实验三 EDA 作业一

自 35 夏弘宇 2023011004

三、必做任务

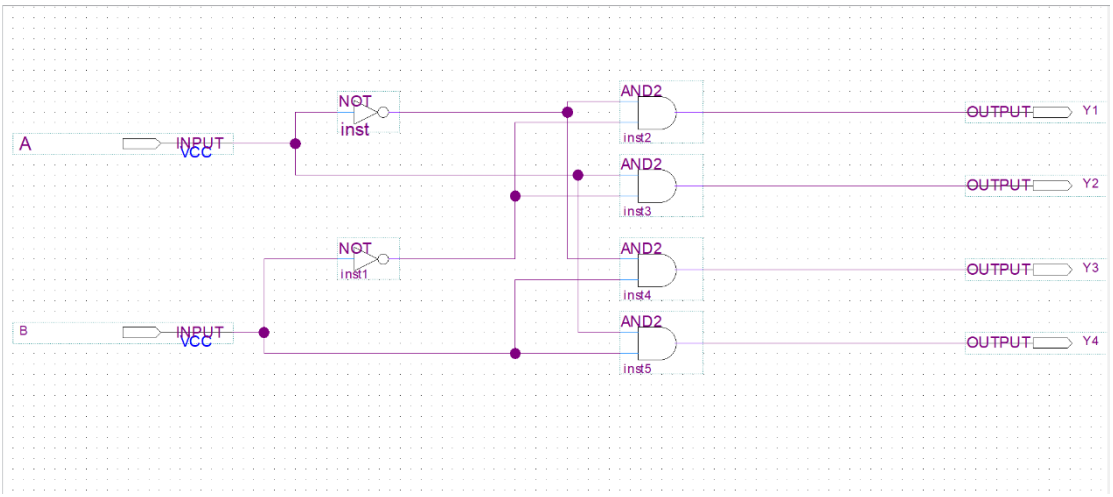


图 1 原理图

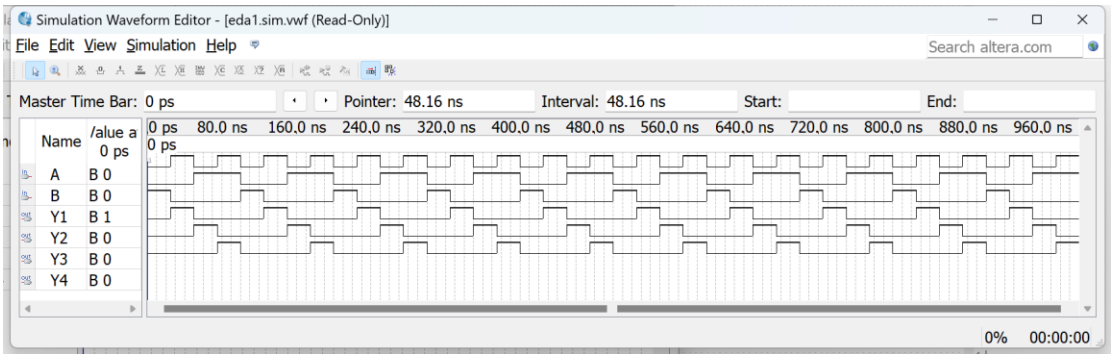


图 2 波形图

Node Name	Direction	Location	I/O Bank	REF Grou	er Locat	O Standa
A	Input	PIN_10	1	B1_N0	PIN_10	3.3...lt)
B	Input	PIN_11	1	B1_N0	PIN_11	3.3...lt)
Y1	Output	PIN_64	4	B4_N1	PIN_64	3.3...lt)
Y2	Output	PIN_63	4	B4_N1	PIN_63	3.3...lt)
Y3	Output	PIN_61	4	B4_N1	PIN_61	3.3...lt)
Y4	Output	PIN_60	4	B4_N1	PIN_60	3.3...lt)

