Digital Design Verification (CS676)
End Semester Question
Dept of Computer Sc and Engg, IIT Madras
Answer all questions

NB:

1. All the variables used in the codes should be defined.
2. Codes should be commented.

Full Marks:50
Time:3 hourse

1. Compute the expression coverage for the following expression in a run that has encountered these values for a, b and c: (1,1,0), (0,0,1), (1,0,1) and (0,1,0).

$$\boxed{((a\,?\,\overline{b} : c) \oplus (ab + \overline{\overline{a}c}))}$$

(5 marks)

2. Write a verilog assertion for the property that vector S of length 8 is always of even parity and the Hamming distance among its feasible values is 2. (Hamming distance between two values is the number of bits in which the two values differ).

(5 marks)

3. There are two bus masters linked to a central arbiter by individual request (REQ#) and grant (GNT#) signals. Each master has its own REQ# and GNT# lines. Apart from this the arbiter also receives an input reset (RST#) signal. Consider the following timing specification:

Whenever the signal RST# is deasserted and in the next clock cycle REQ# is asserted, the GNT# signal is asserted after 3 clock cycles after the REQ# assertion and remains high for 7 clock cycles. Write an 'e' code snippet to check for the above specification. Add to your code an assertion to check that only one GNT# signal is asserted by the arbiter at an instant of time.

(5 marks)

4. Write a verilog code to generate the clock waveform shown in the following figure (Fig 1). The fall and the rising transitions have 300 ps and 200 ps jitter respectively.

(5 marks)

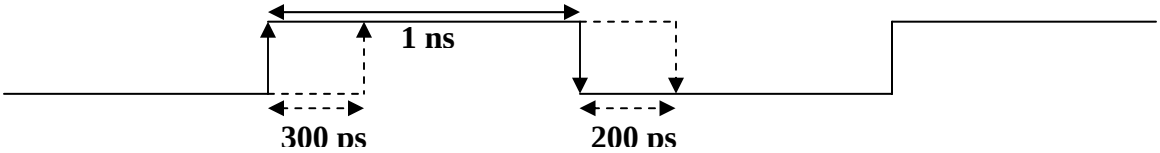

Fig 1. Waveform with timing jitters

5. 2 processor cores access a shared DPRAM, using a read/write wrapper. The description of the wrapper is:

dram(addr0, data0, addr1, data1)

The dimension of the data and address bus are 1 bit each. The wrapper written in verilog have tasks **read_memory** and **write_memory** which you can invoke from a top level test-bench to read or write.

The cores also access a common register "LOCK". Initially, the memory contents and the LOCK register is set to 0.

Each of the two processors maintains three states: N (non-critical), T(trying) and C (Critical) to access a critical resource with Mutual Exclusion. The processors maintains a register "KEY" and the state transition of a processor[i] (i = 0 or 1), depicted in the following diagram depends on the values of the register key and the content of the memory dram[i].

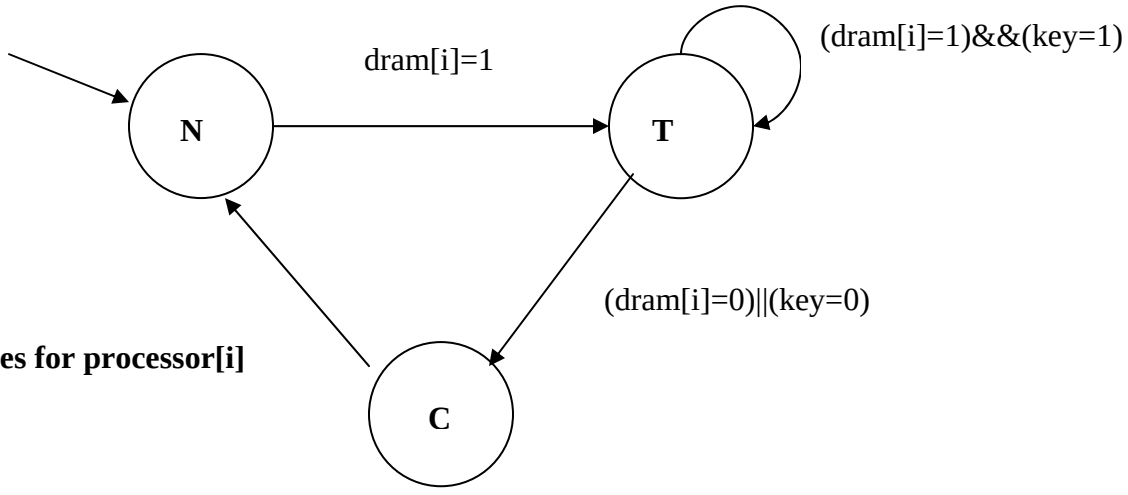

Fig 2. States for processor[i]

Each process accesses a central block called "TestnSet". This block is accessed when the processor is in the state 'T'. The block probes the value of the register LOCK and updates the register KEY of the accessing processor. Simultaneously it sets LOCK to 1. As shown in the transition diagram, the processor goes to the state C if any one of dram[i] or key is 0.

Upon going to the state C, the processor sets dram[i] to 0. After performing its task with the critical resource, it checks whether the other processor is waiting. This is done by checking dram[(i+1)%2]=1 for the other processor. If yes, it sets dram[(i+1)%2]=0 for the other processor, else it sets LOCK as 0.

a) Develop an e-based framework to verify the above functionality. You can assume that the current state of the processor can be accessed through the variable currentstate.

b)

The two properties that we are interested to verify are:

1. Mutual Exclusion: That is both the processes are not accessing the critical section at the same time.
2. Liveness: If a process starts to wait for the critical section, it eventually gets it.

