# 模块实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 叶福伦 | 学号 | 21051331 |
| 专业 | 计算机科学与技术 | 班级 | 21052313 |
| 指导教师 | 髙志刚 | 课程名称 | 计算机组成原理课程设计（甲） |
| 实验序号 | 01 | 实验名称 | 加法器设计实验 |
| 实验时间 | 05.05 | 实验地点 | 1教223 |

# 成绩表

|  |  |  |
| --- | --- | --- |
| **考核项目** | **考查分项** | **得分** |
| **一、实验方案设计（30分）** | **实验目的与要求（5分）** |  |
| **模块构成与连接（15分）** |  |
| **板级验证方案（10分）** |  |
| **二、FPGA程序设计（20分）** | **程序源代码（15分）** |  |
| **管脚约束程序源代码（5分）** |  |
| **三、仿真实验（15分）** | **仿真代码（5分）** |  |
| **仿真波形及结果分析（10分）** |  |
| **四、板级实验（15分）** | **实验操作（5分）** |  |
| **结果记录（5分）** |  |
| **结果分析（5分）** |  |
| **五、思考与探索（10分）** | **拓展提升（10分）** |  |
| **格式（10分）** | **书写规范、图表清晰、表达清楚** |  |
| **合计** | |  |

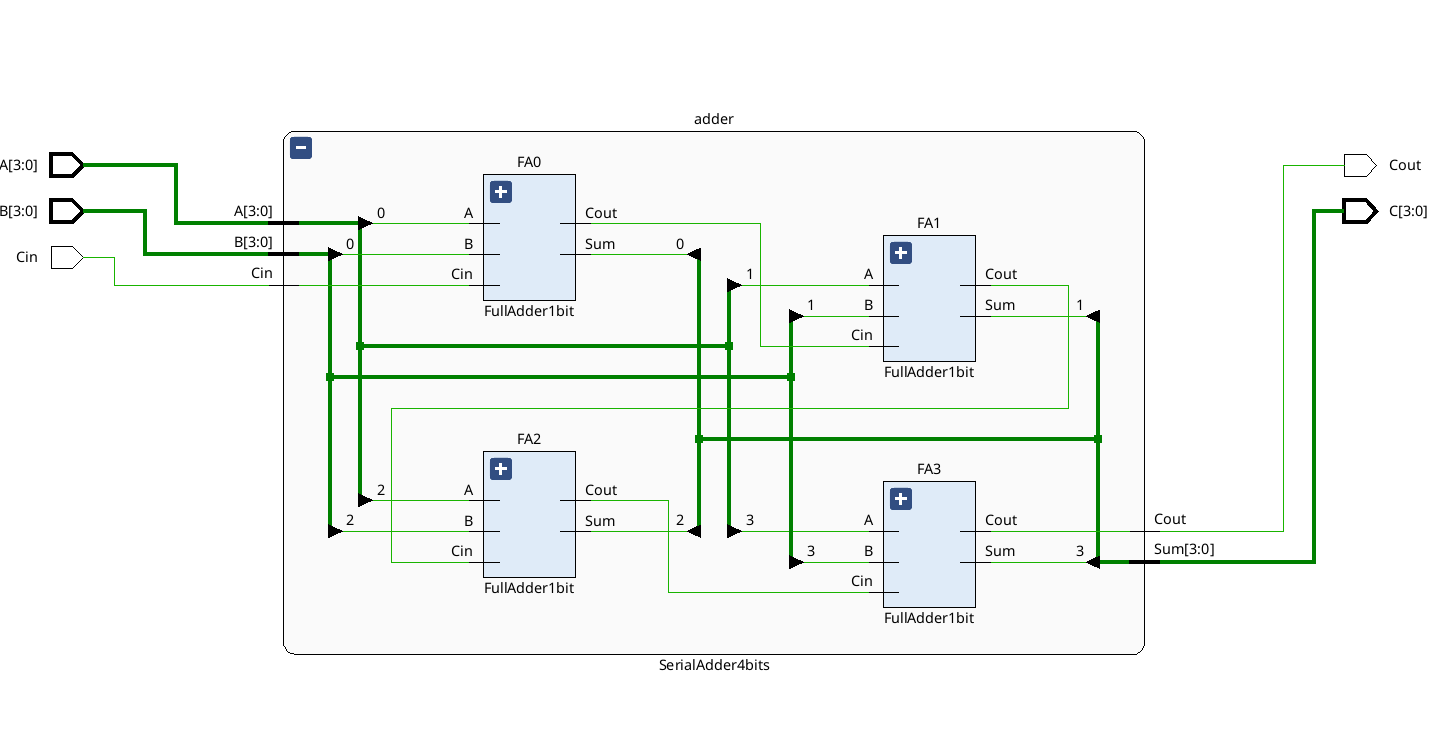
# 一、实验方案设计

1. 实验目的与要求（5分）

掌握全加器、串行进位加法器和超前进位加法器的原理与设计方法； 掌握运用Verilog HDL语言进行结构和数据流描述与建模方法； 掌握运用Verilog HDL语言进行模块调用的方法。

1. 模块构成与连接（15分）

（请附图，并说明各模块的功能，以及模块间的连接关系。也可附开发工具中显示的电路模块连接图，加以说明。）



cin传进adder内作为第一级的全加器的进位信号。次一级的全加器的计算依赖于前一级全加器的产生的进位信号和对应位的AB。最后一级的全加器产生的进位信号作为整个加法器产生的进位信号。

1. 板级验证方案（10分）

（说明顶层测试模块的输入/输出信号，及对应使用的I/O设备；可附图说明）

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 信号名 | 使用的IO设备 | 说明 |
| input | Cin | 逻辑开关 | 输入的进位信号 |
| A[3:0] | 一组逻辑开关 | 被加数 |
| B[3:0] | 一组逻辑开关 | 加数 |
| output | Cout | LED灯 | 产生的进位信号 |
| C[3:0] | 一组LED灯 | 结果 |

# 二、FPGA程序设计

1. 实验程序源代码（25分）

（实验各个模块的代码，包含功能注释）

//FullAdder1bit.v

