ECE 271

Digital Logic Design
Final Project

Nick Olson
Michael ASD
Sienna ASD

November 30, 2019
Instructor Shuman
Oregon State University

Contents

List of Figures

1 Project Description

Intro to Project paragraph. The inputs and outputs of the overall design immediately follow. An overall description diagram is then shown in **Figure 1** and a hardware diagram is shown in **Figure 2**.

- **Inputs:** inputs
- **Outputs:** outputs

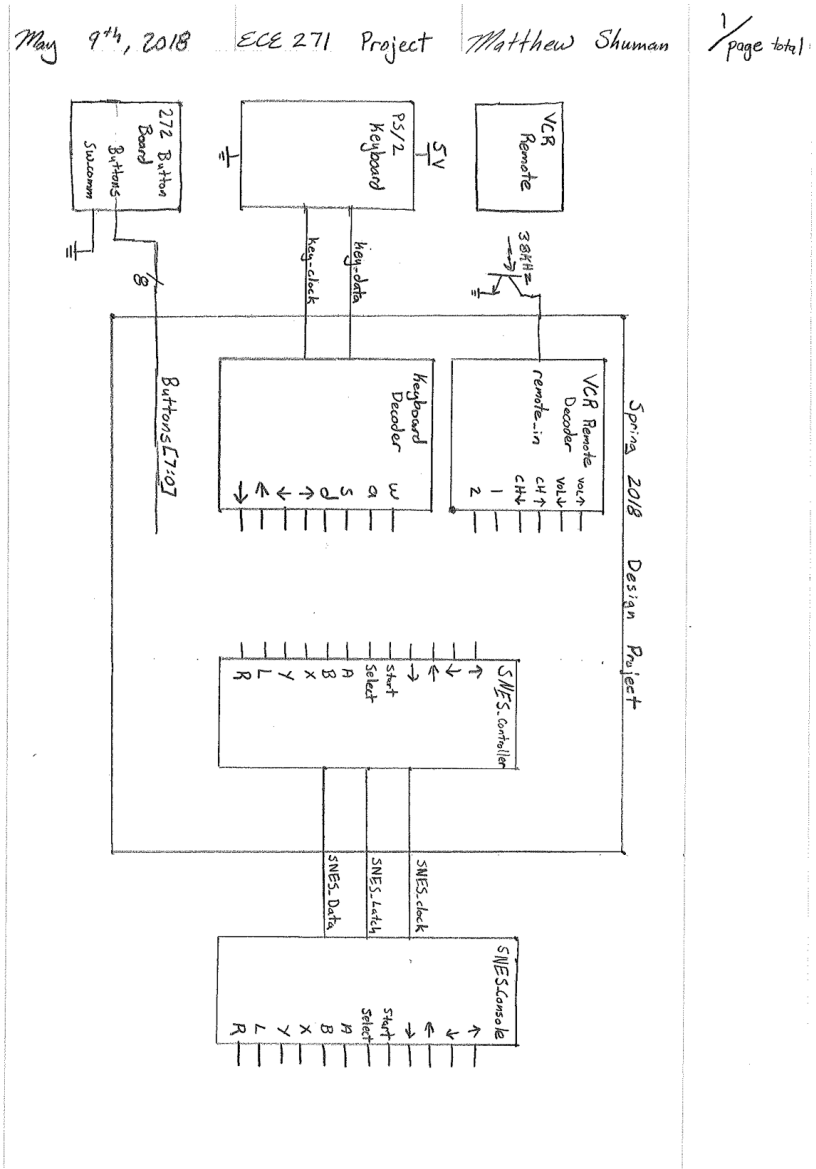

Figure 1: This image is legible, and conveys the point of the design. Your image can be hand drawn, but it must have straight lines, use your OSU ID. I don't recommend drafting this on the computer, because there aren't any decent tools to draw these block diagrams quickly.

