Spring 2009 Name: Mark F

Computer Organization Test 1

Question 1. (25 points)

a) Convert 174_{10} to a binary (base 2) value.

4

174 − 128 = 46   46 − 32 = 14

256	128	64	32	16	8	4	2	1
0	1	0	1	0	1	1	1	0_2

b) Convert 174_{10} to a hexadecimal (base 16) value.

4   A E$_{16}$

c) Convert -174_{10} to a two's complement value.

25) 3

```
      1 0 1 0 1 0 0 0 1
  +1                  +1
  1 1 1 0 1 0 1 0 0 1 0
```

d) Perform the following arithmetic operations:

```
  1010011₂       1100110₂        A 8 7 D 4₁₆      E 9 D 3 A₁₆
 +1100110₂     - 0101011₂      + 7 3 A 1 8₁₆    - B 9 C 4 B₁₆
 10111001₂        111011₂       1 1 C 1 E C₁₆    3 0 0 E F₁₆
    4               3                4                3
```

A	10
B	11
C	12
D	13
E	14
F	15

Question 2. (15 points)

a) Convert the value 55.375_{10} to its binary representation.

4

64	32	16	8	4	2	1	.	.5	.25	.125	.0625
0	1	1	0	1	1	1	.	0	1	1	0

b) Normalize the above value so that the most significant 1 is immediately to the left of the radix point. Include the corresponding exponent value to indicate the motion of the radix point.

3    $1.\boxed{101110110} \times 2^{\boxed{5}}$ +127 = 132

c) Write the corresponding 32-bit IEEE 754 floating point representation for 55.375_{10}.

(15) 4

Sign bit	8-bit Exponent (bias 127)	23-bit Mantissa (for normalized values, leading 1 not stored)
0	10000100	1011101100 ... 0

128 64 32 16 8 4 2 1

d) Explain why the real value of 0.1_{10} cannot be stored exactly as in a 32-bit IEEE 754 floating point variable.

4   Decimal values are the sum of fractional powers of two which don't happen to sum be 0.1.

40

Name: ______________________

Question 3. (15 points) For the Boolean function F specified in the following truth table use Boolean identities write a simplified sum-of-products (SOP) Boolean function.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

+6

Identity Name	AND Form	OR Form
Identity Law	$1x=x$	$0+x=x$
Null (or Dominance) Law	$0x=0$	$1+x=1$
Idempotent Law	$xx=x$	$x+x=x$
Inverse Law	$x\bar{x}=0$	$x+\bar{x}=1$
Commutative Law	$xy=yx$	$x+y=y+x$
Associative Law	$(xy)z=x(yz)$	$(x+y)+z=x+(y+z)$
Distributive Law	$x+yz=(x+y)(x+z)$	$x(y+z)=xy+xz$
Absorption Law	$x(x+y)=x$	$x+xy=x$
DeMorgan's Law	$(\overline{xy})=\bar{x}+\bar{y}$	$(\overline{x+y})=\bar{x}\bar{y}$
Double Complement Law	$\bar{\bar{x}}=x$	

(+2) $F = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}BC\bar{D} + A\bar{B}\bar{C}\bar{D} + AB\bar{C}\bar{D}$

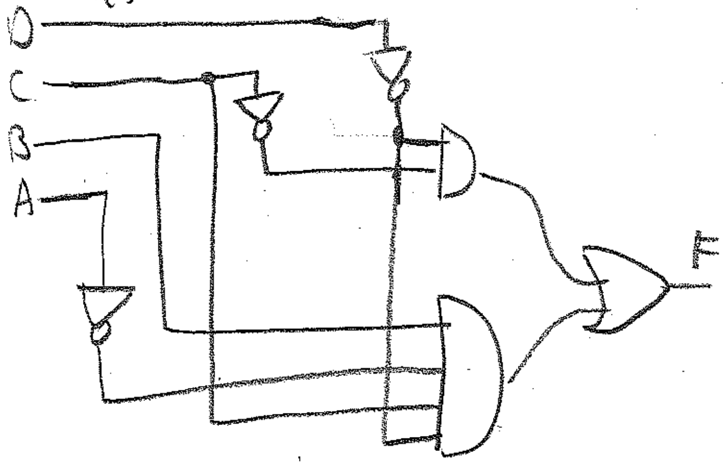
$= \bar{A}\bar{C}\bar{D}(\bar{B}+B) + \bar{A}BC\bar{D} + A\bar{C}\bar{D}(\bar{B}+B)$

