Verification of Digital Systems, Spring 2019

J. A. Abraham

Homework No. 2

Assigned February 14, due February 28, 2019

Total Points: *180*

1. (**20 Points**) Consider a, b and c to be Boolean variables and $S(a, b, c)$ and $C(a, b, c)$ to be Boolean functions, such that $S(a, b, c) = a \oplus b \oplus c$ and $C(a, b, c) = ab + ac + bc$. Solve the following expressions and give the result in SOP form.

Review for those who are not familiar with quantifiers.
The universal quantifier, symbolized by $\forall vF$, with respect to a variable v in F, is used to express that the formula F holds for all values of v quantified. The existential quantifier, symbolized $\exists vF$, expresses that the formula F holds for some (at least one) value of that quantified variable.

Existential Quantification $\exists v[f] = f|_{v=0} + f|_{v=1}$
Universal Quantification $\forall v[f] = f|_{v=0} \cdot f|_{v=1}$

(a) (**5 Points**) $\forall a : S(a, b, c)$
(b) (**5 Points**) $\exists a : S(a, b, c)$
(c) (**5 Points**) $\forall a : C(a, b, c)$
(d) (**5 Points**) $\exists a : C(a, b, c)$

2. (**25 Points**) A deterministic four-state machine produces the output sequence Z in response to the input sequence X, shown below.

$X : 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1$

$Z : 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1$

Sketch the state machine diagram that produces the above sequence Z in response to the above input sequence X.

Is this description unique?

Can you sketch a deterministic state machine diagram with 3 states that complies with the above conditions? If yes, sketch the diagram. If no, why?

3. (**15 Points**) A deterministic four-state machine is in an unknown initial state. Show that the application of 55 consecutive 0s must leave the machine in the same state as the application of 7 consecutive 0s.

4. (**20 Points**) Find a regular expression on the alphabet {0, 1, 2} for the set of strings recognized by the graph shown in Fig. 1.

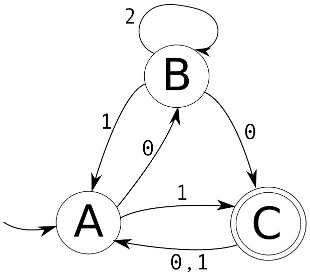


Figure 1: Graph for Problem 4

5. (**30 Points**) Which of the following sets can be recognized by finite-state machines? Justify your answer. In case that the set can be recognized by finite-state machines, show the state diagram (or part of it in case of large state machines). In each case the alphabet is {0, 1}.

(a) (**5 Points**) The set consisting of those strings that contain, for all k (k=0, 1, ...), k 1s and k+1 0s in any order.

(b) (**5 Points**) The set of strings in which the magnitude of the difference between the number of 0s and the number of 1s is a multiple of five.

(c) (**5 Points**) The set of strings in which every 0 is immediately preceded by at least k 1s and is immediately followed by exactly k 1s, where k is a specified positive integer.

(d) (**5 Points**) The set of strings that contain more 1s than 0s.

(e) (**5 Points**) The set of strings in which the number of groups of consecutive 1s equals the number of groups of consecutive 0s.

(f) (**5 Points**) The set of strings in which every possible sub-sequence of length seven appears at least once.

6. (**15 Points**) In which of the following cases do the two expressions describe the same set?

(a) (**5 Points**) $(011^* + 01^*0)^*$ *and* $(00)^*01(0+1)^* + \lambda$

(b) (**5 Points**) $(1^*0 + 001)^*01$ *and* $(1^*001 + 00101)^*$

(c) (**5 Points**) $0^*1(0 + 10^*1)^*$ *and* $(1 + 00^*1) + (1 + 00^*1)(0 + 10^*1)^*(0 + 10^*1)$

7. (**10 Points**) For each of the following expressions, find a transition graph that recognizes the corresponding set of strings.

(a) (**5 Points**) $(0 + 1)(11 + 0^*)^*(0 + 1)$

(b) (**5 Points**) $(1010^* + 1(101)^*0)^*1$

8. (**15 Points**) Answer the questions belop based on the following module definition

```
 module arbiter (clk, rst_n, req0, req1, grant0, grant1);


...
always @(posedge clk or negedge rst_n) begin
if (rst_n != 1'b0)
if (grant0 & grant1)
$display ("ERROR: Grants not mutex");
...
endmodule
```

(a) (**5 Points**) The above arbiter design has a display statement to check if the grants are mutually exclusive.

Why is this a bad idea?

(b) (**10 Points**) Correct the mistake in the following assertion so that it asserts when the grants are not mutually exclusive.

```
assert property ( @(posedge clk) disable iff (rst_n)  (grant0 + grant1));
```

9. (**5 Points**) Consider the SVA assertion and a corresponding state diagram in Fig. 2. Label the states where the assertion holds.

