

第 5 章 乘法器设计

一、单元概述

1. 主要知识内容

本章主要通过一个具体实例,基于 FPGA 的乘法器设计与实现,将高级数字系统设计方法应用于实际项目中,并用 VerilogHDL 语言对电路进行描述及仿真验证,最终在 FPGA 开发板上进行硬件实现。

2. 教学总体目标

- (1) 熟练应用基于 FPGA 的数字系统设计方法;
- (2) 熟练使用 VerilogHDL 进行硬件描述;
- (3) 熟练进行功能仿真及硬件调试。

二、单元重点与难点

1. 重点与难点

重点:熟练应用基于 FPGA 的数字系统设计方法。

2. 解决方案

通过项目范例进行实践,对设计的每个环节进行练习,通过不断的实践达到熟练掌握的目的。

5.1 项目目标

本章通过完成基于 FPGA 的 4 位无符号二进制移位式乘法器的代码编写、仿真、逻辑综合、布局布线及下载,进一步掌握基于 FPGA 的数字系统设计流程及 EDA 工具的使用,并规范撰写项目报告。

5.2 项目设计

5.2.1 功能要求

乘法器系统需求描述如下：

1. 用时序电路实现一个二进制正整数乘法器；
2. 该乘法器使用二进制乘法算法进行乘法运算；
3. 该系统可以进行两个 4 位二进制的乘法运算，并给出计算结果；
4. 输入输出接口定义，如图 5.1 所示。

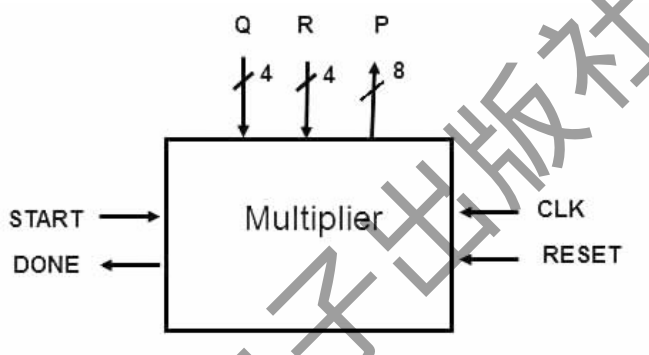


图 5.1 系统整体模块图

各管脚功能说明如表 5.1 所示。

表 5.1 系统整体管脚说明

信号	I/O	宽度	功能描述
clk	I	1	系统时钟
RESET	I	1	系统异步复位,高电平有效
START	I	1	开始一次运算的控制信号
Q	I	1	乘数
R	I	1	被乘数
P	I	1	计算结果
DONE	O	1	DONE

系统具有一个控制信号:START,当系统探测到 START 信号有效时从端口取得乘数 Q 和被乘数 R 进行乘法运算。计算完成后输出计算结果 P,同时通过一个系统级状态信号 DONE 通知外界乘法完成。另外,系统具有异步复位信号 RESET,用于进行系统复位操作。

5.2.2 系统定义

从系统高层定义开始,逐步细化一个或多个可能设计方案,采用自上而下的设计方法

进行系统设计。

设计的难点在于算法的设计。对于给定的乘法运算功能,导出控制算法,画出系统流程图,并反复完善。值得注意的是,算法的确定不是轻而易举的事,往往需要经过不断流程图细化分析修改,在考虑了系统可能遇到的各种可能性之后(包括正常使用的情况和可能出现的异常情况)才能确定下来。

我们采用乘数的每一位与被乘数相乘,最后将所有结果位移相加的算法来完成系统功能设计。具体步骤如表 5.2 所示,R 为被乘数,Q 为乘数,P 为乘积,G 为位移相加过程中产生的进位。

