

14:332:437 Digital Systems Design

Fall 2007

Problem Set 7

Electrical and Computer Engineering Department

College of Engineering

Rutgers University

Prof. Michael L. Bushnell

Assigned: November 7, 2007

Due: November 14, 2007

STUDENTS ARE EXPECTED TO WORK INDEPENDENTLY ON BOTH HOMEWORK AND EXAMINATIONS. NO COLLABORATION IS PERMITTED. YOUR WORK MUST BE YOUR OWN.

1. *State Reduction by the Method of Partitions.* The following state transition table for a finite state machine has many redundant states. Use the method of partitions to reduce the number of states in the machine.

	XY/Z			
	00	01	10	11
A	B	C	D	F/1
B	C	I	G	E
C	H	D	G	A
D	A	H	G	I
E	F	I	B	G
F	C	I	E	A
G	H	I	E	B
H	F	D	B	A
I	A	C	B	D

2. *Synchronizing Sequences.* Find whether synchronizing sequences exist for the states in the following machine. If they exist, write down the shortest synchronizing sequence for each state that has one.

	X/Z	
	0	1
A	F/0	B/1
B	B/0	C/1
C	B/1	D/0
D	B/1	E/1
E	C/0	C/0
F	E/0	D/1

3. *Time Redundancy.* Design a Boolean/arithmetic unit in Verilog that uses two extra bit positions and *Recomputing with Shifted Operands* (RESO) (time redundancy) to determine whether there is a fault in the unit. What are the resource penalties of using RESO? Can you diagnose the broken bit position for the Boolean operations? Can you diagnose the broken bit position for the arithmetic operations? Write Verilog code for this RESO design, synthesize it, and simulate it. Turn in the following: Verilog code, behavioral block diagram, diagram of synthesized logic, behavioral simulation, and logic simulation.

Signal	Type	Meaning
<i>clk</i>	input	System clock
<i>reset</i>	input	Active-low system reset
<i>A</i>	input	32-bit first operand
<i>B</i>	input	32-bit second operand
<i>carry_in</i>	input	Input carry signal
<i>carry_out</i>	output	Output carry signal
<i>add</i>	input	Add <i>A</i> to <i>carry_in</i> and <i>B</i> when 1, producing a 32-bit output and a carry output
<i>subtract</i>	input	Subtract input <i>B</i> and <i>carry_in</i> from <i>A</i> when 1, producing a 32-bit output and a borrow output
<i>andfunc</i>	input	AND <i>A</i> with <i>B</i> when 1, producing a 32-bit output (set <i>carry_out</i> to 0)
<i>orfunc</i>	input	OR <i>A</i> with <i>B</i> when 1, producing a 32-bit output (set <i>carry_out</i> to 0)
<i>bit_compare</i>	input	AND <i>A</i> with \overline{B} when 1 (set <i>carry_out</i> to 0)
<i>broken</i>	output	Set to 1 when the ALU has at least one fault, 0 if no faults
<i>Z</i>	output	32-bit result of all operations