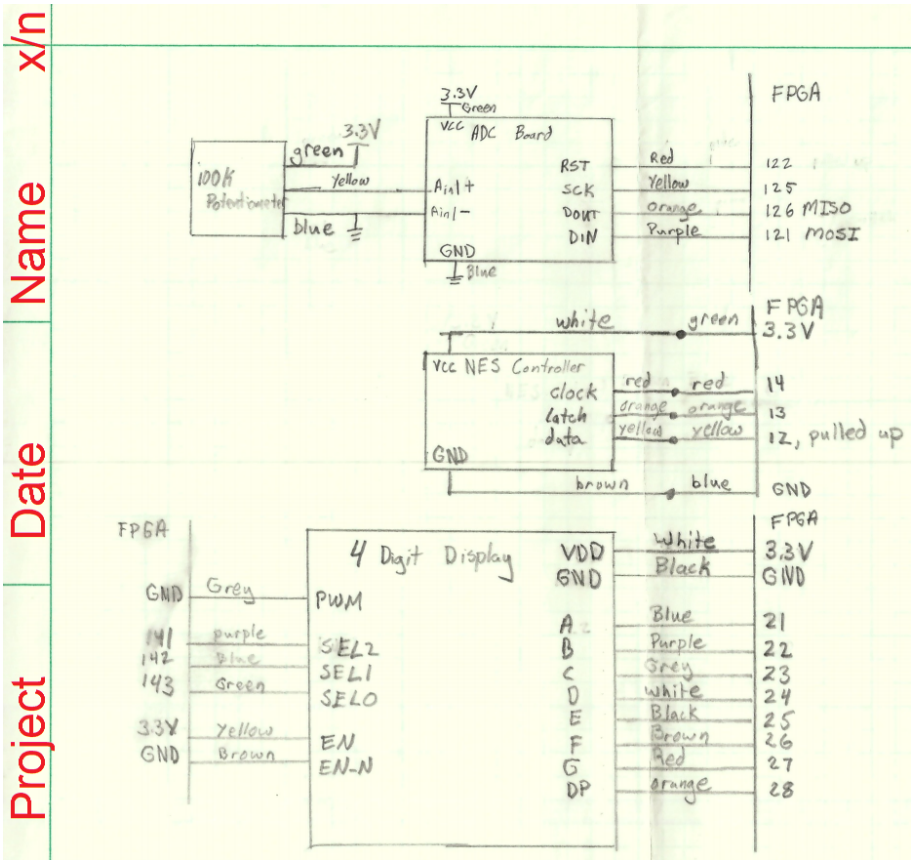

Figure 2: The hardware diagram shows which pins are used on the FPGA, module boards, and relevant supply voltages for the different pieces of hardware used in the system.

2 High Level Description

Top level introduction. The input and output specifications follow, a toplevel diagram follows in **Figure 3**, and the simulation results follow in **Figure 4**.

- **Inputs:** inputs
- **Outputs:** outputs

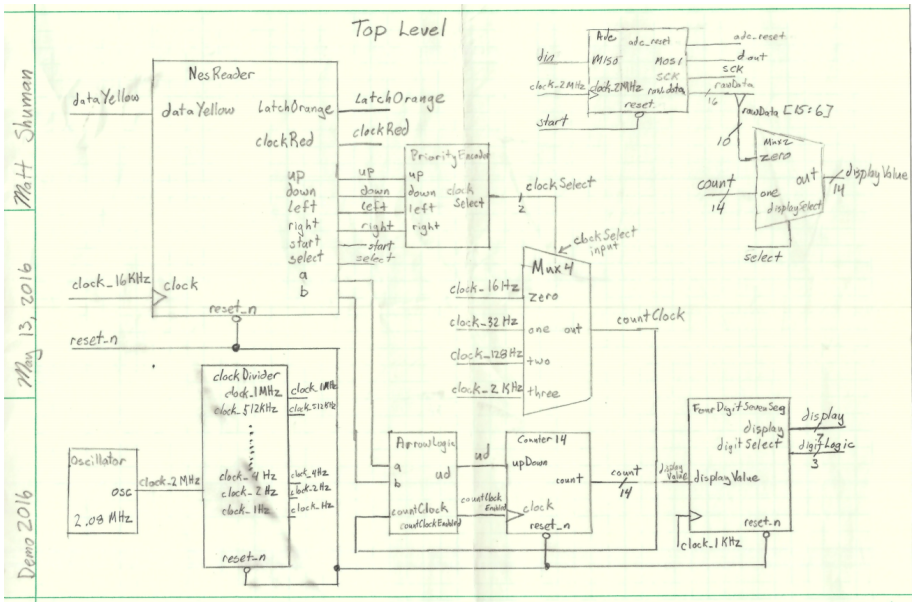

Figure 3: The top level design for the project. This would be improved by combining the priority encoder and Mux4 into a single clock select block. Combining the ArrowLogic and Counter14 would also make this diagram better. Use chapter 1 concepts wisely on this diagram, specifically hierarchy, modularity, regularity, and discipline.