图 3 引脚锁定

注：开关 3 对应 A 输入，开关 2 对应 B 输入

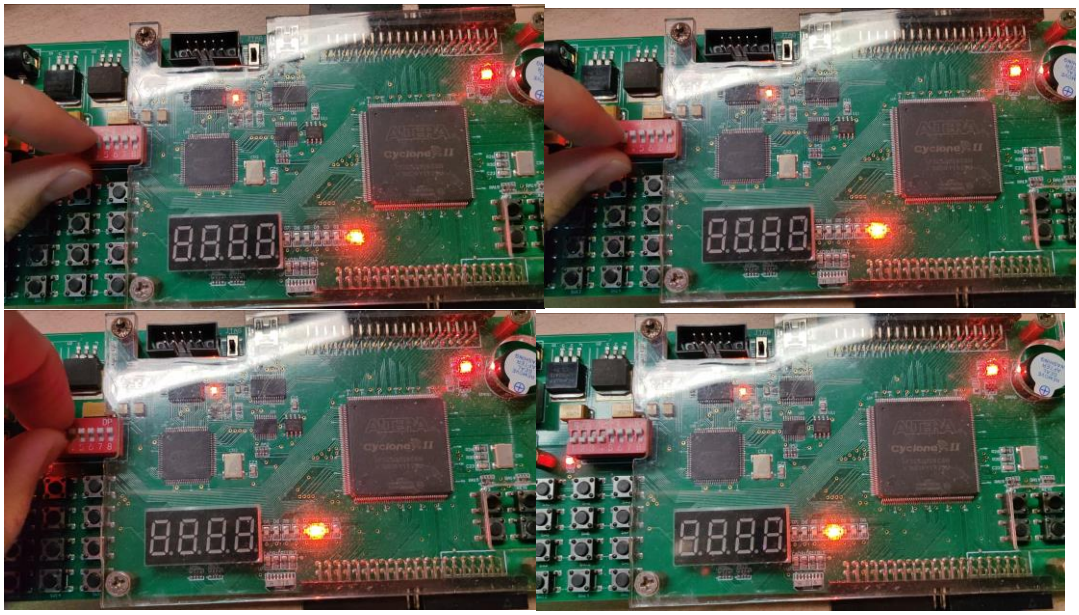


图 4 实物检验

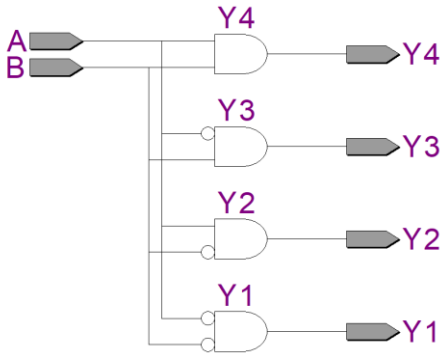
四、选做任务

1. 组合逻辑模式

```
module eda2(
    input wire A,
    input wire B,
    output wire Y1,
    output wire Y2,
    output wire Y3,
    output wire Y4
);

assign Y1 = ~A & ~B;   // Y1 = A'B'
assign Y2 = A & ~B;    // Y2 = AB'
assign Y3 = ~A & B;    // Y3 = A'B
assign Y4 = A & B;     // Y4 = AB

endmodule
```



2. 结构化逻辑模式

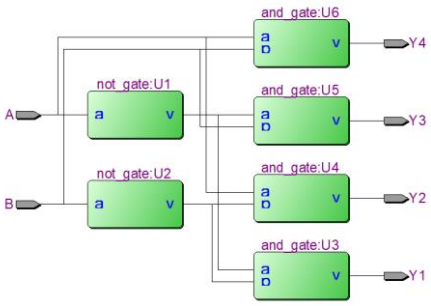
```
module and_gate (
    input wire a,
    input wire b,
    output wire y
);
assign y = a & b;
endmodule

module not_gate (
    input wire a,
    output wire y
);
assign y = ~a;
endmodule

module eda2(
    input wire A,
    input wire B,
    output wire Y1,
    output wire Y2,
    output wire Y3,
    output wire Y4
);
wire A_not, B_not;

not_gate U1 (.a(A), .y(A_not));
not_gate U2 (.a(B), .y(B_not));

and_gate U3 (.a(A_not), .b(B_not), .y(Y1)); // Y1 = A'B'
and_gate U4 (.a(A), .b(B_not), .y(Y2));     // Y2 = AB'
and_gate U5 (.a(A_not), .b(B), .y(Y3));     // Y3 = A'B
and_gate U6 (.a(A), .b(B), .y(Y4));         // Y4 = AB
endmodule
```

