University of California, Santa Barbara

Department of Electrical and Computer Engineering

ECE 152A – Digital Design Principles

Final Exam
August 29, 2007

Name _____

Perm # _____

Lab Section _____

Problem #1 (25 points) _____

Problem #2 (25 points) _____

Problem #3 (25 points) _____

Problem #4 (25 points) _____

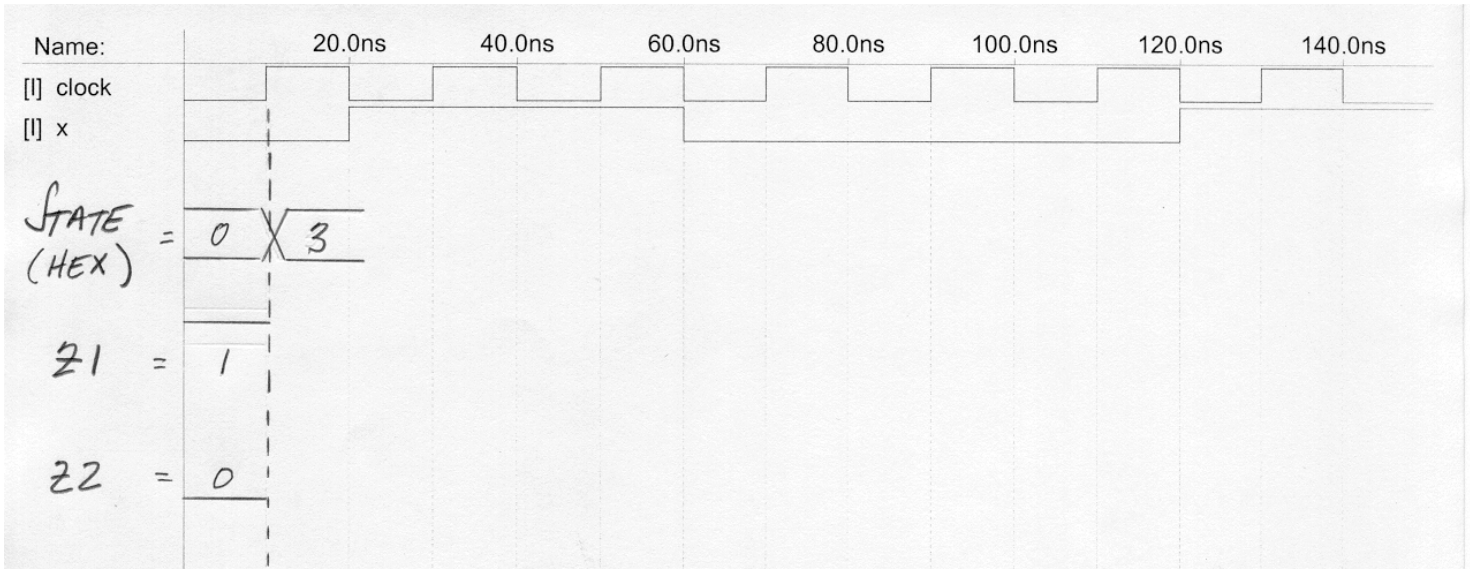
Total (100 points) _____

- This is a 75 minute exam; closed book, closed notes, no calculators.
- Answer all questions on the exam.

Problem #1.

Complete the functional (zero delay) timing diagram below for the circuit realized by the following Verilog code:

```
module final (clock,x,z1,z2);

input clock,x;
output z1,z2;

wire z1;
reg z2;
reg [1:0]state;

      assign z1 = x & state[1] | ~x & ~state[1] & ~state[0];

always @ (posedge clock)
begin

      if(x)
            state <= state + 1;
      else
            state <= state - 1;

      z2 <= x & state[1] | ~x & ~state[0] & ~state[0];

end

endmodule
```



Problem #2.

In this problem, you are to design a portion of the controller for a high definition, hard disk, digital video recorder.

The controller receives four inputs from the remote control:

Play/Pause (PP),
- When playing (1X), causes video to be frozen
- Otherwise, causes playing to resume

Fast Forward (FF)
- Causes video to fast forward at 2X speed
- When fast forwarding, causes video to fast forward at 4X speed

Rewind (RW)
- Causes video to rewind at 2X speed
- When rewinding, causes video to rewind at 4X speed

Live (LV)
- Sets video source to live television

You can assume that only one button can be pressed at any time. You can also assume that if the hard disk is rewound to the beginning, it will automatically begin playing from that point and if the hard disk is fast forwarded to the end, it will revert to playing live television.