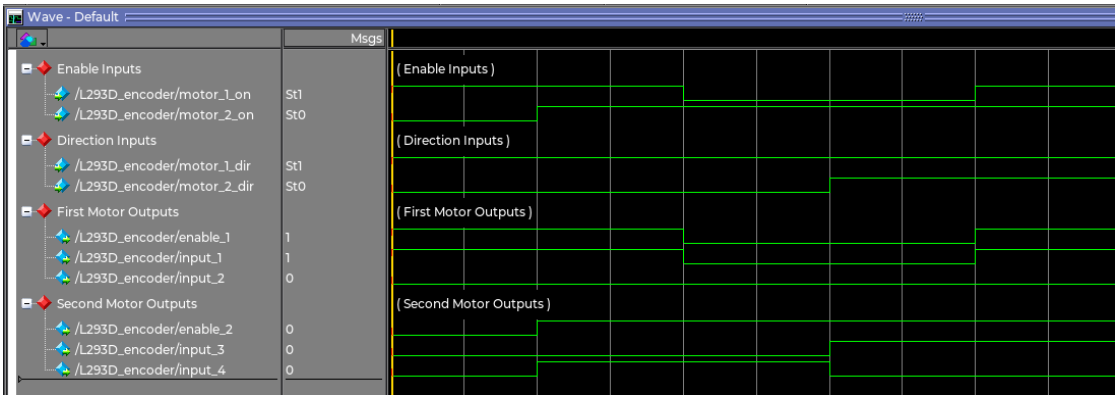

Figure 4: The simulation results of the top level design for the project.

The following subsections will discuss the inputs, outputs, designs, and simulation results of all elements of the design at two levels of scrutiny: functional units and individual blocks of digital logic.

GENERATE ME

2.1 DisplaysDecoder Functional Unit

The DisplayDecoder Module converts a 4-bit input value into a display value on the seven segment display of the FPGA. The DisplayDecoder is able to output a hexadecimal display value between 0 and F. A block diagram of the unit follows in **Figure A** and the simulation results for the unit follows in **Figure B**.

- **Inputs:** The DisplayDecoder module takes a 4-bit binary value input, data, as its only input.
- **Outputs:** The DisplayDecoder module outputs a 7-bit binary value that is used to activate specific segments in the FPGA seven segment display.

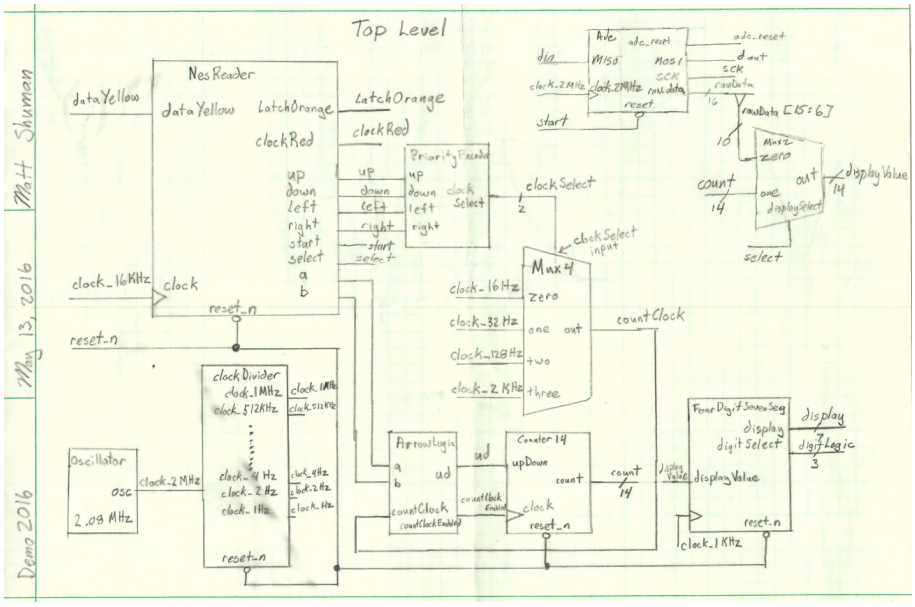

Figure 5: The logic design of the DisplayDecoder functional unit used in the final design.

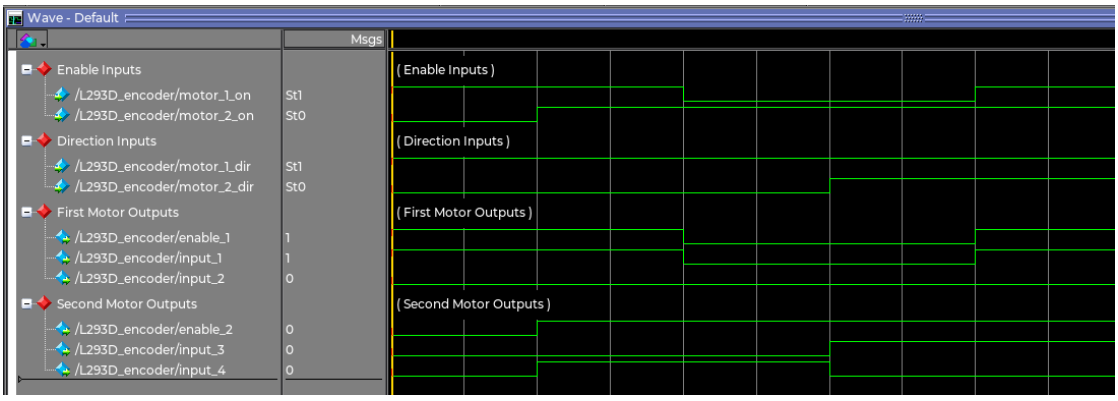

Figure 6: The simulation results for the DisplayDecoder Individual Unit.