module FullAdder1bit (

input A,

input B,

input Cin,

output Sum,

output Cout

);

wire AxorB, AandB, AxorBandCin;

xor (Sum, A, B, Cin);

and (AandB, A, B);

xor (AxorB, A, B);

and (AxorBandCin, AxorB, Cin);

or (Cout, AandB, AxorBandCin);

endmodule

//SerialAdder4bits.v

`include "./FullAdder1bit.v"

module SerialAdder4bits (

input [3:0] A,

input [3:0] B,

input Cin,

output [3:0] Sum,

output Cout

);

wire carry1, carry2, carry3;

FullAdder1bit FA0 (.A(A[0]), .B(B[0]), .Cin(Cin), .Sum(Sum[0]), .Cout(carry1));

FullAdder1bit FA1 (.A(A[1]), .B(B[1]), .Cin(carry1), .Sum(Sum[1]), .Cout(carry2));

FullAdder1bit FA2 (.A(A[2]), .B(B[2]), .Cin(carry2), .Sum(Sum[2]), .Cout(carry3));

FullAdder1bit FA3 (.A(A[3]), .B(B[3]), .Cin(carry3), .Sum(Sum[3]), .Cout(Cout));

endmodule

//顶层模块

module SerialAdder4bitsTopper(

input [3:0] A,

input [3:0] B,

input Cin,

output [3:0] C,

output Cout

);

SerialAdder4bits adder(A,B,Cin,C,Cout);

endmodule

1. FPGA管脚约束代码（5分）

（引脚约束文件的内容，备注使用的具体设备）

A，B 使用逻辑开关，input Cin使用逻辑开关，C[3:0]，Cout使用led灯

set\_property PACKAGE\_PIN R1 [get\_ports Cout]

set\_property PACKAGE\_PIN AB8 [get\_ports Cin]

set\_property PACKAGE\_PIN T3 [get\_ports {A[3]}]

set\_property PACKAGE\_PIN U3 [get\_ports {A[2]}]

set\_property PACKAGE\_PIN T4 [get\_ports {A[1]}]

set\_property PACKAGE\_PIN V3 [get\_ports {A[0]}]