表 5.2 二进制乘法算法验证表

R	Q	G	P	备注
1101	1011	0	0000 0000	初始状态
		0	1101 0000	$P1 = P0 + R$
	0101	0	0110 1000	$\{G, P\} \gg \gg$ $Q \gg \gg$
		1	0011 1000	$P2 = P1 + R$
	0010	0	1001 1100	$\{G, P\} \gg \gg$ $Q \gg \gg$
	0001	0	0100 1110	$\{G, P\} \gg \gg$ $Q \gg \gg$
		1	0001 1110	$P4 = P3 + R$
	0000	0	1000 1111	$\{G, P\} \gg \gg$ $Q \gg \gg$

根据该算法,画出系统流程图,记录在项目报告相应部分,并进行说明。

根据自上而下的设计方法,由系统流程图,提取出数据通道及控制单元。将数据通道及控制单元状态机分别画在项目报告的相应部分,并画出数据通道与控制单元连接图及管脚说明。

将自动售咖啡机顶层模块,包括对数据通道及控制单元进行实例化的 RTL 代码写在实验报告相应位置处。

5.3 项目实施

5.3.1 数据通道详细设计

根据系统流程图,从流程图中识别出数据处理相关的操作,分析并选择相应的 RTL 级器件实现相应的数据处理操作。同时识别出进行数据处理过程中需要从控制单元取得的相

关控制信号,最终完成数据通道模块各子模块及其连接关系,数据通道框图如图 5.2 所示,包括加法器、移位寄存器、计数器、比较器等。

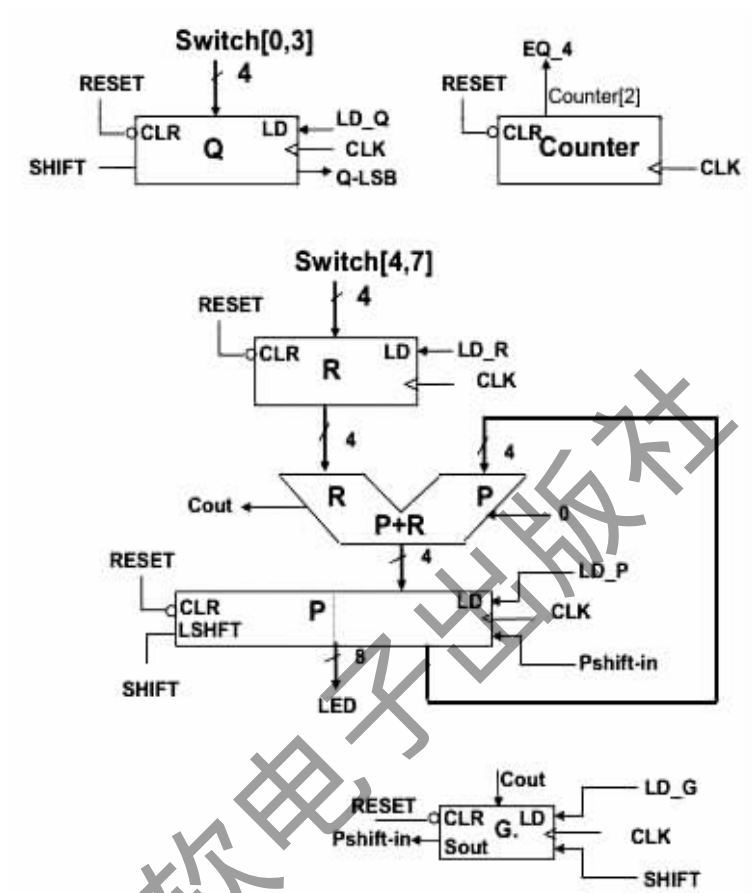


图 5.2 数据通道框图

按照数据通路原理图,用 VerilogHDL 完成数据通路中每个器件定义单独的模块(module),并用 RTL 级描述各模块功能。完成每个器件的模块定义之后,再定义一个顶层模块,在该模块内将所有 RTL 级器件模块进行实例化,进行各模块连接,并在 Quartus II 中完成语法检查及综合。

5.3.2 控制单元详细设计

根据设计出的乘法器的算法,以及系统流程图,完成状态机的状态转移图设计。并将状态机绘制在项目报告的相应部分。

根据状态转换图,运用 HDL 和 EDA 工具完成控制单元(状态机)的设计。使用 verilog 语言完成乘法器控制单元的 RTL 级描述,采用两段式结构描述,并将描述记录在项目报告的相应部分。在 Quartus II 中完成语法检查及综合。