2.2 vcr_decoder Functional Unit

The vcr_decoder module converts an IR signal sent from a VCR remote into a decimal value between 0 and 9. A block diagram of the unit follows in **Figure A**. A state diagram describing the unit as well as the simulation results for the unit follow in **Figure B**, and the details for each individual block comprising the unit follow after.

2.3 vcr decoder Functional Unit

The vcr_decoder module converts an IR signal sent from a VCR remote into a decimal value between 0 and 9. A block diagram of the unit follows in **Figure A**, the simulation results for the unit follows in **Figure B**, and the details for each individual block comprising the unit follow after.

- **Inputs:** The vcr_decoder module two inputs, clk and IR. clk is a 10 KHz clock signal that is used to drive the module. IR is the Infrared signal coming from the VCR remote that will be translated by the module.
- **Outputs:** The vcr_decoder module has a single output, displayValue, which is the 0-9 representing the IR signal that was received by the module as input.

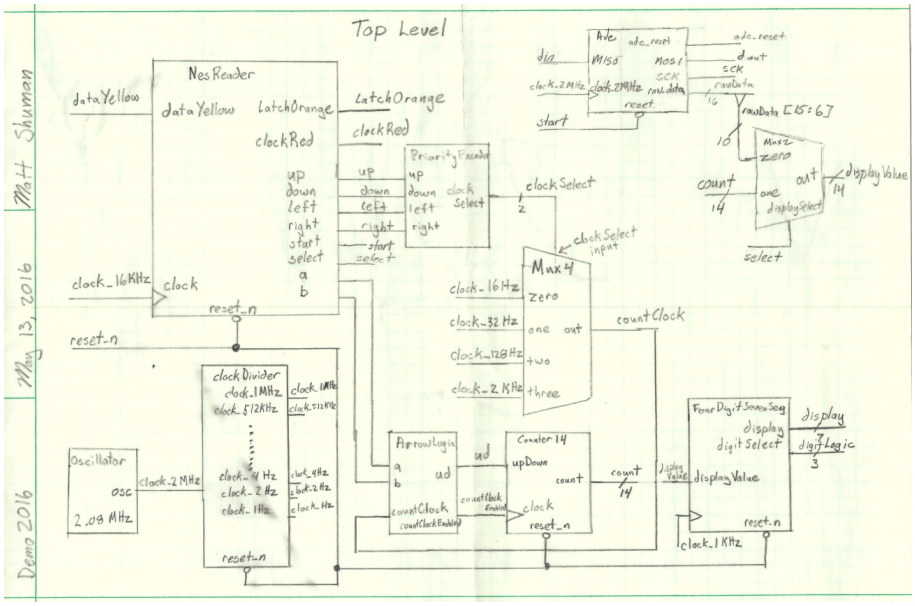

Figure 7: The logic design of the vcr_decoder functional unit used in the final design.

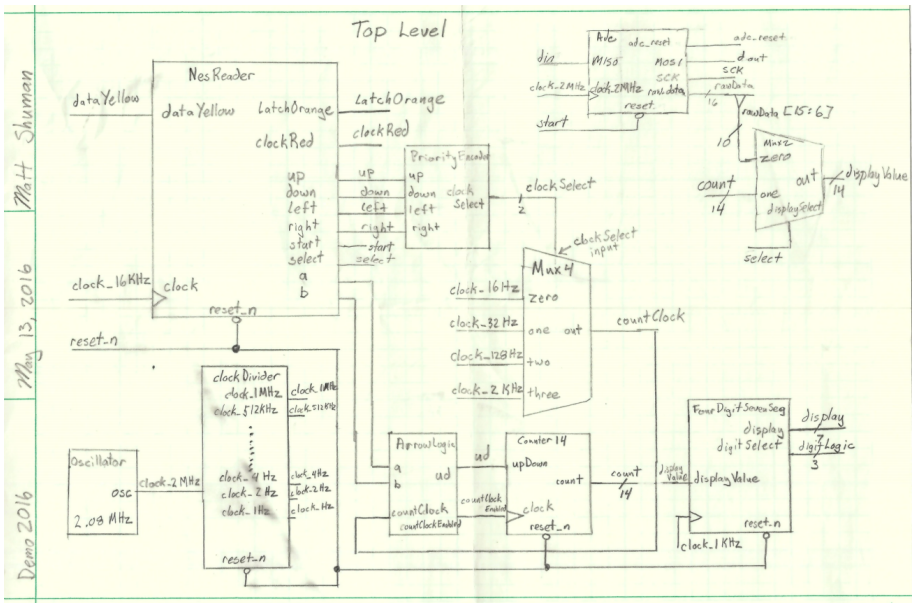

Figure 8: The State Diagram for the vcr_decoder functional unit used in the final design.