There are four output bits from the controller:

Play from disk (PFD)
- 0 = live television, 1 = hard disk

Mode (MOD1, MOD0)
- 0 = continuous (1X speed), 1 = 2X speed, 2 = 4X speed, 3 = freeze

Direction (DIR)
- 0 = forward, 1 = reverse

The 2X speed and 4X speed outputs are used for regular speed and high speed fast forward and fast reverse playing from the hard disk.

Note that only a small number of input and output combinations are actually valid. For instance, although there are four buttons on the remote control, there are only 5 (not 16) valid input combinations since, at most, one button can be active at any time. On the output side, for example, it is not possible to watch live television in reverse at high speed (viewing in reverse must come from video stored on the hard disk).

1. (10 points) Identify and describe all valid (and invalid) output combinations.

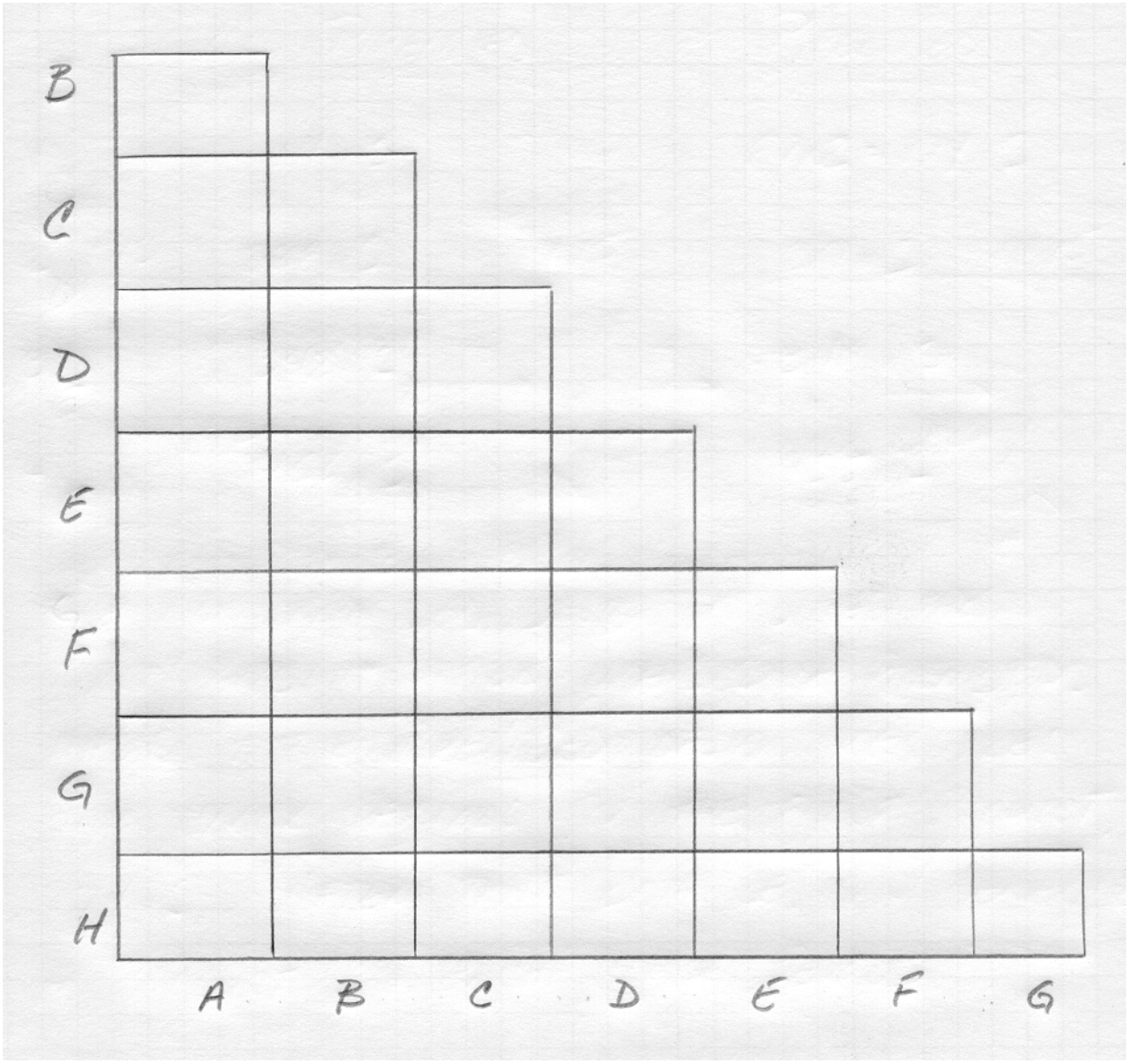
2. (15 points) Based on the valid output combinations identified in part 1 above and all possible input combinations, construct a ***state table*** for the Moore machine implementing the controller. Use symbolic state names such as FRZ, FF2X, PLV, etc. on the state table and be sure the operation of your machine is clear.