set\_property PACKAGE\_PIN V4 [get\_ports {B[3]}]

set\_property PACKAGE\_PIN W4 [get\_ports {B[2]}]

set\_property PACKAGE\_PIN Y4 [get\_ports {B[1]}]

set\_property PACKAGE\_PIN Y6 [get\_ports {B[0]}]

set\_property PACKAGE\_PIN G1 [get\_ports {C[3]}]

set\_property PACKAGE\_PIN E1 [get\_ports {C[2]}]

set\_property PACKAGE\_PIN D2 [get\_ports {C[1]}]

set\_property PACKAGE\_PIN A1 [get\_ports {C[0]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports Cin]

set\_property IOSTANDARD LVCMOS18 [get\_ports Cout]

set\_property IOSTANDARD LVCMOS18 [get\_ports {A[3]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {A[2]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {A[1]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {A[0]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {B[3]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {B[2]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {B[1]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {B[0]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {C[3]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {C[2]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {C[1]}]

set\_property IOSTANDARD LVCMOS18 [get\_ports {C[0]}]

set\_property PULLDOWN true [get\_ports Cin]

set\_property PULLDOWN true [get\_ports {A[3]}]

set\_property PULLDOWN true [get\_ports {A[2]}]

set\_property PULLDOWN true [get\_ports {A[1]}]

set\_property PULLDOWN true [get\_ports {A[0]}]

set\_property PULLDOWN true [get\_ports {B[3]}]

set\_property PULLDOWN true [get\_ports {B[2]}]

set\_property PULLDOWN true [get\_ports {B[1]}]

set\_property PULLDOWN true [get\_ports {B[0]}]

set\_property BITSTREAM.GENERAL.COMPRESS TRUE [current\_design]

# 三、仿真实验

（只对CPU模块仿真，无需对板级验证的顶层测试模块仿真）

1. 仿真代码（5分）

（含仿真源代码、仿真验证方案）

`timescale 1ns / 1ps

module SerialAdder4bitsTester();

reg [3:0]A;

reg [3:0]B;

reg Cin;

wire [3:0]C;

wire Cout;

SerialAdder4bits SerialAdder4bits(

A,

B,

Cin,

C,

Cout

);

initial begin

A = 4'b0000;

B = 4'b0000;

Cin = 0;

#100

A = 4'b0001;

B = 4'b0100;

Cin = 0;

#100

A = 4'b1111;

B = 4'b1111;

Cin = 1;

#100

A = 4'b1100;

B = 4'b0011;

Cin = 0;

#100

A = 4'b1101;

B = 4'b0011;

Cin = 1;

#1000

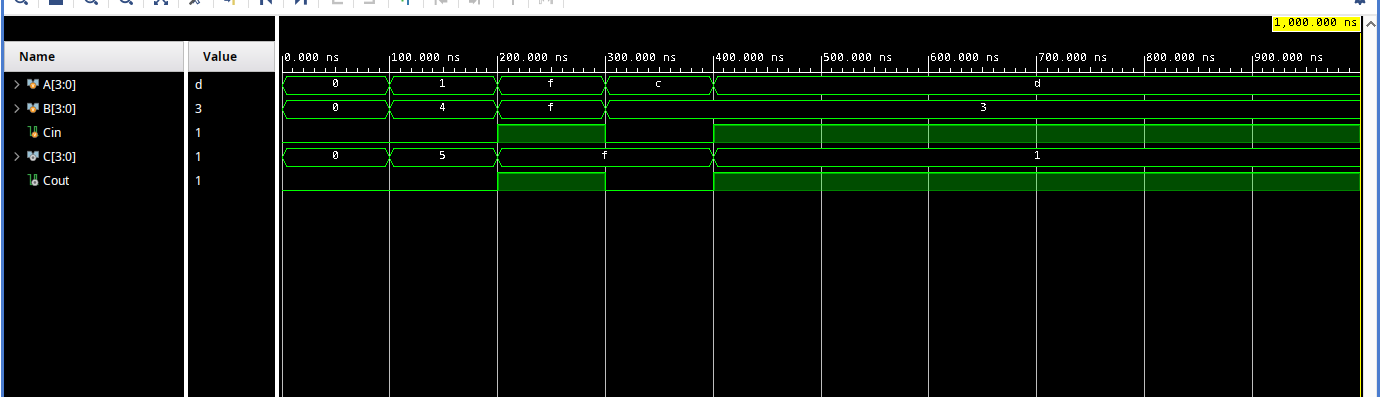
$finish;