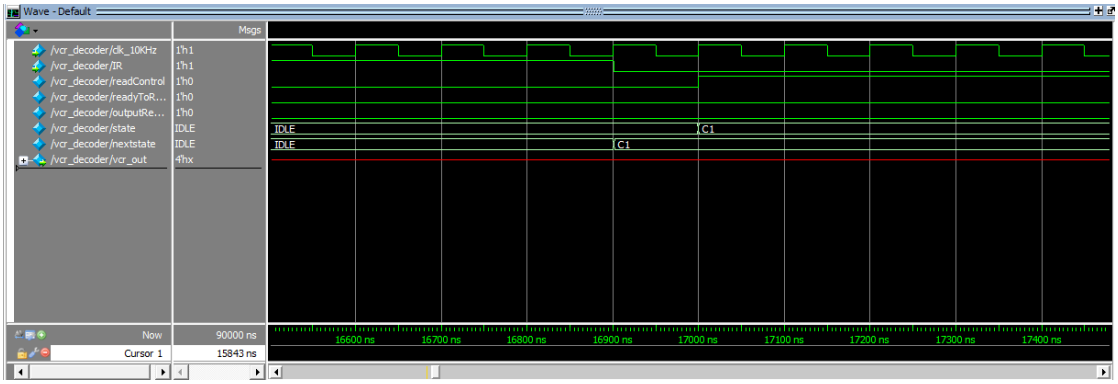

Figure 9: Simulation results of the vcr_decoder Functional Unit showing the state transition from the IDLE state to the C1 control state following the first time the IR signal goes to a logic LOW.

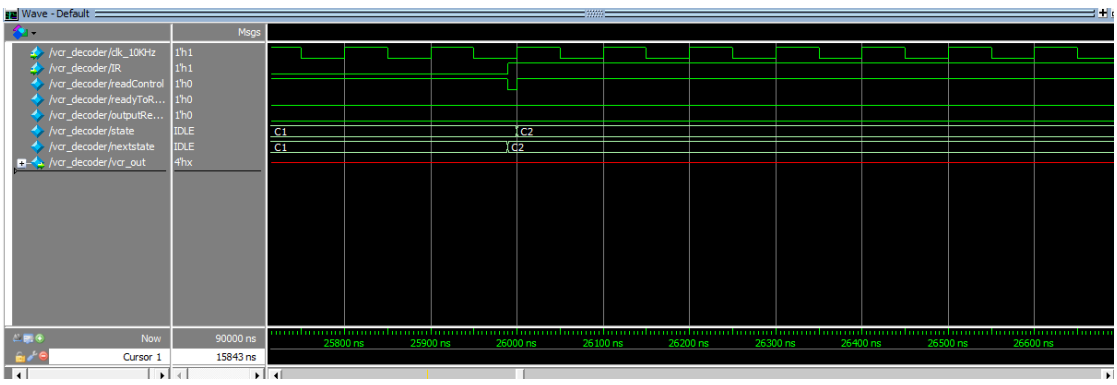

Figure 10: Simulation results of the vcr_decoder Functional Unit showing the state transition from the C1 control state to the C2 control state.

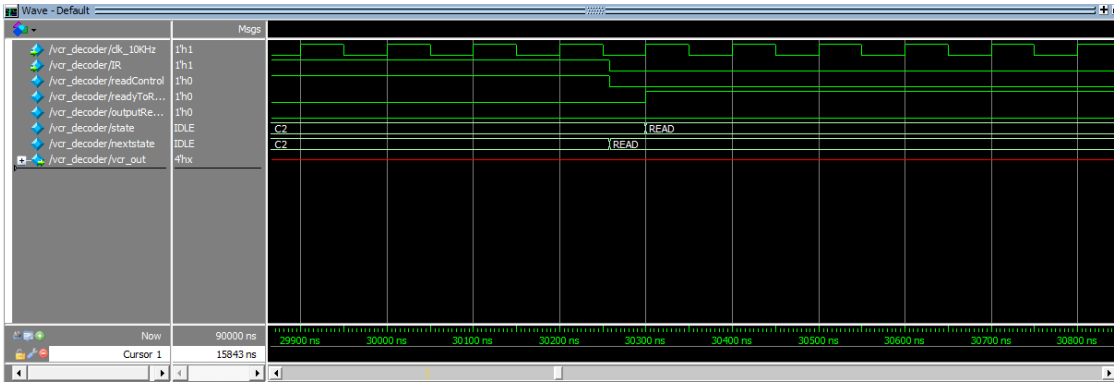

Figure 11: Simulation results of the vcr_decoder Functional Unit showing the transition from the C2 control state to the READ state, indicicating that it is currently reading a 32-bit IR signal corresponding the button on the VCR remote that was pressed.

