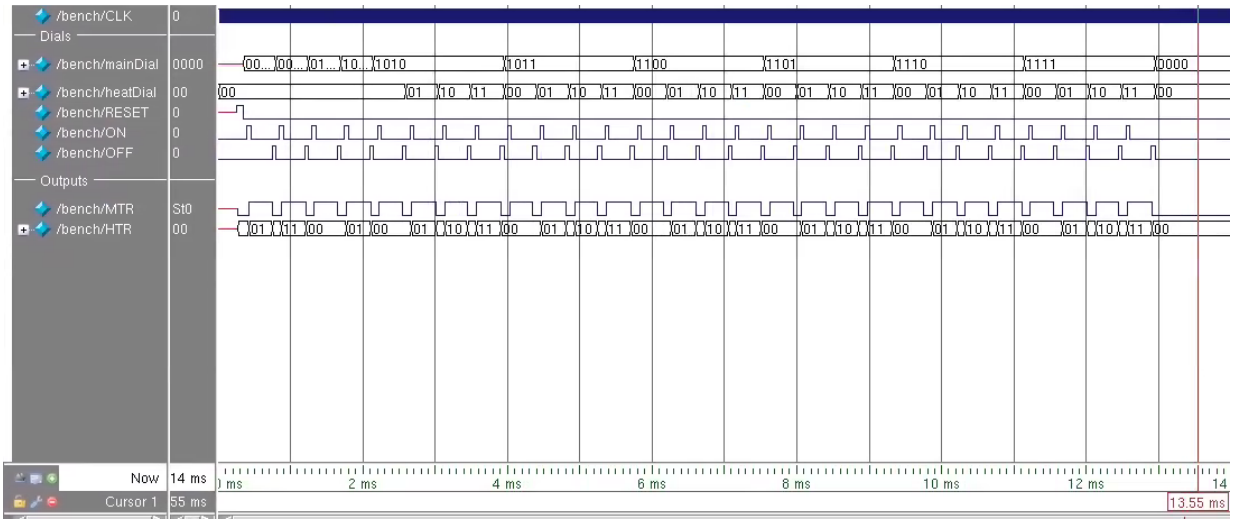
At the heart of the on/off test is the idea that the IDLE state should be directly accessible from each other state, and each state should be directly accessible from the IDLE state provided the dials are turned to the correct positions. Additionally, each of these states should produce the expected output combinations.

First, each of the “one-hot” automatic dryer cycles is accessed **briefly**, then shut off using the ‘OFF’ button. Next, each of the manual times is accessed, for each of the possible heater settings.

This test is active for approximately 13 ms.



Above: (Figure 1.2) Waveform diagram from ModelSim showing the results of the “On/Off” test performed on the behavioral Verilog module. First, observing mainDial, we can see that all four ‘one-hot’ states are accessed, the ‘ON’ button is pushed followed by the ‘OFF’ button for each. Just after the 2ms mark, we begin accessing the manual states, starting with ‘A’. It is at this time that the heatDial begins to be adjusted, in order to reach all possible states.



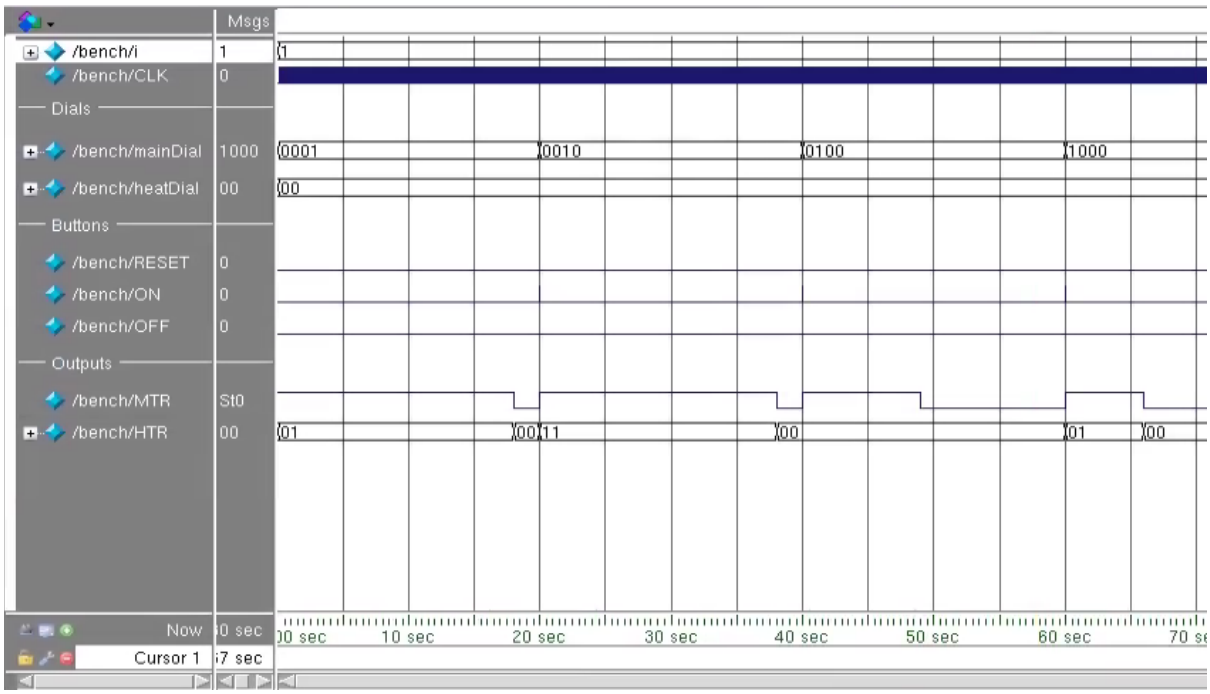
Above: (Figure 1.3) Waveform diagram from ModelSim displaying the results of the "On/Off" test as performed on the Verilog module synthesized using Design Vision. The same series of inputs is delivered to the module, and the same set of outputs result.

