

计算机组成原理

Homework 1

Made by TA

2024 年 3 月 11 日

注意

本次作业提交截止时间为 2024-03-25 14:00（下午上课前）。超过该时间的提交会被扣除一定的作业分数。规则为：

- 迟交一周以内：得分不超过满分的 80%；
- 迟交一周及以上：得分不超过满分的 50%。

请在 BB 系统的对应作业处提交。

修改说明

2024-03-11：修改第三题的相关表述。

题目 1.

德·摩根律 (DeMorgan's Law) 的公式为

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

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

- (1) 请证明这两个式子的正确性。
- (2) 请仅使用二输入或非门搭建一个二输入异或门。本小题对于使用的或非门数量不做要求。

- (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。

题目 3.

某程序在某计算机上会被编译成由 N 条指令组成的序列，其中加法指令占比 50%，乘法指令占比 40%，访存指令占比 10%。已知该计算机的 CPU 主频为 f (Hz)，并且在该计算机上，加法指令的 CPI 为 1，乘法指令的 CPI 为 2.5，访存指令的 CPI 为 5。请据此回答下面的问题。

- (1) 该程序在该计算机上的平均 CPI 是多少？
- (2) 该程序在该计算机上执行一次的时间是多少？
- (3) 如果只优化加法指令，使其在该计算机上的 CPI 变为 0.6，则该程序的运行时间降低为原来的多少？
- (4) 如果只优化乘法指令，令该程序的运行速度变得与 (3) 中一样，则需要使乘法指令在该计算机上的 CPI 变为多少？
- (5) 如果只优化访存指令，令该程序的运行速度变得与 (3) 中一样，则需要使访存指令在该计算机上的 CPI 变为多少？

题目 4.

假定我们需要实现一个位宽为 8、带有使能信号的寄存器。我们希望所有触发器都由 clk 的上升沿(正边缘)触发。此外，寄存器采用高电平有效的同步复位方式，复位值为 0x08 而不是零。该模块的部分实现如下所示：

```

module MyReg (
    input          [ 0 : 0]          clk ,
    input          [ 0 : 0]          rst ,
    input          [ 7 : 0]          d ,
    input          [ 0 : 0]          en ,

```

```

    output          reg    [ 7 : 0]          q
);
always @( <1> ) begin
    if ( rst )
        <2>
    else if ( en )
        <3>
end
endmodule

```

其中，d 为即将写入寄存器的数据，q 为寄存器中存储的数据。

- (1) 请根据上面的内容，在 <1>、<2>、<3> 处填上合适的代码。
- (2) 当复位信号 rst 和使能信号 en 同时为 1 时，寄存器会执行什么操作？
- (3) 如果将同步复位改为异步复位，代码应当做出怎样的修改？

题目 5.

阅读下面的 Verilog 程序，回答有关问题。

```

module Foo(A, B, Cin, Sum, Cout);
    input A, B, Cin;
    output Sum, Cout;
    reg Cout;
    reg T1, T2, T3;
    wire S1;

    xor X1(S1, A, B);

    always @(A or B or Cin) begin
        T1 = A & Cin;
        T2 = B & Cin;
        T3 = A & B;
    end
endmodule

```

```
Cout = (T1 | T2) | T3;  
end  
  
assign Sum = S1 ^ Cin;  
  
endmodule
```

- (1) 根据以上程序，将下面的真值表补充完整。
- (2) 根据真值表回答：上面的 Verilog 程序实现了什么功能？
- (3) 上面的 Verilog 程序包括了实例化的逻辑门、assign 赋值语句以及 always 赋值语句。请修改代码，使其只包含 always 赋值语句。

| Cin | A | B | Sum | Cout |
|-----|---|---|-----|------|
| 0 | 0 | 0 | | |
| 0 | 0 | 1 | | |
| 0 | 1 | 0 | | |
| 0 | 1 | 1 | | |
| 1 | 0 | 0 | | |
| 1 | 0 | 1 | | |
| 1 | 1 | 0 | | |
| 1 | 1 | 1 | | |