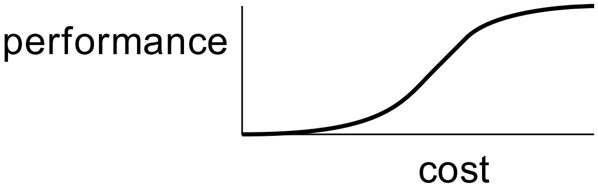
University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS150, Spring 2010

Homework 1 Solutions: Combinational Logic and MIPS Review

1. At the "high-end" of performance, such as for supercomputers, performance comes at a very high cost. At the "low-end", say for embedded processors, low-cost comes with the expense of very low performance. In the middle, for instance desktop and laptop computers, the tradeoff is more balanced.



2. DDCA problem 1.2:

   (a) Automobile designers use hierarchy to construct a car from major assemblies such as the engine, body, and suspension. The assemblies are constructed from subassemblies; for example, the engine contains cylinders, fuel injectors, the ignition system, and the drive shaft. Modularity allows components to be swapped without redesigning the rest of the car; for example, the seats can be cloth, leather, or leather with a built in heater depending on the model of the vehicle, so long as they all mount to the body in the same place. Regularity involves the use of interchangeable parts and the sharing of parts between different vehicles; a 65R14 tire can be used on many different cars.

   (b) Businesses use hierarchy in their organization chart. An employee reports to a manager, who reports to a general manager who reports to a vice president who reports to the president. Modularity includes well-defined interfaces between divisions. The salesperson who spills a coke in his laptop calls a single number for technical support and does not need to know the detailed organization of the information systems department. Regularity includes the use of standard procedures. Accountants follow a well-defined set of rules to calculate profit and loss so that the finances of each division can be combined to determine the finances of the company and so that the finances of the company can be reported to investors who can make a straightforward comparison with other companies.

3. DDCA problem 1.4: An accuracy of +/- 50 mV indicates that the signal can be resolved to 100 mV intervals. There are 50 such intervals in the range of 0-5 volts, so the signal represents $\log_2 50$ = 5.64 bits of information.

4. DDCA problem 1.49:

A	*B*	*C*	*Y*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

5. DDCA problem 1.50:

A	*B*	*C*	*Y*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

6. DDCA problem 1.51:

A	*B*	*C*	*Y*
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

7. DDCA problem 1.52:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	0

Zero

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

A NOR B

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	0

$\overline{\text{A}}\text{B}$

A	B	Y
0	0	1
0	1	1
1	0	0
1	1	0

NOT A

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	0

$\text{A}\overline{\text{B}}$

A	B	Y
0	0	1
0	1	0
1	0	1
1	1	0

NOT B

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

XOR

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NAND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

XNOR

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	1

B

A	B	Y
0	0	1
0	1	1
1	0	0
1	1	1

$\overline{\text{A}} + \text{B}$

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	1

A

A	B	Y
0	0	1
0	1	0
1	0	1
1	1	1

$\text{A} + \overline{\text{B}}$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

OR

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	1

One

8. DDCA problem 1.53: 2^{2^N}

9. DDCA problem 1.54: $V_I L$ = 2.5; $V_I H$ = 3; $V_O L$ = 1.5; $V_O H$ = 4; NM_L = 1; NM_H = 1

10. DDCA problem 1.55: No, there is no legal set of logic levels. The slope of the transfer characteristic never is better than -1, so the system never has any gain to compensate for noise.

11. DDCA problem 1.58: (a) AND gate; (b) $V_I L$ = 1.5; $V_I H$ = 2.75; $V_O L$ = 0; $V_O H$ = 3

12. DDCA problem 1.59: (a) XOR gate; (b) $V_I L$ = 1.25; $V_I H$ = 2; $V_O L$ = 0; $V_O H$ = 3

13. DDCA problem 6.2: Yes, it is possible to design a computer architecture without a register set. For example, an architecture could use memory as a very large register set. Each instruction would require a memory access. For example, an add instruction might look like this:

    ```
    add 0x10, 0x20, 0x24
    ```

    This would add the values stored at memory addresses 0x20 and 0x24 and place the result in memory address 0x10. Other instructions would follow the same pattern, accessing memory instead of registers. Some advantages of the architecture are that it would require fewer instructions. Load and store operations are now unnecessary. This would make the decoding hardware simpler and faster.

    Some disadvantages of this architecture over the MIPS architecture is that each operation would require a memory access. Thus, either the processor would need to be slow or the memory small.

Also, because the instructions must encode memory addresses instead of register numbers, the instruction size would be large in order to access all memory addresses. Or, alternatively, each instruction can only access a smaller number of memory addresses. For example, the architecture might require that one of the source operands is also a destination operand, reducing the number of memory addresses that must be encoded.

14. DDCA problem 6.8:

    0x20100049
    0xad49fff9
    0x2f24822

15. DDCA problem 6.9:

    (a)

    ```
    addi $s0, 0, 73
    sw   $t1, -7($t2)
    ```

    (b)

    0x00000049 (`addi`)
    0xfffffff9 (`sw`)