Develop an e-based framework to verify the above properties. Any assumptions made should be clearly stated.

(Hint: refer to the topic of verification of state machines discussed in the class).

(6+4=10 marks)

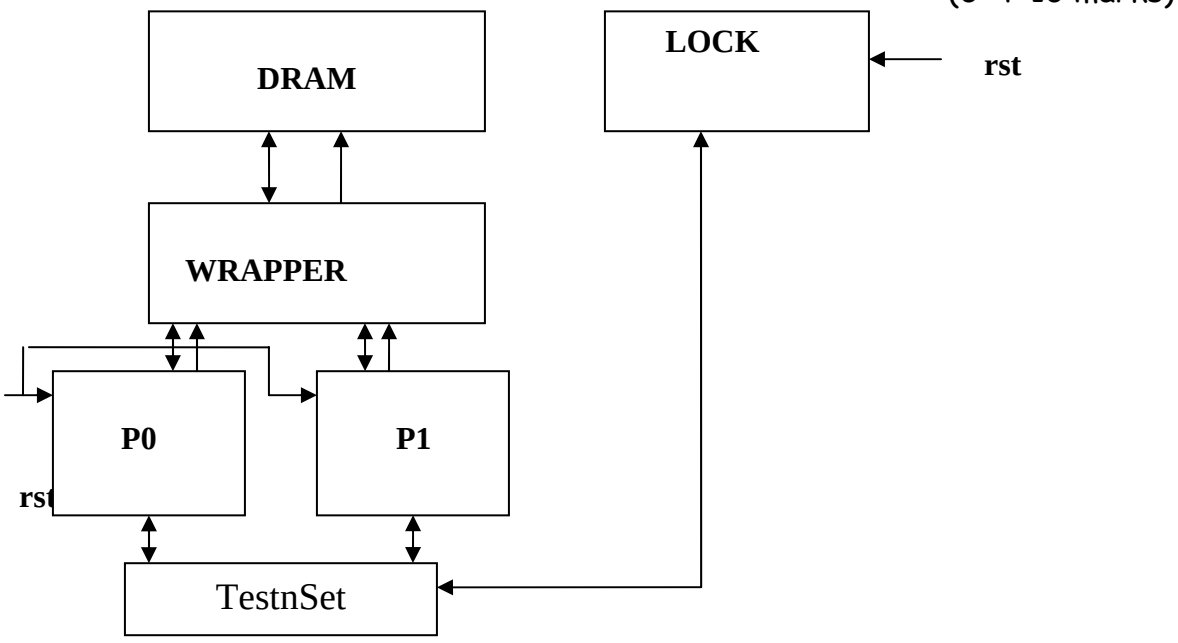

Fig 3. The Mutual Exclusion Hardware

6. Find a path from the initial state q_3 which satisfies aUb. Determine whether in the model M, q_3 satisfies the LTL (Linear Temporal Logic) property aUb. (5 marks)

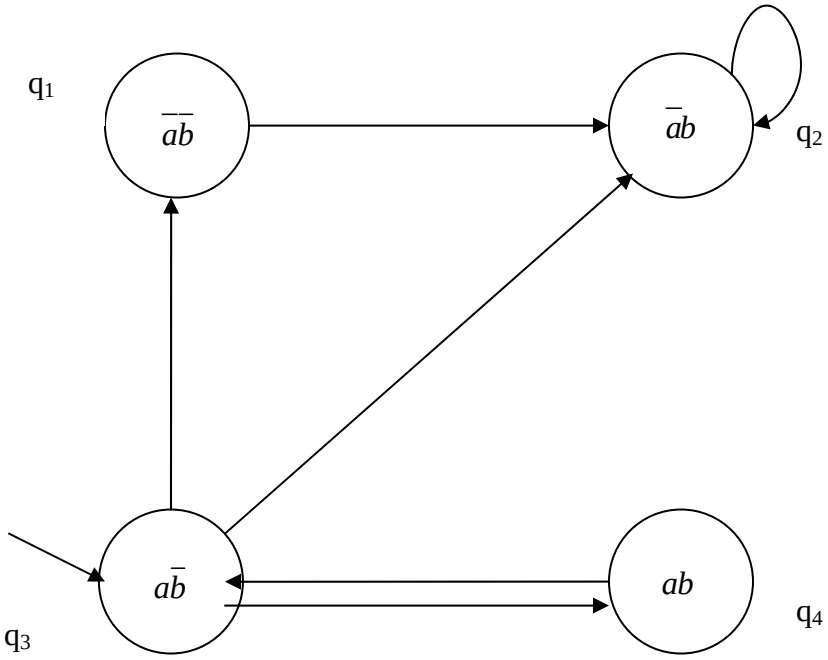

Fig 3. A model M

7. Express the following properties in LTL:

i) Whenever p is followed by q (after finitely many steps), then the system enters an interval in which no r occurs until t.

ii) Between the events q and r, event p is never true.

(5 marks)

8. Draw the BDDs for $(\bar{a}+b)$ and $(a+\bar{b})$, where a and b are Boolean variables. Evaluate the ApplyAnd between them with the variable ordering a<b. What is the effect of changing the variable ordering to b<a. What are such kinds of Boolean function called?

(5 marks)

9. Apart from verification of digital design, can you imagine some application areas where the techniques which you learnt in the course can be applied?

(5 marks)