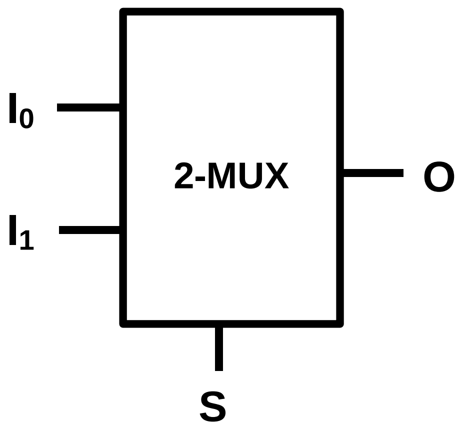
Computer Organization

Fall 2000

Test #1 Answers

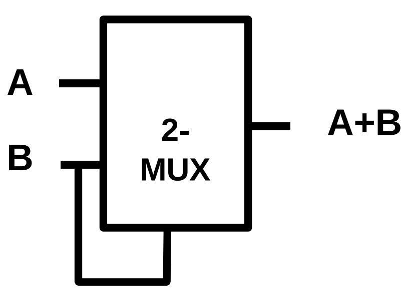
Question 1 (15 *pts*): Prove that a 2 input multiplexor is a universal operator (*functionally complete*). That is – that using only a 2-input multiplexor we can build any Boolean function.

To prove this we can show how to implement AND, OR and NOT using only a 2-input decoder.

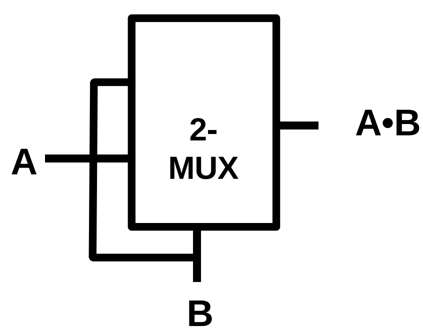


I_0	I_1	S	O
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

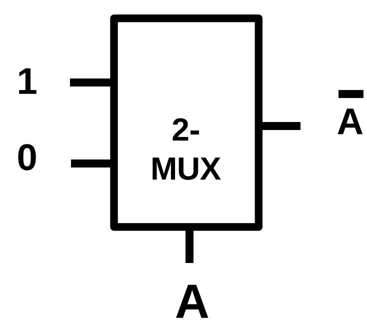
OR



AND



NOT



Question 2 (20 *pts*): Addition of 2 bits (interpreted as unsigned integers) can be defined by showing the result of all 4 possible 2 bit addition operations:

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

5 pts): Show a truth table for 2 bit addition.

```
      A
 +    B
 -------
 S0   S1
```

A	*B*	S_0	S_1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

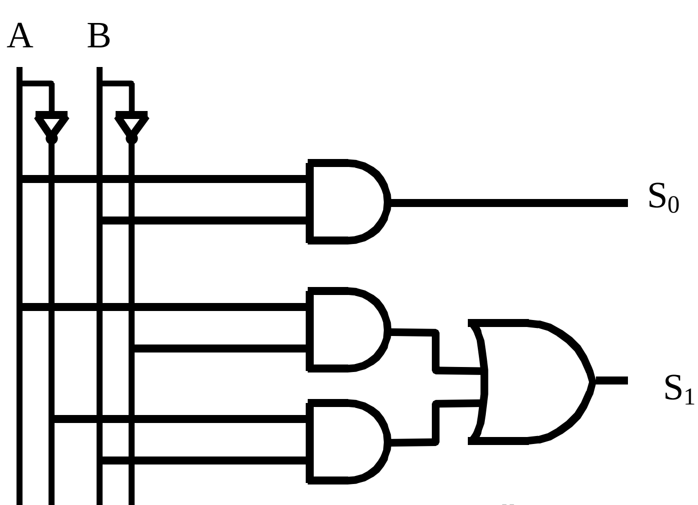
5 pts): Show the Boolean expressions based on Sum of Products form.

$$S_0 = A \bullet B$$
$$S_0 = A \bullet \overline{B} + \overline{A} \bullet B$$

5 pts): Show the Boolean expressions based on Product of Sums form.

$$S_0 = (A + B) \bullet (A + \overline{B}) \bullet (\overline{A} + B)$$
$$S_1 = (A + B) \bullet (\overline{A} + \overline{B})$$

5 pts): Draw a logic diagram that implements 2 bit addition.



Question 3 (20 *pts*): True/False - 2 *pts* each. No explanation needed (or graded)!