The Automatic Cycle Timer Test

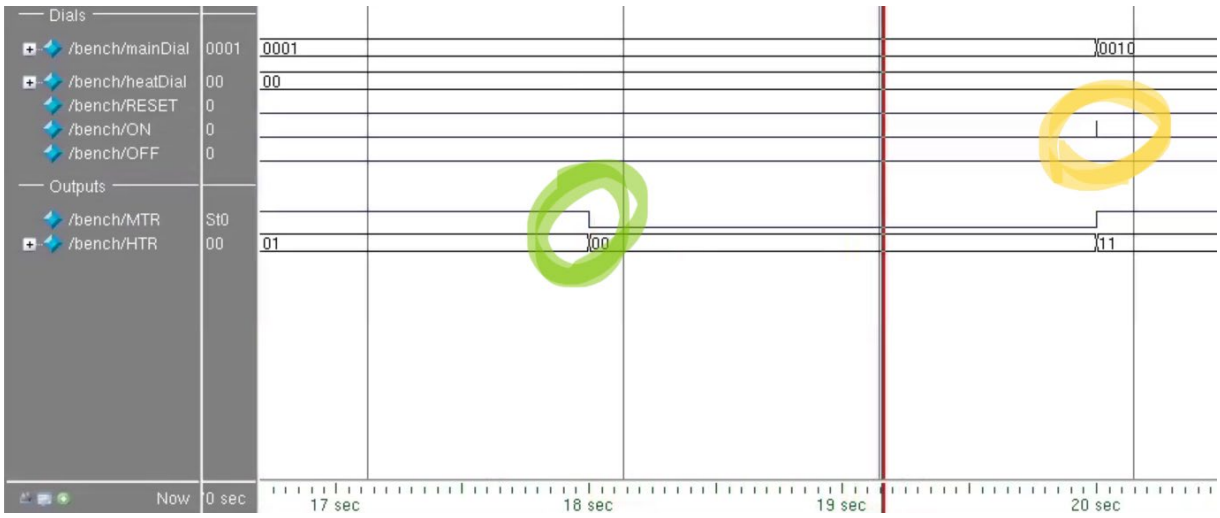
While the previous test did not utilize the timing mechanism, the next two tests focus on that mechanism primarily.

In the automatic cycle timer test, we access each of the 'one-hot' automatic cycle dial settings, turn them on, and wait for them to turn off **without using the 'OFF' button**. First, the main dial is set to 'b0001 for the standard cycle, which runs for 6 periods. Second, we turn the dial to 'b0010 (equivalent to bit-shift left) for the towel cycle. This cycle also runs for 6 periods. Next, we bit-shift again, setting the dial to 'b0100 for the delicates cycle. This cycle only runs for 3 periods. Last, setting the dial to 'b1000 selects the touch up cycle, which runs for two periods.