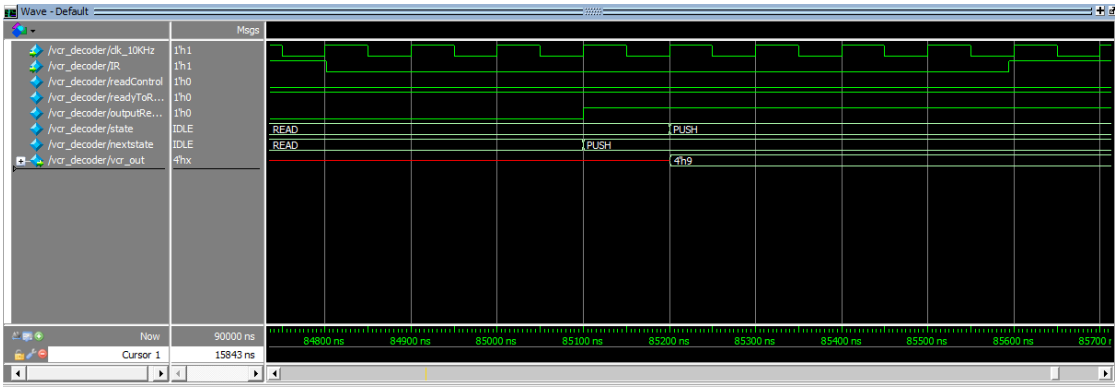

Figure 12: Simulation results of the vcr_decoder Functional Unit showing the transition from the READ state to the PUSH state indicating that the full 32-bit input value has been processed and an output value is produced.

2.3.1 ReadState Individual Block

The ReadState Individual Block is used to implement the READ state within the vcr_decoder Module. The block diagram (**Figure C**), State Diagram, and simulation results **Figure D** are provided below.

- **Inputs:** inputs
- **Outputs:** outputs

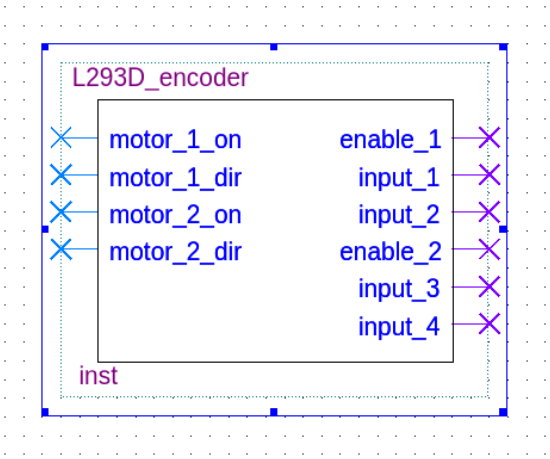

Figure 13: The block symbol of the ReadState individual block used in the vcr_decoder functional unit.

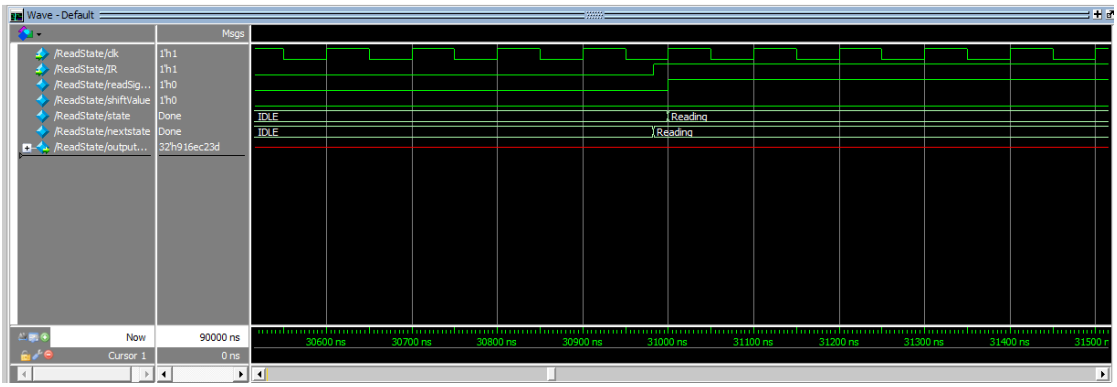

Figure 14: Simulation results of the ReadState individual block used in the vcr_decoder functional unit. The simulation shows the transition from the IDLE state to the Reading State within the ReadState Module. This transition is in response to the initial HIGH IR signal that represents the 32-bit value encoding the button that was pressed.