A 400MHz computer is always faster than a 300MHz computer.	TRUE	(FALSE)
A *benchmark* is a comparison of the performance of computers.	TRUE	(FALSE)
The Instruction Count (IC) for a program depends on the instruction set and on the compiler used.	(TRUE)	FALSE
Anything that can be expressed with Boolean Algebra can be built using only OR gates.	TRUE	(FALSE)
A Multiplexor with 64 inputs requires 7 selection inputs.	TRUE	(FALSE)
Performance depends on CPI.	(TRUE)	FALSE
SPEC95 is a benchmark for Windows95	TRUE	(FALSE)
A benchmark summary based on normalized times should be based on an arithmetic mean.	TRUE	(FALSE)
CPI (cycles per instruction) can depend on the compiler used.	(TRUE)	FALSE
A 1 bit memory can be built using combinational logic.	TRUE	(FALSE)
OR gates are named for Bill Gates.	TRUE	(FALSE)

Question 4 (15 *pts*): We have two implementations of the same instruction set. Machine A has a clock cycle time of 2ns and Machine B has a 200MHz clock.. We have a program named `napinclass` that has an instruction count of 22,313 on Machine A. On machine A the program `napinclass` averages 4.5 cycles per instruction and machine B averages 3 cycles per instruction.

Part 1 (8 *pts*): If possible, fill in the blanks in the following sentence (and show all your work!). If it is not possible to fill in the blanks, state what additional information is required.

Machine _________ is _________ times faster than _________ when running program `napinclass`.

*Execution Time on A is: IC * 4.5 * 2ns = IC * 9 ns*

*Execution Time on B is: IC * 3 * 5ns = IC * 15 ns*

Perf A/Perf B = 15/9 = 5/3 = 1.67

Machine A is 1.67 times faster than Machine B

Part 2 (7 *pts*): When we run Machines A and B on the programs `blah` and `foo` we get the following execution times (in seconds):

	blah	foo
Machine A	50	5
Machine B	25	10

The same compiler is used in all cases. Explain how it is possible that A can be faster on one program and B can be faster on the other program. Be as precise as possible!

*Execution Time = IC * CPI * Clock cycle.*

The only thing that can vary is CPI. Since CPI is an average over the instructions in a program, it can be different for different programs. Specifically the instruction mix can be different, so that program blah might include lots of instructions that are slower on A than on B, and foo might include lots of instructions that are slower on B than on A.

Question 5 (15 *pts*): The following 2 Boolean expressions represent the same Boolean function (one is SOP the other is POS). Prove that these represent the same Boolean function (show that *f* = *f'*) using algebraic manipulation (use the Laws of Boolean Algebra to convert one expression in to the other). You must show your work, including a list of the Law(s) used at each step in your proof, a list is shown at the bottom of this page.

$$f(A,B) = A \bullet B + \overline{A} \bullet \overline{B}$$

$$f'(A,B) = \left(A + \overline{B}\right) \bullet \left(\overline{A} + B\right)$$

$$A \bullet B + \overline{A} \bullet \overline{B} = \left(A + \left(\overline{A} \bullet \overline{B}\right)\right) \bullet \left(B + \left(\overline{A} \bullet \overline{B}\right)\right) \quad \textit{Distributive (twice)}$$

$$= \left(\left(A + \overline{A}\right) \bullet \left(A + \overline{B}\right)\right) \bullet \left(B + \left(\overline{A} \bullet \overline{B}\right)\right) \quad \textit{Distributive}$$

$$= \left(\left(A + \overline{A}\right) \bullet \left(A + \overline{B}\right)\right) \bullet \left(\left(B + \overline{A}\right) \bullet \left(B + \overline{B}\right)\right) \quad \textit{Distributive}$$

$$= \left(1 \bullet \left(A + \overline{B}\right)\right) \bullet \left(\left(B + \overline{A}\right) \bullet 1\right) \quad \textit{Inverse}$$

$$= \left(A + \overline{B}\right) \bullet \left(B + \overline{A}\right) \quad \textit{Identity}$$

Boolean Algebra Laws and Identities:

Identity:	$A \cdot 1 = A$	$A + 0 = A$
Commutative:	$A \cdot B = B \cdot A$	$A + B = B + A$
Inverse:	$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$
Distributive:	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
Zero and Ones:	$A \cdot 0 = 0$	$A + 1 = 1$
Associative:	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
DeMorgan's:	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$

Question 6 (15 *pts*): Explain why MIPS (*million instructions per second*) is not a good performance measure. Give as many reasons as you can (don't stop at one)!

Does not take in to account how much work an instruction does.

Can vary between 2 programs on the same computer.

Can vary inversely with performance!

Extra Credit: What doe PRCVD stand for?