5.4 项目运行

5.4.1 仿真验证

在 verilog 仿真软件中对系统进行仿真。仿真是对系统的功能性是否正确进行验证,我们需要编写测试平台(testbench)来对乘法器进行功能仿真。在测试平台中,需要使用典型和非典型输入向量来验证该系统行为是否正确。我们一般选取几组数据进行乘法验证,需要注意的是尽可能保证测试数据能够涵盖一些特殊的情况,比如取一些极端值,如 0000、1111 等,并尽量使被乘数每一个 bit 都能取到 1,0 两种值。

注意,在实际的数字系统中或在实验室里,时钟源通常来自电路外部的一个石英振荡器,我们使用 altera 开发板给我们提供的时钟信号。而在代码仿真的过程中,我们还不能使用开发板,所以需要设计一个时钟的行为级的 Verilog-HDL 模型,用于系统仿真。但是需要注意的是,这是一个不可逻辑综合的模型。

```
module Sys_Clk(clk);  
output clk;  
reg clk;  
parameter clk_freq= 100; // 时钟频率  
initial begin  
    repeat (100)  
        begin  
            #(clk_freq/2) clk=0; //高低电平各占半个时钟周期  
            #(clk_freq/2) clk=1;  
        end  
    end  
endmodule
```

在上面给出的时钟信号参考代码中,可以通过改变参数 clk_freq 的取值,方便的改变该时钟模块输出的时钟信号频率。在系统仿真时,将该时钟模块代码放入 testbench 中,输出信号接入系统 clk 端口,为系统提供时钟信号。

在仿真波形图中观察系统的输入输出信号值,确定测试平台中的测试向量按照系统时序正确的被输入到被测系统中,观察系统输出信号值,是否正确完成了乘法功能,并且输出信号的时序是否正确。

5.4.2 FPGA 实验板验证

1. 完成电路仿真及综合,并检查综合结果是否正确。
2. 为设计添加约束,包括时序约束及管脚约束。

Quartus II 提供两种时序约束方式,分别为 Classic Timing Analyzer 和 TimeQuest Timing Analyzer,默认情况下使用前者。

将管脚约束对应表记录在项目报告相应部分。用到的 Altera FPGA 开发板硬件资源有外部时钟、拨码开关、按钮、LED 灯、数码管等。如果使用数码管,需要增加显示译码电路,各硬件资源管脚分配情况参见附录 A 开发板常用硬件资源。

将最终时序分析结果、占用 FPGA 资源情况,记录在项目报告相应部分。

3. 在 Quartus II 中实现综合布局布线,并生成 .sof 和 .pof 配置文件。当处于硬件调试阶段,推荐将 .sof 文件下载到 FPGA 中,虽然掉电丢失,但足以满足调试要求。

5.5 项目检验

通过课前提问,了解学生的预习情况,并通过项目仿真以及 FPGA 实验板实现的完成情况情况进行检查。课后检查项目报告的撰写情况。

5.6 强化练习

本项目完成了一个四位乘法器的设计和验证,在设计状态机和数据通道的过程中应该不断思考,不断对系统进行优化,追求使用更少的资源完成相同的功能,同时在性能上避免损失。考虑系统中是否还存在可以改进的地方,从下面几方面进行思考:

1. 数据通道中使用了一个 8bit 的寄存器 P 进行乘积的存储,考虑 8bit 是否必要? 是否可以使用更小容量的寄存器达到同样的功能? 如果可以的话,应该怎么实现?

2. 数据通道中多次出现了位移寄存器,试验中为每个寄存器我们都编写了一个寄存器模块代码。是否可以考虑编写一个通用的寄存器模块,使该模块能够满足所有位移寄存器的功能需求,当需要使用位移寄存器的时候,只需要实例化出一个实例进行使用即可,实现代码复用,提高效率。考虑系统中是否还有可以进行代码复用的相似模块。