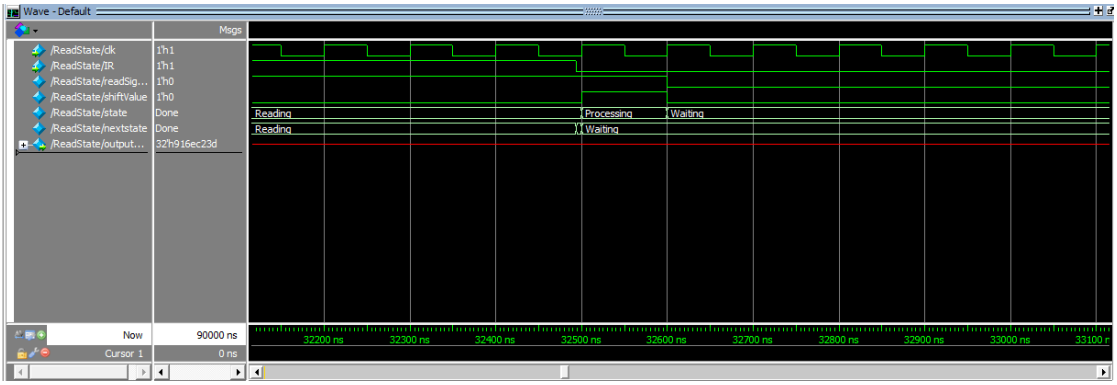

Figure 15: Simulation results of the ReadState individual block used in the vcr_decoder functional unit. The simulation shows Processing State of the ReadState Module. When the state machine is in the Reading state, that means it is actively reading a HIGH signal for IR. As seen in this waveform, when IR goes LOW again, the state machine switches to the Processing state in which it checks the length of the signal to see if it is a logic 1 or a logic 0. The 1 or 0 is then shifted into the shift register on the rising edge of the shiftValue signal, as seen in the waveform. The state machine then switches into the Waiting state, while it waits for IR to go HIGH again.

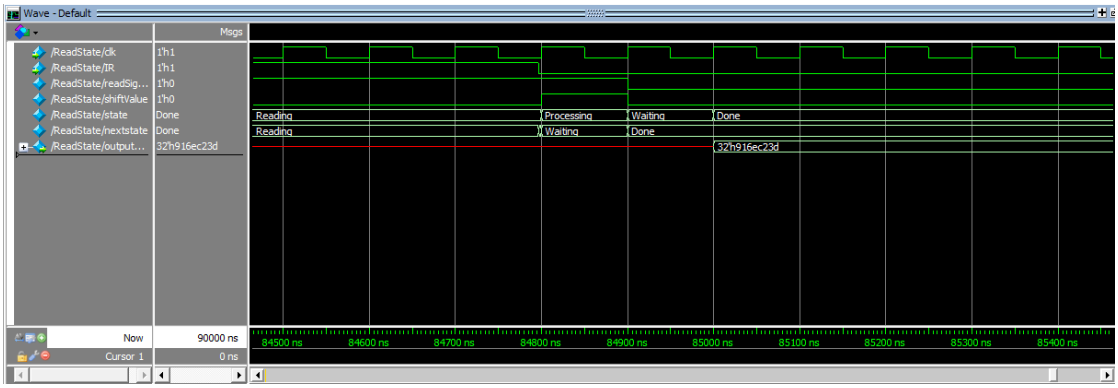

Figure 16: Simulation results of the ReadState individual block used in the vcr_decoder functional unit. The simulation shows the transition from the reading, Processing, Waiting cycle to the Done state in which it outputs the 32-bit hexadecimal value identified by the IR signal input. In each Waiting state, the state machine will check if 32-bits have been read into the shift register. If so, the state machine will switch to the Done state and output the value for the IR signal that it received as input. In addition, it will drive an output signal outputReady HIGH. This signal is used by the vcr_decoder module, to know when to output a result.

2.3.2 SignalDecoder Individual Block

The SignalDecoder Individual Block is used to convert the 32-bit value generated by the ReadState Individual Block into a 4-bit value by 0 and 15 that indicated which button on the VCR remote was pressed. The input and output specifications follow, as well as the block diagram (**Figure C**), and simulation results **Figure D** for the individual block.

- **Inputs:** The SignalDecoder Individual Block takes a single 32-bit input that corresponds to the complete IR signal that was read in by the ReadState Individual Block
- **Outputs:** The SignalDecoder Individual Block outputs a 4-bit value betwee 0 and 15.

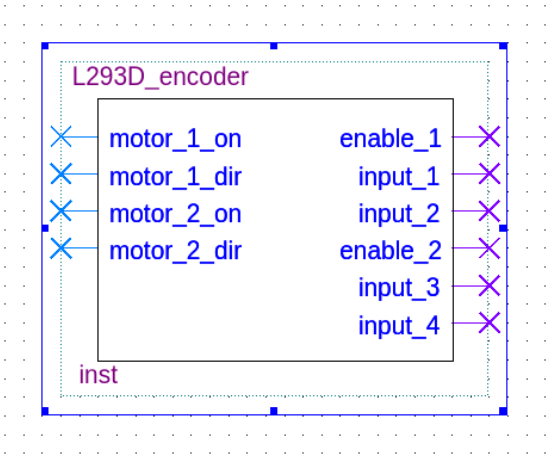

Figure 17: The block symbol of the (NAME) individual block used in the (NAME) functional unit.