end

endmodule

1. 仿真波形及结果分析（15分）

（仿真波形截图，并对仿真波形进行分析）



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 时间段 | A | B | Cin | C | 期望的C | Cout | 期望的Cout | 说明 |
| 000 - 100ns | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 考察没有cin时是否出错 |
| 100 - 200ns | 1 | 4 | 0 | 5 | 5 | 0 | 0 | 同上 |
| 200 - 300ns | F | F | 1 | F | F | 1 | 1 | 考察有cin且有进位的情况 |
| 300 - 400ns | C | 3 | 0 | F | F | 0 | 0 | 考察没有cin但是有进位的情况 |
| 400ns之后 | D | 3 | 1 | 1 | 1 | 1 | 1 |  |

# 通过仿真结果可以大致确定电路是正确的

# 四、板级实验

1. 板级实验操作说明（5分）

将引脚约束好的FPGA程序下载到芯片中，然后利用FPGA芯片引脚上所连接的输入输出设备（开关、按键、显示灯和数码管等）验证：操作输入设备，读取输出设备上的结果，来判断程序的逻辑是否正确。

1. 板级实验结果记录（5分）

（以表格形式展现，具体见教材上实验结果记录表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | C | Cout |
| 0000 | 0000 | 0 | 0000 | 0 |
| 1111 | 1111 | 1 | 1111 | 1 |
| 0000 | 0001 | 0 | 0001 | 0 |
| 1111 | 0000 | 1 | 0000 | 1 |
| 1110 | 1011 | 0 | 1001 | 1 |
| 1010 | 1011 | 1 | 0110 | 1 |
| 0001 | 1100 | 1 | 1110 | 0 |
| 0111 | 0011 | 0 | 1010 | 0 |

1. 结果分析与结论（5分）

（分析实验结果，给出实验结论）

C和Cout的结果正确，虽然没有覆盖到所有可能性，但能判断加法器的正确性。

# 五、思考与探索

（10分）

1. 问题与解决方案

（整个实验过程中发生了什么问题？你是如何解决的。）

不会门级电路建模：巩固基础知识，温习verilog语法

1. 思考题

（力所能及，尝试实践或回答教材上的思考与探索题目，至少完成1道）

1. 不一致，采用了不同的优化算法、采取了优化策略等

3. 4\*t n\*t

4. 不同verilog加法是超前进位电路

# 六、实验心得体会、意见建议

（如果是小组合作，请给出具体分工、分别给出心得体会）

在进行串行加法器的实验过程中，我深刻认识到了数字电路设计的重要性。以下是我在实验中的心得体会：

1. 熟悉串行加法器的原理和设计思路：在进行实验前，需要深入了解串行加法器的原理和设计思路，包括串行进位、串行加法等概念，这有助于更好地理解电路的功能和设计。
2. 设计合理的电路结构：根据实验要求，我设计了一个包括移位寄存器、异或门和与非门等模块的串行加法器电路。需要注意的是，电路结构的设计需要综合考虑电路的性能和复杂度，同时需要考虑电路的稳定性和可靠性。
3. 编写Verilog HDL代码：在实验中，我使用Verilog HDL编写了电路的描述代码，实现了电路的逻辑功能。需要注意的是，代码的编写需要符合Verilog HDL的语法规范，并且需要考虑模块之间的连接和数据传输等问题。
4. 验证电路的功能和性能：我使用不同的测试数据对电路进行了验证，包括测试加法、进位、溢出等功能。同时，我还对电路的运行速度和功耗等性能进行了评估，以确保电路的稳定性和可靠性。

通过这次实验，我不仅深入了解了串行加法器的原理和设计方法，同时也掌握了Verilog HDL的基本使用方法，对计算机组成原理有了更深刻的认识。