14:332:437 Digital Systems Design
Fall 2011
Practicum 2 Version C
Electrical and Computer Engineering Department
College of Engineering
Rutgers University

Prof. Michael L. Bushnell

November 18, 2011

STUDENTS ARE EXPECTED TO WORK INDEPENDENTLY ON THIS PRACTICUM. NO COLLABORATION IS PERMITTED. YOUR WORK MUST BE YOUR OWN. NO USAGE OF WEB BROWSERS, EMAIL, FTP, OR TELNET IS ALLOWED DURING THIS SESSION. NO TALKING IS PERMITTED. HOWEVER, YOU MAY USE YOUR TEXTBOOK AND NOTES.

1. Design a simple *finite state machine* (FSM) with the behavior given in Table 1. Note that the machine goes to state A when */init* is pulled to logic 0. The machine changes states on the falling CLK edge. In state C, when $q == 1$, the machine initializes a modulo-19 up counter to 0, and stays in state C. In state C, when $q == 0$, the machine increments the up counter and stays in state C until the counter reaches 18, at which point the machine sets the counter to 0 and transitions to state H. The asterik (*) in Table 1 shows the state transition that will be delayed until the counter rolls up to 18.

Table 1: Finite State Machine Behavior

Present State	Next State / Output (z)	
	$q = 0$	$q = 1$
A	G/0	C/0
B	H/0	G/0
C	H*/1	C/0
D	E/0	F/0
E	A/0	G/0
F	J/0	J/0
G	D/0	B/0
H	A/0	F/0
J	H/0	G/0

Use the method of partitions to find and eliminate the redundant states in the finite state machine. A new state created by combining states should take the letter name of the lowest-lettered combined state. Write behavioral VERILOG code to implement this machine in the pre-2001 dialect of VERILOG. Show the TA your combined states before proceeding further. Use the **verilog** (Cadence Verilog-XL) simulator to simulate this machine. Use the following partial testbench from the file **/vlsi/home/bushnell/pc2vctestbench.v** to exercise your machine:

```
  begin
    init = 0;
    clk = 1;
    q = 0;
#5  clk = 0;
    init = 1;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 1;
    clk = 1;
#5  clk = 0;
#5  q = 1;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 1;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 1;
    clk = 1;
#5  clk = 0;
#5  q = 1;
    clk = 1;
#5  clk = 0;
#5  q = 0;
```

```
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
```

```
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
#5  q = 0;
    clk = 1;
#5  clk = 0;
  end
```

Credit will not be given if you do not use EXACTLY these timing waveforms.