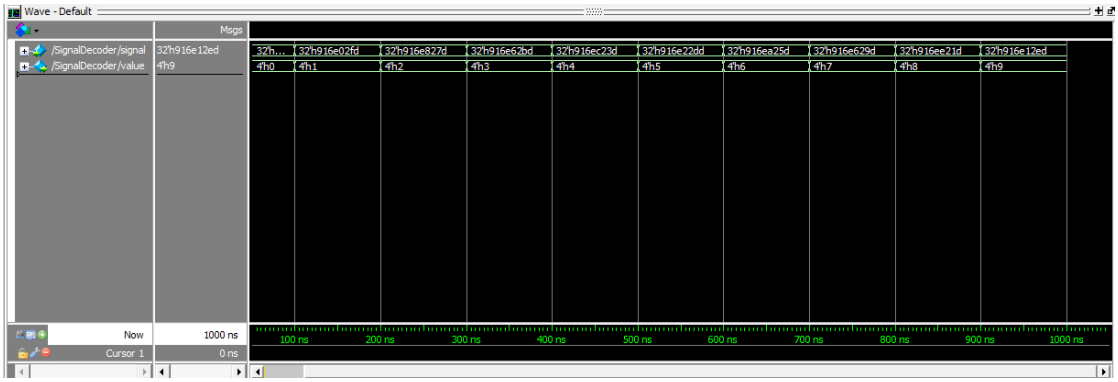

Figure 18: The simulation results of the SignalDecoder individual block used in the vcr_decoder functional unit.

2.3.3 ShiftRegister Individual Block

The ShiftRegister Individual Block is used to store the 32-bit IR signal value bit by bit as as it is read in by the ReadState Module. The input and output specifications follow, as well as the block diagram (**Figure C**), and simulation results **Figure D** for the individual block.

- **Inputs:** Inputs
- **Outputs:** Output

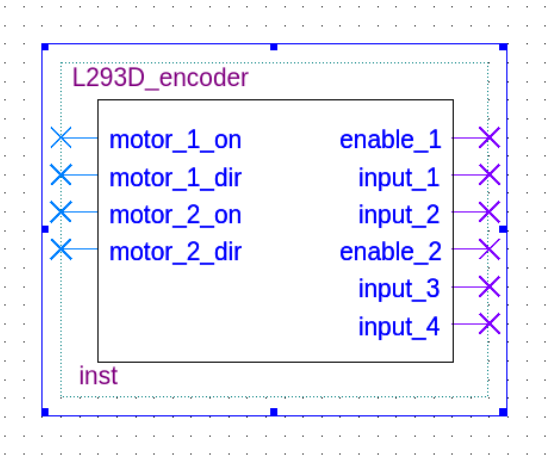

Figure 19: The block symbol of the ShiftRegister individual block used in the vcr_decoder functional unit.

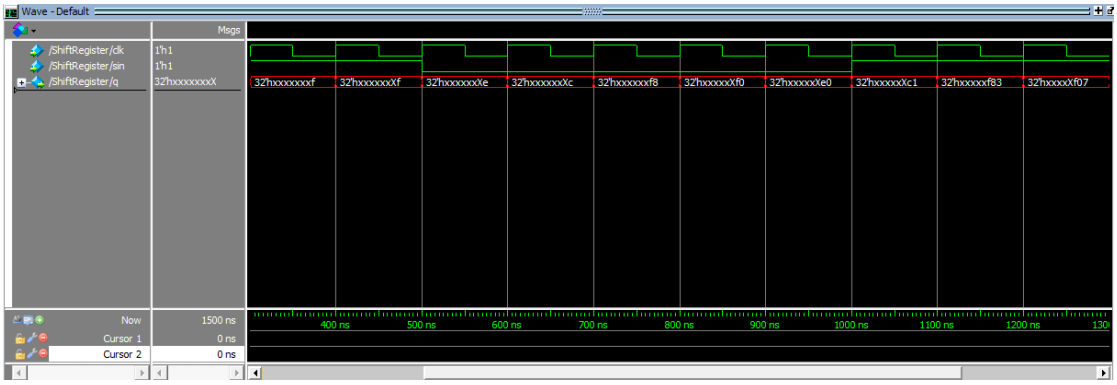

Figure 20: The simulation results of the ShiftRegister individual block used in the vcr_decoder functional unit.

A SystemVerilog Files

This appendix will list the SystemVerilog code used for each block used in the design project.
GENERATE ME

B Simulation Files (Do Scripts)

This appendix will list the Do Scripts used to simulate each block used in the design project.
GENERATE ME