Problem #3.

1. (10 points) For the state table given below, complete the implication table on the following page, illustrating its contents after the first and second passes. Do not include same state pairs or self-implied pairs in the implication table.

NS, input = xy

PS	00	01	11	10	z
A	B	C	A	A	0
B	F	D	B	C	1
C	D	D	G	G	1
D	C	B	C	B	0
E	F	D	B	C	1
F	E	C	G	F	0
G	B	C	F	G	0
H	C	E	C	E	0


B
C
D
E
F
G
H
A
B
C
D
E
F
G

2. (5 points) Identify all (1) same states and (2) equivalent states and construct the reduced (simplified) state diagram.

3. (10 points) Verify your answer above by finding the equivalence partition using the Moore reduction procedure.

Problem #4.

The truth table for the multiplication of one-bit binary numbers is shown below:

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

Obviously, the truth table above is also that of a 2-input AND gate.

Like the multiplication of decimal numbers, multiplication of multiple-bit, unsigned binary numbers can be accomplished by generating the necessary partial products and adding them as shown below for the two-bit case.

```
                X2    X1
          x     Y2    Y1
          ----------------
                X2Y1  X1Y1
+       X2Y2    X1Y2
-------------------------------
P4      P3      P2      P1
```

The multiplication of 2, 2-bit binary numbers results in a 4-bit product. Product bit P1 is partial product X1Y1 (as in the truth table above). Product bit P2 is the sum of partial products X2Y1 and X1Y2. Product bit P3 is the sum of partial product X2Y2 and any carry generated in the creation of P2. Product bit P4 is simply any carry generated in the creation of P3.

This type of binary multiplier is know as an “array multiplier” and can be expanded to any number of bits.

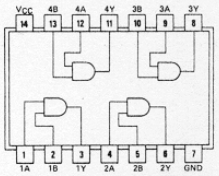
1. (5 points) Partial data sheets for the 7408 quad, 2-input AND gate and 7482 2-bit, binary full adder are given below. Using only these two devices (as many as necessary), construct the schematic diagram for a 2-bit array multiplier. Show all signal connections (i.e., no “floating inputs”).

7408 Data Sheet

QUADRUPLE 2-INPUT
POSITIVE-AND GATES

08

positive logic:
Y = AB



SN5408 (J, W) SN7408 (J, N)
SN54LS08 (J, W) SN74LS08 (J, N)
SN54S08 (J, W) SN74S08 (J, N)

See page 6-10

switching characteristics at V_{CC} = 5 V, T_A = 25°C

TYPE	TEST CONDITIONS#	t_{PLH} (ns) Propagation delay time, low-to-high-level output			t_{PHL} (ns) Propagation delay time, high-to-low-level output		
		MIN	TYP	MAX	MIN	TYP	MAX
'08	C_L = 15 pF, R_L = 400 Ω		17.5	27		12	19
'H11, 'H21	C_L = 25 pF, R_L = 280 Ω		7.6	12		8.8	12
'LS08, 'LS11 'LS21	C_L = 15 pF, R_L = 2 kΩ		8	15		10	20
'S08, 'S11	C_L = 15 pF, R_L = 280 Ω		4.5	7		5	7.5
	C_L = 50 pF, R_L = 280 Ω		6			7.5	

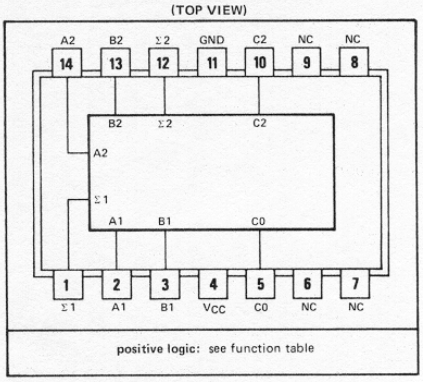
#Load circuit and voltage waveforms are shown on pages 3-10 and 3-11.