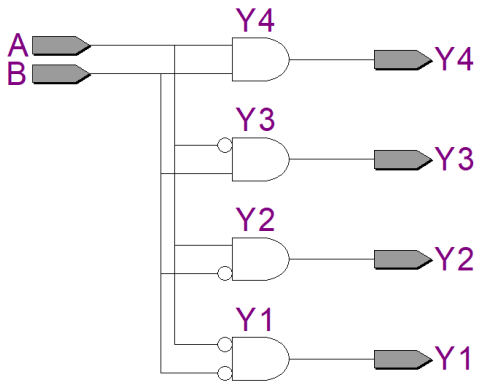


3. 行为描述模式

```
module eda2 (
    input wire A,
    input wire B,
    output reg Y1,
    output reg Y2,
    output reg Y3,
    output reg Y4
);

always @(*) begin
    Y1 = ~A & ~B;   // Y1 = A'B'
    Y2 = A & ~B;    // Y2 = AB'
    Y3 = ~A & B;    // Y3 = A'B
    Y4 = A & B;     // Y4 = AB
end

endmodule
```



五、实验报告-实验总结

1. 使用 EDA 软件进行电路设计的基本流程

1） 新建工程：按照向导步骤设置项目名称和位置，选择适当的 FPGA 器件。

2） 创建设计文件并进行电路设计（原理图或硬件设计语言）

3） 分析电路：初步检查电路中可能存在的问题

4） 功能仿真：建立波形文件，检查逻辑功能是否正确

5） 执行全编译

6） 引脚锁定与下载

2. 比较使用原理图输入方式和硬件描述语言输入方式进行电路设计的各自优势。

• 原理图：

• 原理图提供了直观的电路图，直观展示电路结构，方便分析电路中的信号流，进行分析与调试。

• 硬件描述语言：

• 如果已知逻辑表达式，想要用电路实现，可以直接转写为硬件描述语言，无需人工进行电路本身的设计

• 支持更复杂的设计逻辑，适合大规模和复杂电路设计

• 代码的形式便于重用与集成，减少重复性工作

3. 记录实验过程中软件提示的各项红色 Error 信息，并说明如何解决

```
Command: quartus_sim --simulation_results_format=VWF select -c selec
t
FLEXlm software error: Invalid (inconsistent) license key.  The lice
nse key and data for the feature do not match.  This usually happens
when a license file has been altered. Feature:       quartus License
path:  D:\altera\13.0\license.dat FLEXnet Licensing error:-8,523 For
further information, refer to the FLEXnet Licensing End User Guide,
available at "www.macrovision.com".
Feature Simulator is only available with a valid subscription licens
e. You can purchase a software subscription to gain full access to t
his feature.
Quartus II 64-Bit Simulator was unsuccessful. 0 errors, 2 warnings
  Peak virtual memory: 4452 megabytes
  Processing ended: Thu Oct 03 08:48:32 2024
  Elapsed time: 00:00:00
  Total CPU time (on all processors): 00:00:00
```

• 与 license 相关，重新对软件进行破解即可

Type	ID	Message
		**********
>		Running Quartus II 64-Bit Analysis & Synthesis
		Command: quartus_map --read_settings_files=on --write_settings_files=off eda2 -c eda2
	11104	Parallel Compilation has detected 20 hyper-threaded processors. However, the extra hype
>	12021	Found 1 design units, including 1 entities, in source file verilog1.v
	12007	Top-level design entity "eda2" is undefined
>		Quartus II 64-Bit Analysis & Synthesis was unsuccessful. 1 error, 0 warnings
	293001	Quartus II Full Compilation was unsuccessful. 3 errors, 0 warnings

•注意顶层模块名称要与工程名称一致，工程名为 eda2，则顶层模块也要命名为 eda2