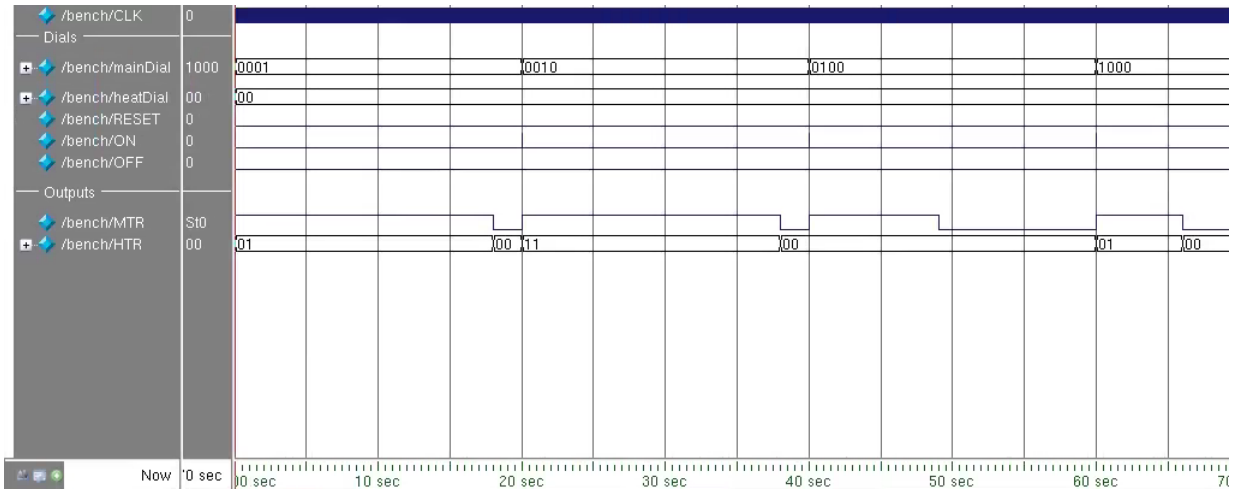
This test remains active for approximately 67 seconds using the testing period duration of 3 seconds.



Above: (Figure 1.4) Waveform diagram from ModelSim displaying the results of the automatic cycle timer test on the behavioral Verilog module. We can see the difference in length between the various cycles, with the first two motor pulses lasting precisely 18 seconds, the 3rd motor pulse lasting 9 seconds, and the last pulse with a width of 6 seconds.



Above: (Figure 1.5 – *Taken from same simulation as Figure 1.4, zoomed in with the gridlines adjusted*) Here we can see plainly that no 'OFF' signal is being sent to the device when the 'MTR' output changes (circled in green) at approx. 18 seconds. Also evident is that an 'ON' signal (circled in yellow) is being used at approx. 20 seconds to retrigger the device.



Above: (Figure 1.6) Waveform diagram taken from ModelSim showing the results of the automatic cycle timer test on the synthesized Verilog module. We can see that it appears nearly identical to the same test performed on the behavioral Verilog module, as seen in figure 1.4.

The Manual Cycle Timer Test

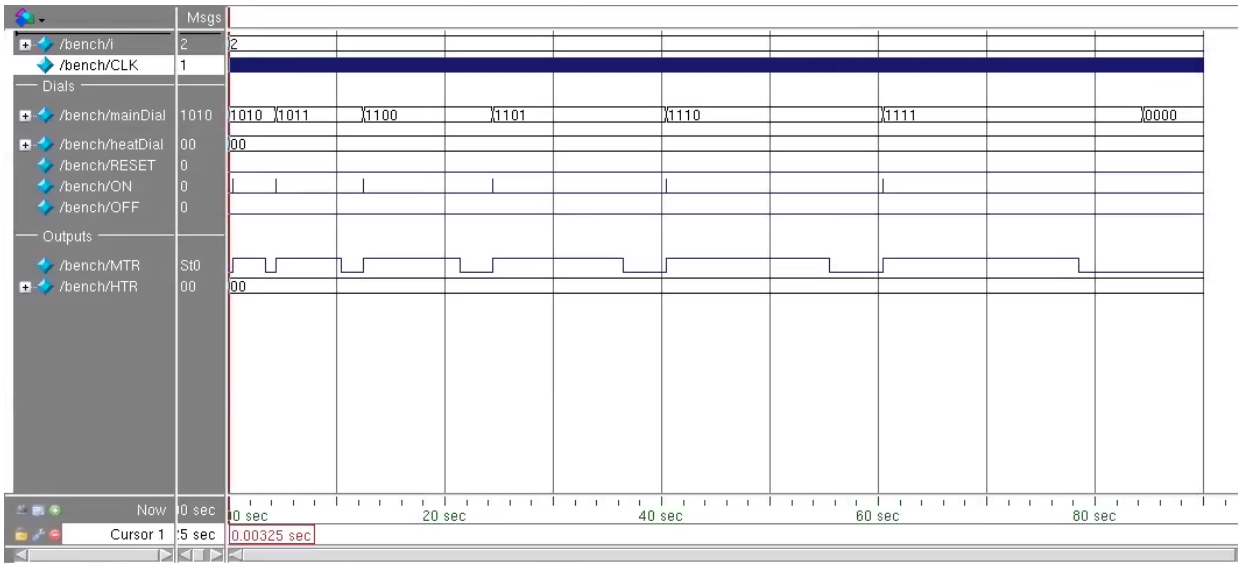
In the manual cycle timer test, we progress through the manual states and allow them to run to completion. This involves 6 different dial setting combinations.