7482 Data Sheet

TYPES SN5482, SN7482
2-BIT BINARY FULL ADDERS

BULLETIN NO. DL-S 7211836, DECEMBER 1972

SN5482 . . . J OR W PACKAGE
SN7482 . . . J OR N PACKAGE
(TOP VIEW)



NC—No internal connection

FUNCTION TABLE

INPUTS				OUTPUTS					
				WHEN C0 = L			WHEN C0 = H		
A1	B1	A2	B2	Σ1	Σ2	C2	Σ1	Σ2	C2
L	L	L	L	L	L	L	H	L	L
H	L	L	L	H	L	L	L	H	L
L	H	L	L	H	L	L	L	H	L
H	H	L	L	L	H	L	H	H	L
L	L	H	L	L	H	L	H	H	L
H	L	H	L	H	H	L	L	L	H
L	H	H	L	H	H	L	L	L	H
H	H	H	L	L	L	H	H	L	H
L	L	L	H	L	H	L	H	H	L
H	L	L	H	H	H	L	L	L	H
L	H	L	H	H	H	L	L	L	H
H	H	L	H	L	L	H	H	L	H
L	L	H	H	L	L	H	H	L	H
H	L	H	H	H	L	H	L	H	H
L	H	H	H	H	L	H	L	H	H
H	H	H	H	L	H	H	H	H	H

H = high level, L = low level

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see note 4)

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	C0	Σ1	C_L = 15 pF, R_L = 400 Ω			34	ns
t_{PHL}						40	
t_{PLH}	B2	Σ2				40	ns
t_{PHL}						35	
t_{PLH}	C0	Σ2				38	ns
t_{PHL}						42	
t_{PLH}	C0	C2	C_L = 15 pF, R_L = 780 Ω		12	19	ns
t_{PHL}					17	27	

¶t_{PLH} ≡ propagation delay time, low-to-high-level output
t_{PHL} ≡ propagation delay time, high-to-low-level output
NOTE 4: Load circuit and voltage waveforms are shown on page 3-10.

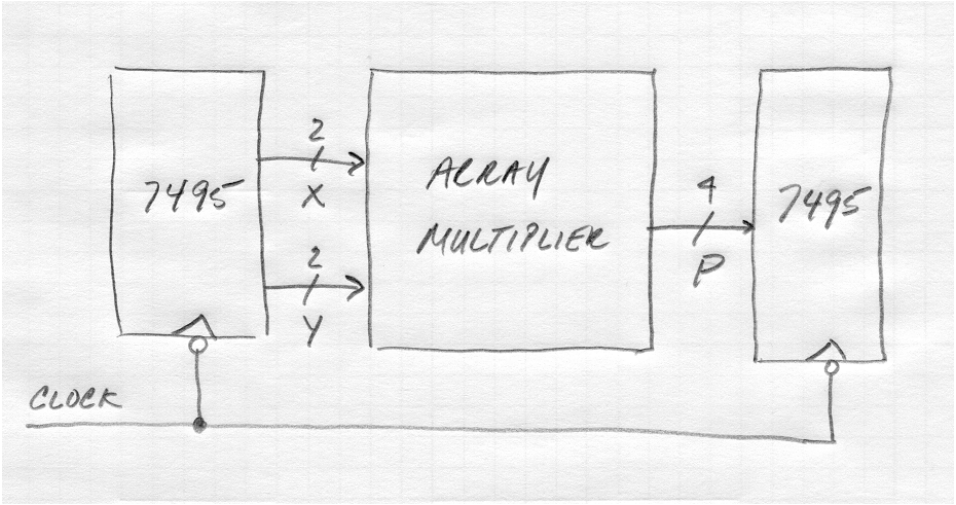
2. (5 points) What is the critical path and maximum propagation delay through the array multiplier and what input conditions are necessary to produce the maximum propagation delay?

   Use the specifications for the type ’08 device (not the ‘LS08 or ‘S08) and note that the 7482 delays from A2 to Sum2 are the same as the B2 to Sum2 delays given on the data sheet.

3. (5 points) We now want to add registers for the inputs and outputs of the array multiplier. On each falling clock edge, two new operands will be loaded on the input side and the result of the previous multiplication will be loaded on the output side. Partial specs for the 7495, 4-bit shift register are given on the following page.

   As is often the case, the register has more functionality than we need (the shift capability) so it will be configured to operate in parallel input/output mode only. The mode control input will be held high making the SERIAL INPUT and CLOCK1 inputs don't cares. On each falling CLOCK2 edge, inputs A through D will be loaded into the register.

   If 7495 registers are added to the inputs and outputs as shown below, what is the minimum clock period necessary to insure correct operation?