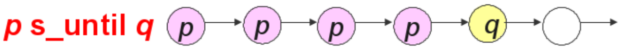


Figure 2: State diagram for Problem 9.

10. (**5 Points**) Explain the difference between @posedge and $rose.

11. (**20 Points**) The block diagram of a target device along with input and output ports is shown in Fig. 3. The target device has a first-in-first-out type memory that can store up to 64 bytes of data.

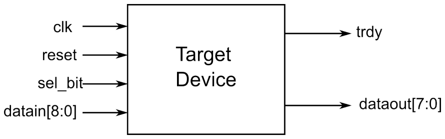
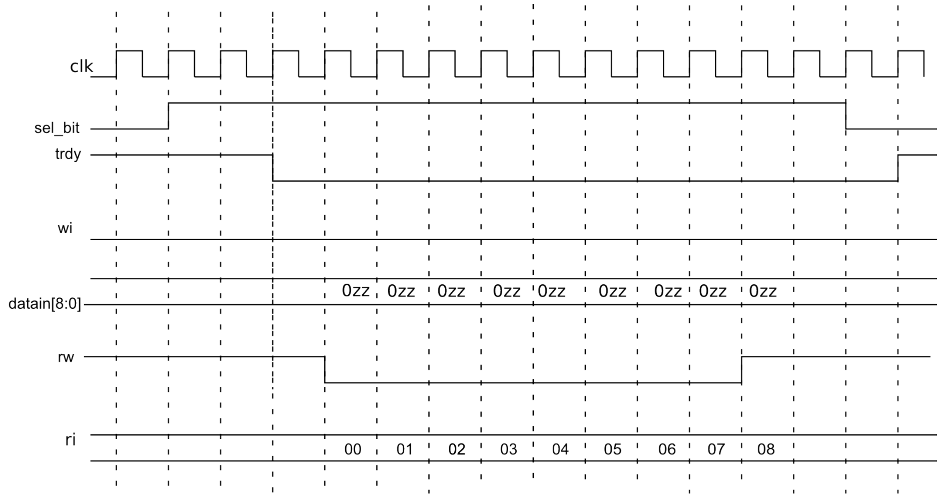
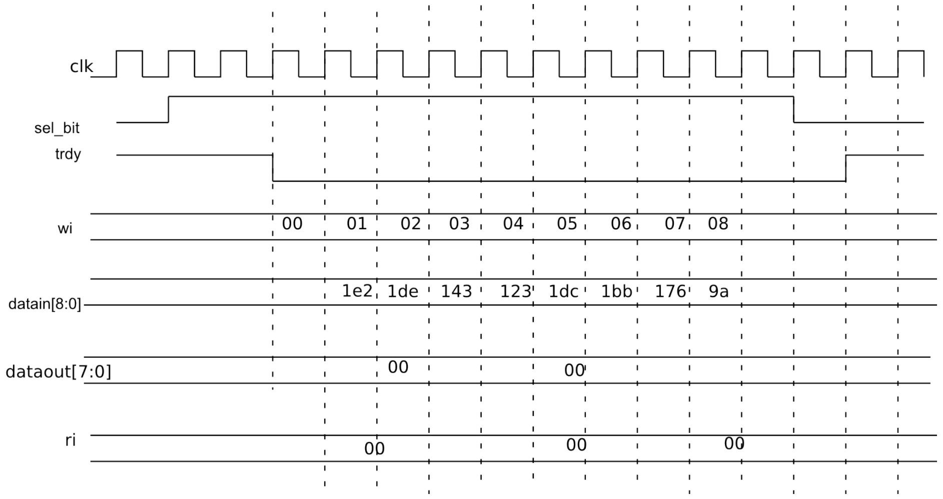


Figure 3: Target device (Problem 11)

The target device waits for the signal "sel_bit" to be asserted. Once the signal "sel_bit" is asserted, the target must acknowledge by asserting the signal "trdy" after 2 clock cycles. After asserting the signal "trdy", the target device waits for a valid data and a valid write signal in the next posedge of clk. Once a valid write signal is detected, the incoming data is stored in the target device in locations starting from the most updated value of the write pointer (wi) register. If it is a read transaction, then the target device reads out 8 data points from its memory using the current read pointer location (ri) as the starting address. The type of transaction is indicated by the MSB of the bus "datain", i.e., (datain[8] = 0:read; datain[8] = 1:write). In a read transaction, the data read appears on the bus vector "dataout". When the transaction is complete, the signal "sel_bit" is de-asserted and one clock cycle after that, the signal "trdy" is de-asserted. The sample waveforms for a target write and read are shown below.



Target Read Transaction (Problem 11)



Target Write Transaction (Problem 11)

Write assertions for the following.

(a) If a target is selected, then it should assert the signal “trdy” after 2 clock cycles.

(b) At the end of a transaction, the “sel_bit” signal is deasserted. One clock cycle after that, the signal “trdy” should be de-asserted.