We do not adjust the heat dial in this test, even though setting the heat dial does change the resulting state when using the manual cycles. There are two reasons for this:

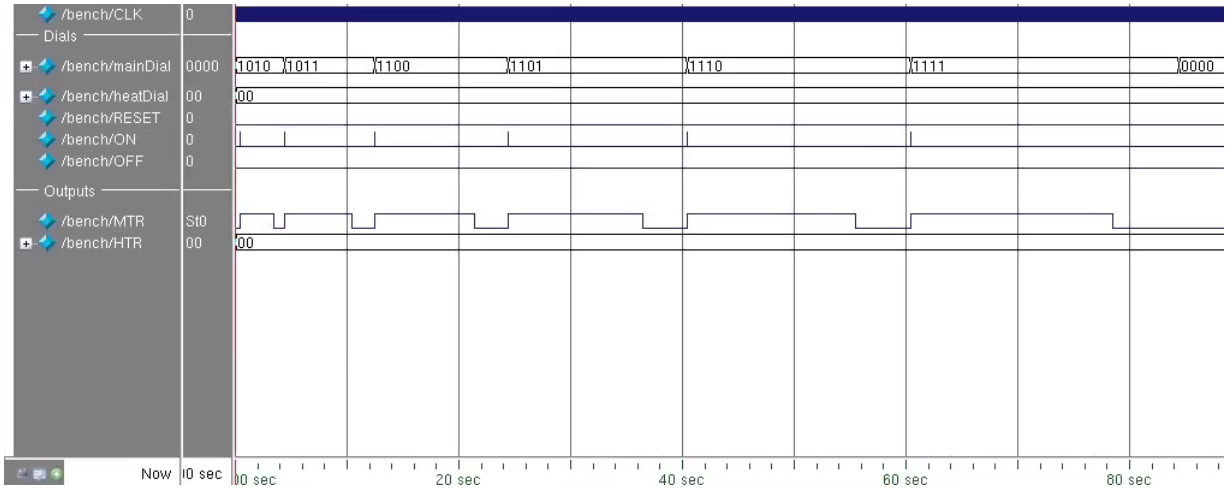
1) The intended output for each possible starting state was verified as correct in the “On/Off” test.
2) More importantly, the logic used to decrement through the states is invariant with respect to the four most significant bits in the state variable, which contain the current cycle. The four least significant bits contain the number of periods remaining in that cycle.

Therefore, in terms of this test, the four most significant bits of the state variable are irrelevant.

This test remains active for a little under 80 seconds (using the same period duration of 3 seconds).



Above: (Figure 1.7) Waveform diagram taken in ModelSim displaying the results of the manual cycle test on the behavioral Verilog module. We start at '15', which runs for a single period. Each time we increment the dial, we add an additional period to our manual cycle.



Above: (Figure 1.8) Waveform diagram taken in ModelSim displaying the results of the manual cycle test on the synthesized Verilog module. Primarily observing the 'MTR' output, we see that the results are nearly identical to those shown in figure 1.7 for the behavioral module.

Conclusions

In the conclusions section of the phase 1 report, three additions to the existing project were proposed. Each of these has been fulfilled in this phase.

A few issues can be identified with the design as it stands currently:

1) The testing period of 3 seconds would need to be adjusted to 15 minutes for a production unit.
2) The machine is not conservative in its use of memory. For example, the state register was expanded to a full 8-bit byte to increase readability/maintainability in the code. However, there are fewer than 2^8 possible states, and this expansion is a poor choice in terms of optimization.
3) The use of a clock counter on the primary clock to keep track of time may be an incredibly wasteful method in terms of power consumption.

Though the next phases of this project do not directly involve this design, these factors (and their possible solutions) will be considered before the project comes to a conclusion.