Lab #5

Counter Design

Lab Report

Bradley Manzo
916953788

TABLE OF CONTENTS

INTRODUCTION

OBJECTIVES

PART I

- Design a Moore Machine using the provided random sequence of numbers in order to simulate the rolling of a 6 sided die.

PART II

- Design a Mealy Machine and understand the underlying differences between the two counter designs

PROCEDURE

DESIGN AND TEST PROCEDURE

PART I

- Design a Truth table with 3 columns titled Current State, Next State, and D-Flip-Flop Input in order to map the desired output to the behavior of the circuit.
- Perform Kmap calculations for Flip-Flop Inputs D2, D1, and D0, using the bits Q2, Q1, Q0 and Rb.
- After calculating the general equations, translate them to a Sum of Products circuit in Quartus, defining Key[0] as Clear, Key[1] as Rb, and Q[2..0] as Flip-Flop Outputs.
- After flashing the circuit to the DE-10 Lite Board, Demo the RNG capabilities of the board to the TA, and present the associated simulation in Modelism-Altera.

PART II

- Adapt the desired outputs to the behavior of a Mealy Machine and complete the paper design of said machine.

ANALYSIS/RESULTS

DATA

PART I:

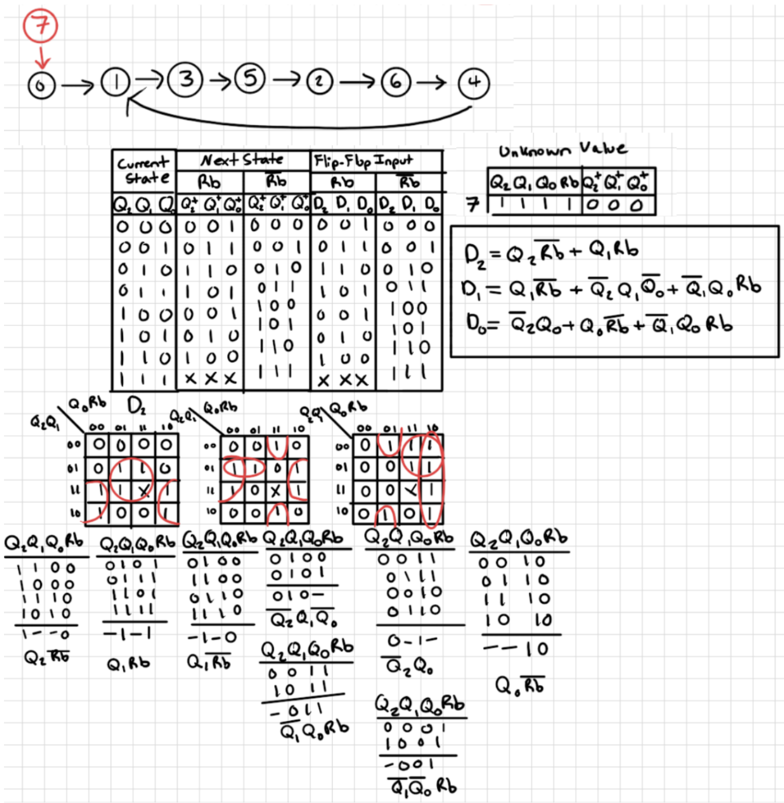

Figure 1: Moore Machine Paper Design

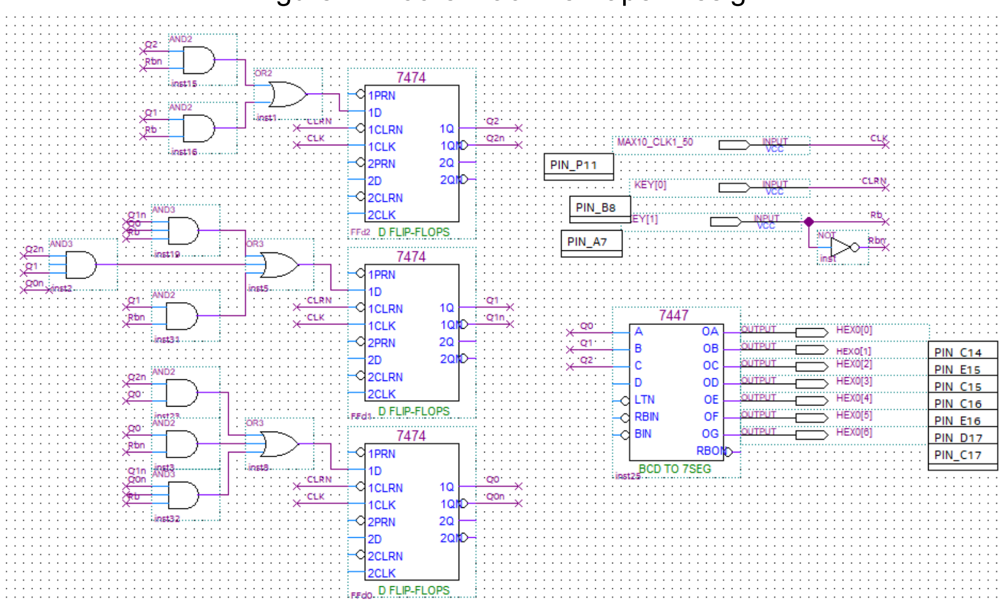

Figure 2: Moore Machine Quartus SoP Circuit

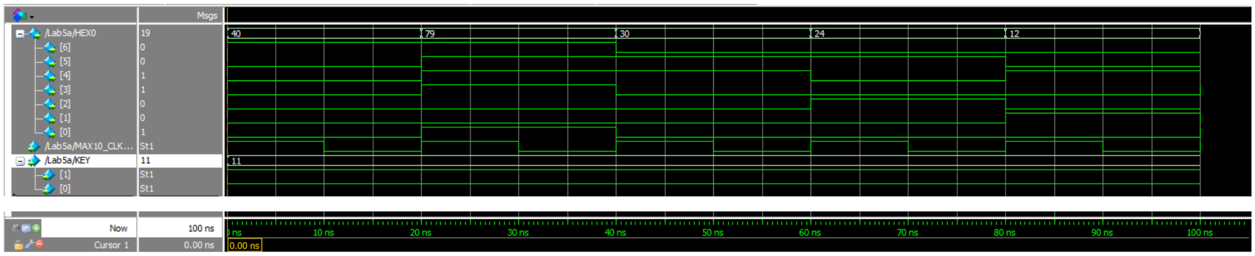

Figure 3: Moore Machine Simulated Waveform

PART II:

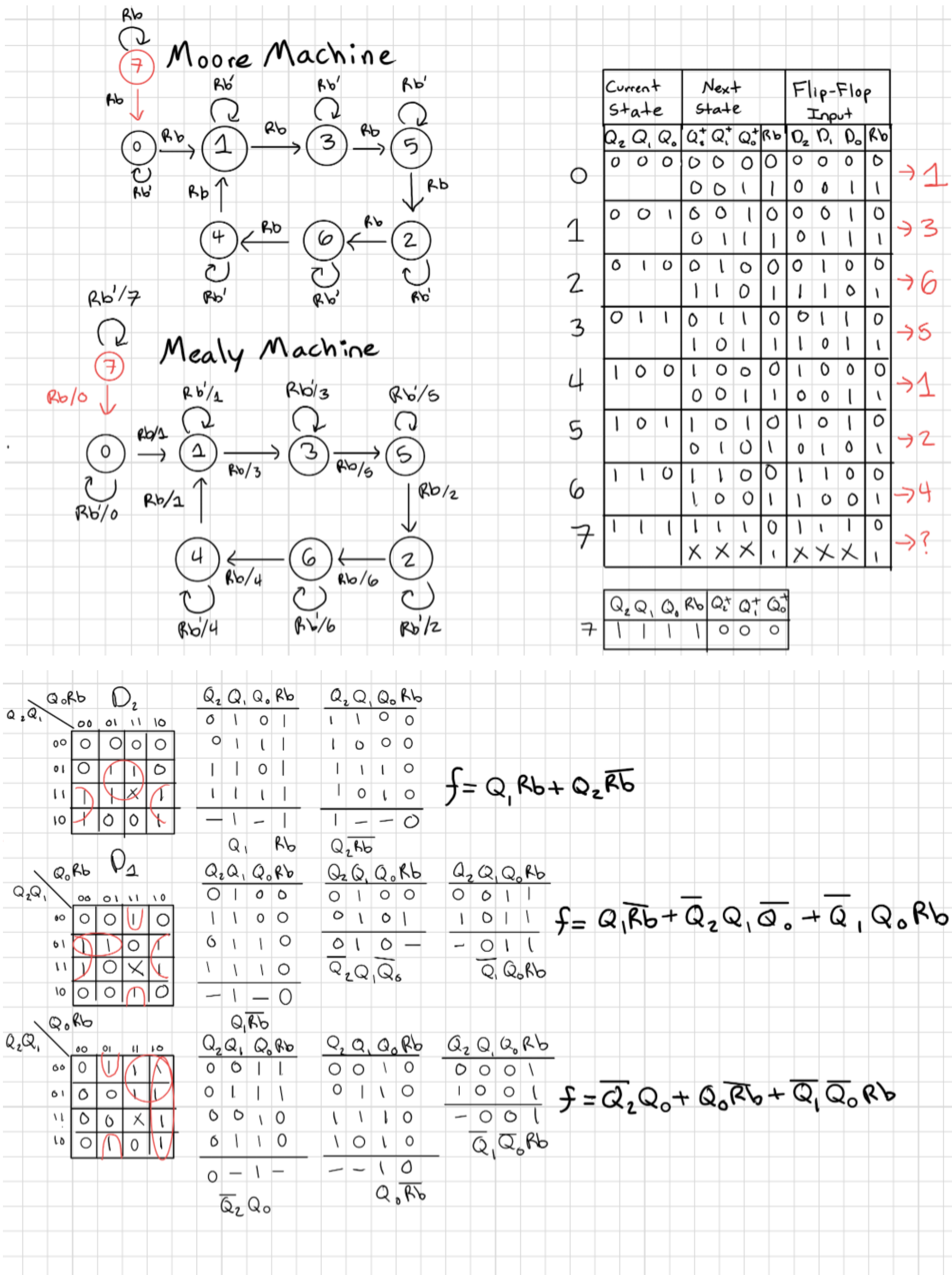

Figure 4: Mealy Machine Paper Design

ANALYZING RESULTS

PART I

- For Part I, designing the truth table properly was the hardest part, as I wasn't completely sure which values of the Next State corresponded to the Current State. However, after intensive proofreading and debugging, I have created a counter circuit that reliably produces a random number between 1 and 6 with a working reset button. I have confirmed this by rolling enough values to generate a reasonable spread. I have demoed the behavior and confirmed my simulated waveform with the TA.

PART II

- After researching what differentiates a Mealy Machine from a Moore Machine, I drew up an equivalent diagram operating as a Mealy Machine in theory, however on paper, it uses the same truth table and calculations as the Moore Machine.

ANSWERS TO HIGHLIGHTED QUESTIONS

PART I

- Compare Design I and Design II of the sequential circuit. Describe any advantages or disadvantages of the Moore design for this circuit.
    - After completing both designs for part I and II, there doesn't seem to be any difference as far as the number of gates required, however one disadvantage Design I has is that it requires the implementation of a clock since Moore machines are edge triggered. The output of a mealy machine is determined entirely from the current state and current input making it more versatile. Typically Mealy Machines will also have less states, as they aren't restricted to a single sequence, this would reduce the computing time of a circuit and would be a second disadvantage for Design I.

- A third design for the counter circuit might use a Moore machine where the state bits were not used as the outputs. For example, the state bits might be the binary count sequence 0 – 6. How would this design compare with Designs I and II(assuming your assigned count was not the straight binary count sequence 1 to 6)? Which design is likely to require the fewest gates? Justify your answer.
    - If the state bits were not used as the output and instead were simply in numerical order, it would be less cost efficient to produce and operate slower than an optimized SoP circuit. If you were to produce kmaps with said values you would end up with much more complicated general equations and therefore more gates.

CONCLUSION

- Overall this lab went hand in hand with lecture, as the Professor went over counter design in great detail. I did however have some trouble understanding how to designate the next state and how to implement the different values of Rb into the Flip-Flop Inputs. I also had a great deal of trouble getting my truth table absolutely correct as I had originally had the next state column as the order of the sequence instead of the state after the specific current state. However, after intensive debugging and talking to a few TAs I managed to produce a circuit that simulates a dice roll reasonably well. By completing this lab, I've learned how counters can be integrated into a moore machine in order to simulate any given sequence of state bits, and I've also learned the difference between Moore and Mealy machines insofar as their underlying mechanics. From what I understand Moore machines rely on the clock and the current state, while Mealy Machines rely only on the inputs and the current state independent of any clock. I will put this information to use in the final Lab 6.