

hw1

题目 1. 德·摩根律(DeMorgan's Law)的公式为

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

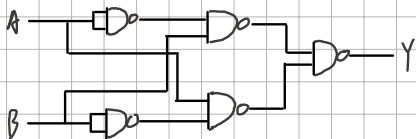
请证明这两个式子的正确性。根据德·摩根律，请仅使用二输入与非门搭建一个二输入异或门。(对于使用的与非门数量不做要求)

真值表如下:

A	B	A+B	$\overline{A+B}$	\overline{A}	\overline{B}	$\overline{A} \cdot \overline{B}$	A·B	$\overline{A \cdot B}$	$\overline{A} + \overline{B}$
0	0	0	1	1	1	1	0	1	1
0	1	1	0	1	0	0	0	1	1
1	0	1	0	0	1	0	0	1	1
1	1	1	0	0	0	0	1	0	0

故 $\overline{A+B} = \overline{A} \cdot \overline{B}$ $\overline{A \cdot B} = \overline{A} + \overline{B}$

搭建如下:



题目 2. 请将下面的数据按从小到大的顺序排序:

$(101001)_2$ $(52)_8$ $(00101001)_{BCD}$ $(33)_{16}$

$$(101001)_2 = (29)_{16}$$

$$(52)_8 = (101010)_2 = (2A)_{16}$$

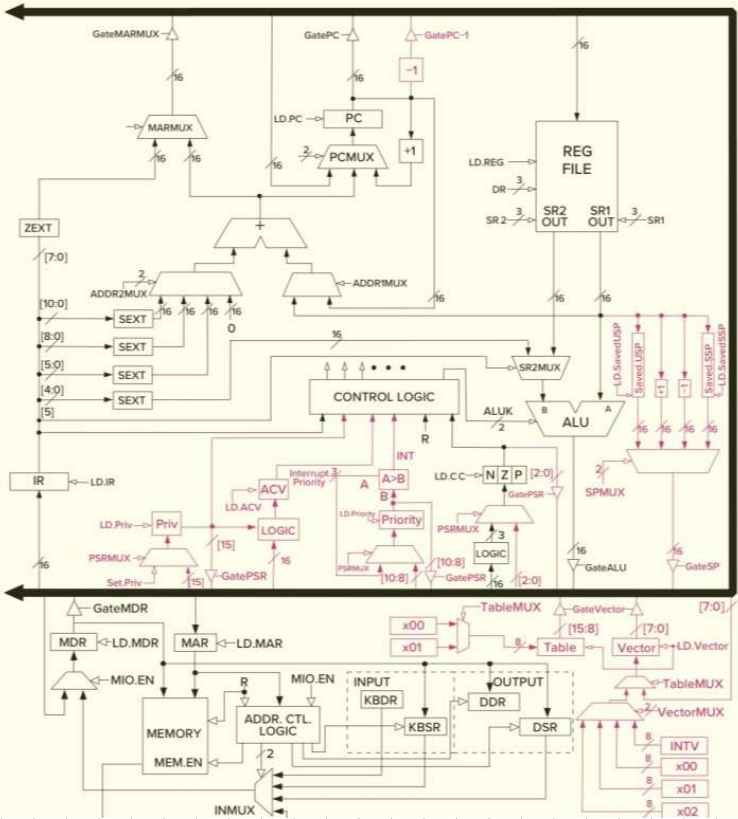
$$(00101001)_{BCD} = (1D)_{16}$$

$$\text{故 } (00101001)_{BCD} < (101001)_2 < (52)_8 < (33)_{16}$$

题目 3. LC-3 是一个简单的、入门级的架构。在先前的《计算机系统概论》课程中，我们已经对其有了基本的了解。一个典型的 LC-3 数据通路往往如下图所示:

可以看到，其中有一些比较明显的结构，例如 ALU、REG FILE、MUX、MEMORY 等。这些结构在 LC-3 的架构中发挥着重要作用。

- (1) 请结合自己所学的知识，查阅有关资料，简述上面四种结构的作用。
- (2) 考虑我们上课时提到的 CPU、内存等概念，这些概念分别对应了上面四种结构中的哪些呢？
- (3) 注意到图中下部有着 INPUT、OUTPUT 的内容，它们对应着计算机的 I/O 接口。请结合自己所学的知识，分别列举两种常用的输入设备和输出设备。



(1) ALU: 对AB进行计算并输出结果
 REGFILE: 读取SR1, SR2
 MUX: 选择数据
 MEMORY: 存储单元

(2) CPU: ALU MUX
 内存: REGFILE MEMORY
 (3) 键盘、鼠标; 显示器、音响

题目 4. 在时序逻辑电路中，always 赋值语句中的敏感变量不仅可以是触发信号的上升沿(posedge)，也可以是下降沿(negedge)。请编写 Verilog 程序，搭建一个 8bit 位宽的寄存器，要求所有触发器都应该由 clk 的下降沿(负边缘)触发。同时，寄存器采用高电平有效的同步复位方式，复位值为 0x12 而不是零。模块的部分信息如下所示：

```

module Reg (
  input clk ,
  input reset ,
  input [7:0] d ,
  output reg [7:0] q
);
// Write your code here
endmodule

```

请根据上面的内容补全 Verilog 代码，如果将同步复位改为异步复位，那么应当做出怎样的修改呢？

```
always@(negedge clk)
begin
    if(reset)
        begin
            a<=0x12;
        end
    else
        begin
            q<=d
        end
    end
end
```

异步复位: 条件改为 `always@(negedge clk, negedge reset)`

实验题 1. Verilog OJ 平台 <https://verilogoj.ustc.edu.cn/oj/> 上第 48 题【计数器】。在作业中

Welcome, 2202111026

提交结果

1/1 个通过测试用例, 获 10/10 分

状态: Accepted

提交时间: 几秒前

测试用例 0 (行为级仿真): Accepted

10 / 10 分 >

Code:

Copy

Download

```
1 module top_module (
2     input clk,
3     input reset, // 异步复位, 高电平有效, 复位值为0
4     output reg [3:0] q);
5     always@(posedge clk or posedge reset) begin
6         if(reset)
7             q <= 0;
8         else
9             q <= q + 1;
10    end
11 endmodule
```