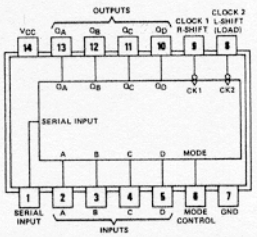


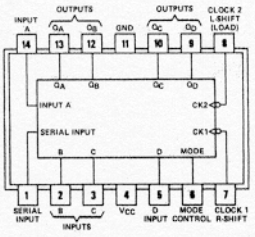
7495 Data Sheet

4-BIT SHIFT REGISTERS

95 PARALLEL IN/PARALLEL OUT
SHIFT RIGHT, SHIFT LEFT
SERIAL INPUT

See Page 7-89





SN5495A (J, W) SN7495A (J, N)
SN54LS95B (J, W) SN74LS95B (J, N)

SN54L95 (J, T) SN74L95 (J, N)

FUNCTION TABLE

INPUTS								OUTPUTS			
MODE CONTROL	CLOCKS 2 (L)	CLOCKS 1 (R)	SERIAL	PARALLEL A	PARALLEL B	PARALLEL C	PARALLEL D	Q_A	Q_B	Q_C	Q_D
H	H	X	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}
H	↓	X	X	a	b	c	d	a	b	c	d

REVISED MARCH 1974

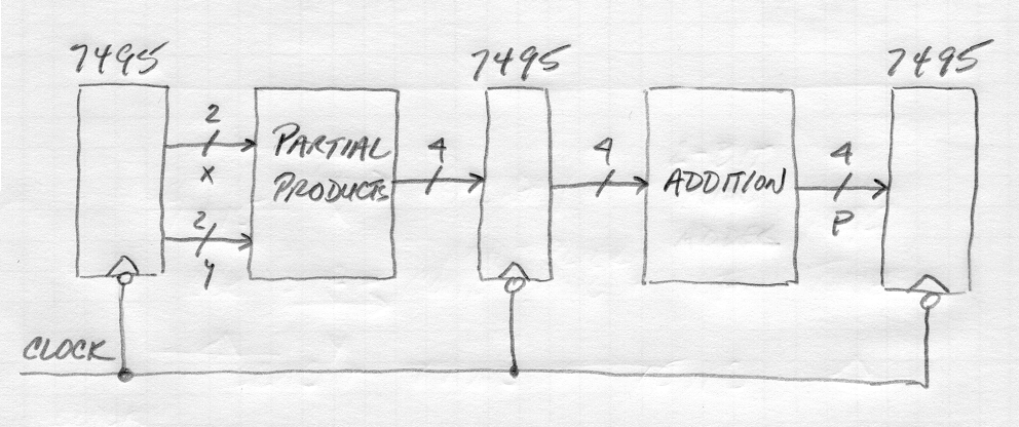
recommended operating conditions

	SN5495A MIN	SN5495A NOM	SN5495A MAX	SN7495A MIN	SN7495A NOM	SN7495A MAX	UNIT
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−800			−800	μA
Low-level output current, I_{OL}			16			16	mA
Clock frequency, f_{clock}	0		25	0		25	MHz
Width of clock pulse, $t_{w(clock)}$ (see Figure 1)	20			20			ns
Setup time, high-level or low-level data, t_{su} (see Figure 1)	15			15			ns
Hold time, high-level or low-level data, t_h (see Figure 1)	0			0			ns
Time to enable clock 1, $t_{enable\ 1}$ (see Figure 2)	15			15			ns
Time to enable clock 2, $t_{enable\ 2}$ (see Figure 2)	15			15			ns
Time to inhibit clock 1, $t_{inhibit\ 1}$ (see Figure 2)	5			5			ns
Time to inhibit clock 2, $t_{inhibit\ 2}$ (see Figure 2)	5			5			ns
Operating free-air temperature, T_A	−55		125	0		70	°C

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{max}	Maximum clock frequency	C_L = 15 pF, R_L = 400 Ω, See Figure 1	25	36		MHz
t_{PLH}	Propagation delay time, low-to-high-level output from clock			18	27	ns
t_{PHL}	Propagation delay time, high-to-low-level output from clock			21	32	ns

4. (5 points) One method of increasing clock frequency (at the cost of latency) is via pipelining. In a pipelined design, the combinational processing is broken up into pipeline stages with registers between each of the stages. For our array multiplier, we can break up the multiplication into partial product generation and addition as shown in the generic block diagram below:



What is the minimum clock period for the 2-stage pipelined implementation of the synchronous, array multiplier?

5. (5 points) For the implementations in parts 3 and 4 above, assume that on the first falling clock edge the first two operands are loaded, on the second falling clock edge the second operands are loaded, on the third the third, etc.

   If the first two operands are loaded at time = 0, when is the first product available for (a) the non-pipelined version and (b) the pipelined version?

   At what point (after how many products are generated) does the pipelined design yield a performance advantage (more products in less time)?