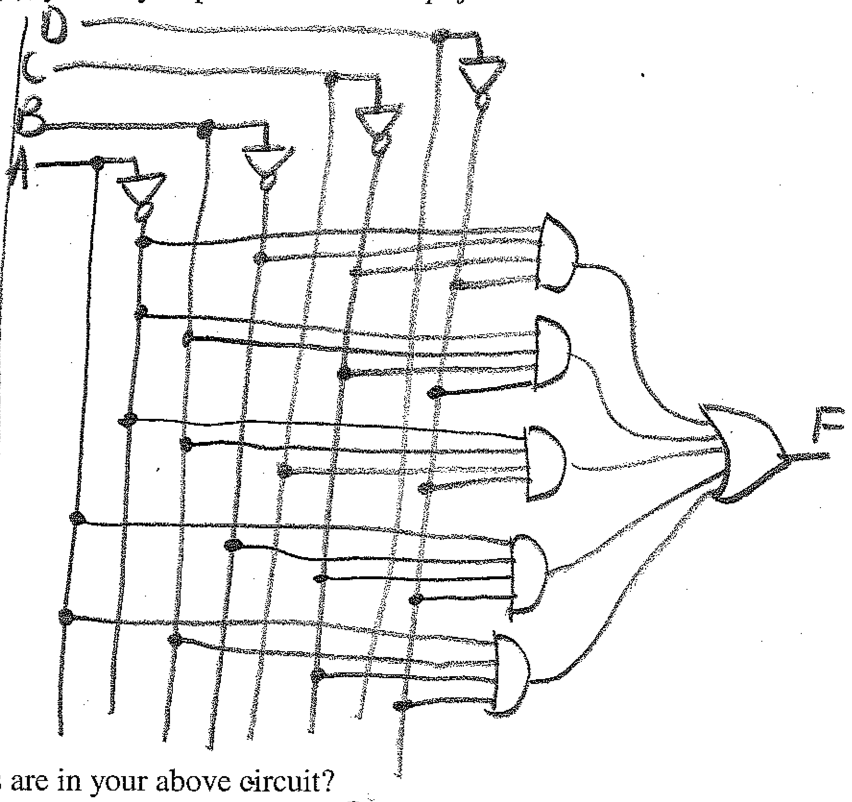
$= \bar{C}\bar{D}(\bar{A}+A) + \bar{A}BC\bar{D}$

$= \bar{C}\bar{D} + \bar{A}BC\bar{D}$

b) Implement your above simplified Boolean function for F using AND, OR, and NOT gates. (Alternatively, if you had problems with part (a), you may implement the *unsimplified* Boolean function F).

+5





+2 c) Ignoring NOT gates, how many gate delays are in your above circuit?

2 2

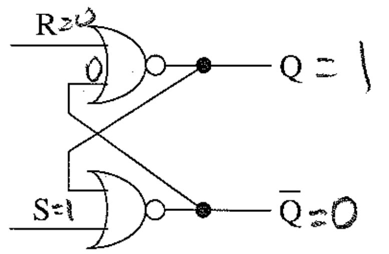
+2 d) What is the complexity (# of gates + # of inputs to those gates) of your circuit?

$3+8=11$ $6+25=31$

15

Question 4. (10 points) a) If R = 0 and S = 1, then what will be the output on Q and $\overline{Q}$?

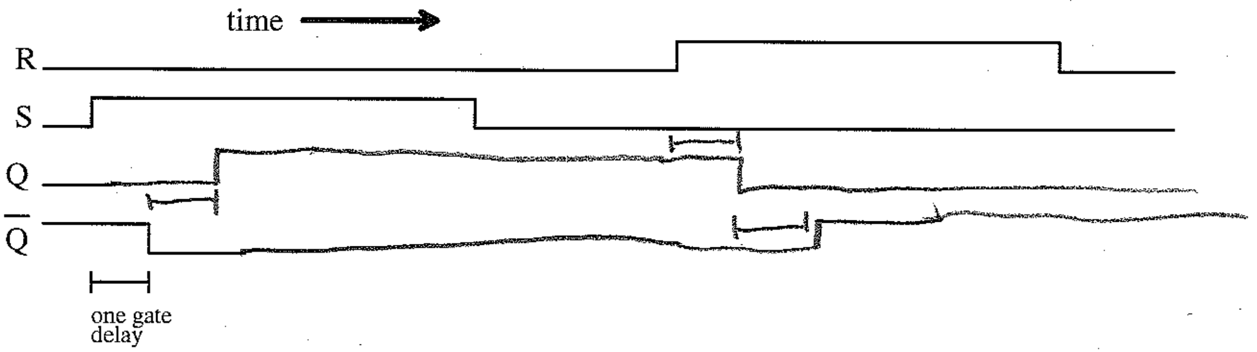
3



3 b) Now, if S goes to a 0 value, what happens to the output on Q and $\overline{Q}$? remain the same

c) Complete the following timing diagram for the SR latch. Include gate delays in the diagram.

4



Question 5. (10 points) If we consider implementing a 32-bit adder by "rippling" together smaller adders, then we get the following gate delays:

Type of Adders Used	Number of Adders Needed	Gate Delay per Adder	Total Gate Delays for 32-bit Adder
one-bit adders	32	2	64
two-bit adders	16	2	32
four-bit adders	8	3	24

a) Explain why the gate delay of a two-bit adder is the same as the gate delay for a one-bit adder.
The SOP for the one-bit adder has 3 product/ANDs, and the SOP for the two-bit adder has 7 products/ANDs. The AND outputs go into an OR gate in either case. Since an OR can handle up to 9-inputs, both adders only need one gate delay for ORing.

b) Explain why the gate delay of a four-bit adder is more than the gate delay for a two-bit adder.
The four-bit adder has 31 products/ANDs, so to OR the output of the ANDs takes 2 gate delays.

10

Question 6. (10 points) Suppose we have a register file with the following specifications:

- 8 registers numbered from R0 to R7
- each register has 32-bits
- one write ports
- two read ports

a) How many bits(/"wires") would be need for each of the following?

- specifying the register number for either a read or write port 3 bits
- data output read from a read port 32-bits

b) How many decoders would be needed in the implementation of the whole register file? one

c) What type of decoders (i.e., number of inputs and number of outputs for each decoder) are needed? 3-to-8 decoder

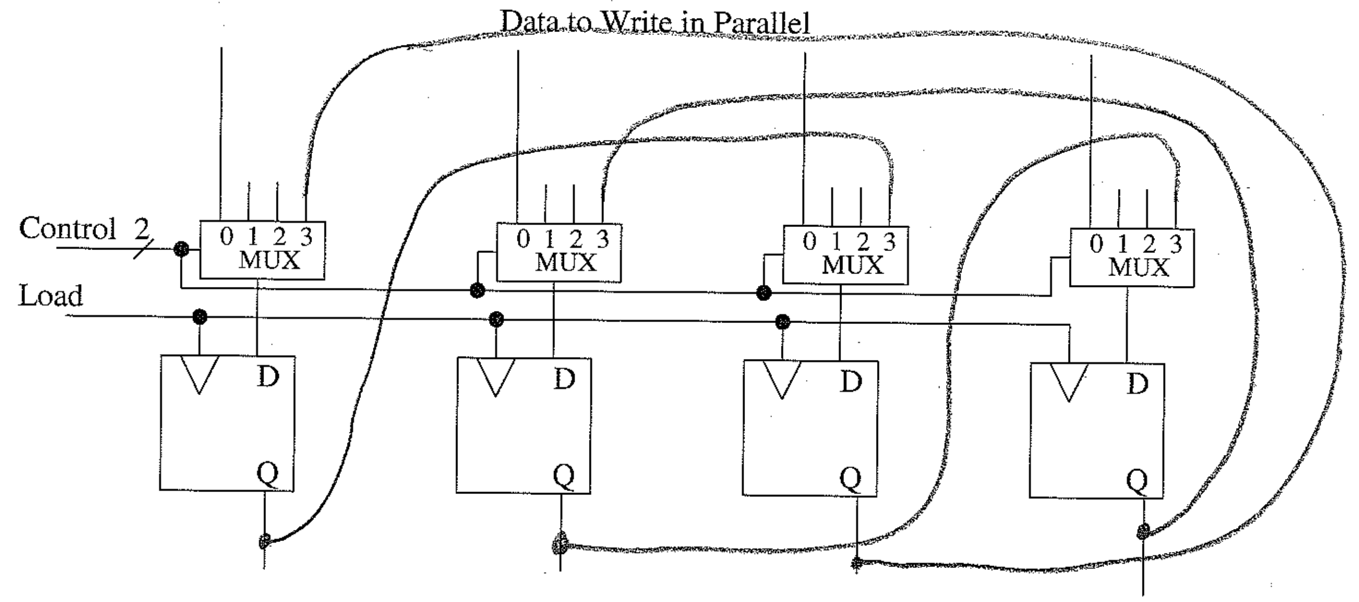
d) How many multiplexers would be needed in the implementation of the whole register file? 64 Muxs

e) What type of multiplexers are needed? 8-to-1 Muxs

Question 7. (10 points) Determine the Hamming codeword if the 8-bits of data (D_7 to D_0) are $1101\ 0110_2$, i.e., what are the values of the four even-parity bits (P_8, P_4, P_2, and P_1) to allow for one-bit error detection and correction.

12	11	10	9	8	7	6	5	4	3	2	1
D_7	D_6	D_5	D_4	P_8	D_3	D_2	D_1	P_4	D_0	P_2	P_1
1	1	0	1	1	0	1	1	1	0	0	1
4+8	1+2+8	2+8	1+8	8	1+2+4	2+4	1+4	4	1+2	2	1

Question 8. (5 points) Suppose that we wanted to use the control value of 3_{10} to do a circular shift right by **TWO** bit positions. Show how to connect the wires in the below diagram to do this.



25