

DOI:10.3969/j.issn.1000-1565.2014.01.017

深亚微米并行 CRC32 编码芯片的设计和实现

郭宝增, 吴鹏飞

(河北大学 电子信息工程学院, 河北 保定 071002)

摘要:在分析 CRC 编码算法的基础上,从传统的串行编码算法着手,推导出适合高速通信的并行算法,通过 FPGA(现场可编程门阵列)验证确保算法代码的逻辑功能正确;采用中芯国际 simc18(180 nm 工艺库)实现了并行 CRC32 编码芯片的设计.该设计具有编码速度快、占用资源少、低功耗、易于量产等优点.

关键词:CRC32;并行;FPGA;Design Compiler;Soc Encounter

中图分类号:TP33.1 **文献标志码:**A **文章编号:**1000-1565(2014)01-0089-05

Design and implementation of deep sub-micron parallel CRC32 coding chip

GUO Baozeng, WU Pengfei

(College of Electronic and Informational Engineering, Hebei University, Baoding 071002, China)

Abstract: This paper analyzes the theory of CRC coding algorithm, calculates parallel algorithm which applies to high-speed communication, according to the traditional serial coding algorithm, ensures the logic function correctly by FPGA verification; achieves parallel CRC32 coding chip design with simc18 (180 nm technology library). The advantages of the design is faster coding, less resource, low-power, easier to large-scale production and so on.

Key words: CRC32; Parallel; FPGA; Design Compiler; Soc Encounter

CRC 的全称是循环冗余校验(cyclic redundancy check),是一种非常著名的检错手段,在通信和数据存储领域得到广泛应用.其特点是检错能力极强,开销小,易于编码器及检测电路实现.从检错能力来看,其不能发现的错误的几率在 0.004 7% 以下.著名的通讯协议 X.25 的 FCS(帧检错序列)采用的是 CRC-CCITT; ARJ, LH 等压缩工具软件采用的是 CRC32;磁盘驱动器的读写采用了 CRC16;通用的图像存储格式 GIF, TIFF 等都使用 CRC 作为检错手段^[1].

收稿日期:2013-02-19

基金项目:河北省自然科学基金资助项目(F2009000226);河北省教育厅科学研究计划项目(2008308)

第一作者:郭宝增(1953-),男,河北承德人,河北大学教授,主要从事集成电路设计及验证方向研究.

E-mail:guobaozeng@yahoo.com.cn

通信作者:吴鹏飞(1987-),男,河北承德人,河北大学在读硕士研究生,主要从事基于 SOPC 的嵌入式技术开发、数字集成电路设计研究. E-mail:wupengfei 2008 2008@126.com.

1 CRC32 算法研究

1.1 LSFR 串行编码算法

CRC32 属于线性分组码的分支,同时还具有循环特性,符合一般循环码的编码方法^[2],其过程可以采用 LSFR 实现编码电路,这种电路主要由移位寄存器和异或门组成^[3-4]. 对于 CRC32 校验码,生成多项式为

$$g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1. \tag{1}$$

以 8 bit 的数据输入为例,其硬件实现框图如图 1 所示.

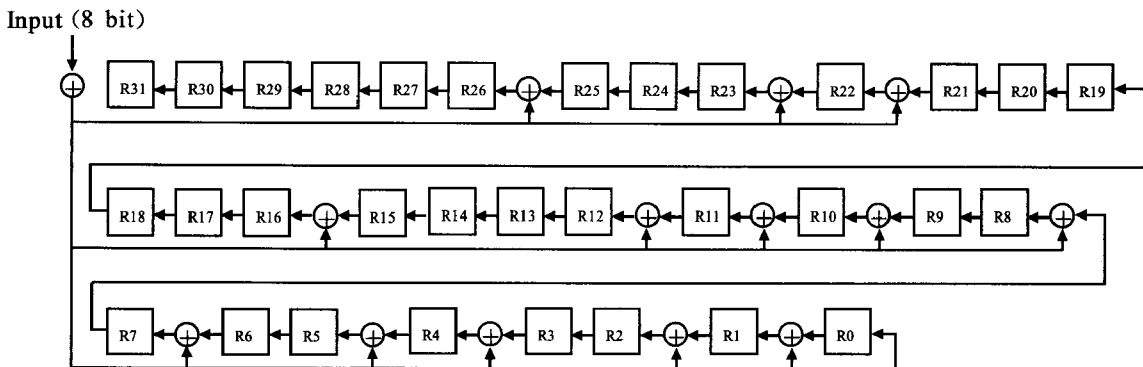


图 1 LSFR 串行编码电路

Fig. 1 LSFR serial encoder circuit

图 1 中,32 位移位寄存器用于存储 CRC32 校验码,称其为 CRC 寄存器. 编码前将寄存器各位置 1,每个时钟周期 1 bit 数据输入,通过移位和异或运算,CRC 寄存器中的数据随之更新,8 个时钟周期后,数据输入完毕,此时 CRC 寄存器的值即为 CRC32 的码序列,该算法原理简单,便于硬件实现,但效率不高,主要应用于串行通信.

1.2 并行 CRC32 算法

从本质上说,LSFR 算法是一种串行算法,针对 LSFR 电路效率低的问题,本文从 LSFR 电路的结构入手,通过相应的逻辑运算推导出高效并行的 CRC32 编码算法.

以 8 bit 的数据输入为例,设 R_j^i 为第 j 位移位寄存器的状态值, i 表示移位的次数, R_0^8 就表示 R_0 移位寄存器经过 8 个时钟后的状态值, D 为输入信息,取值为 D_0 至 D_7 ,由图 1 可知(“ \wedge ”表示异或运算)

$$R_0^8 = R_{31}^7 \wedge D_7, \tag{2}$$

$$R_{31}^7 = R_{30}^6 = R_{29}^5 = R_{28}^4 = R_{27}^3 = R_{26}^2, \tag{3}$$

$$R_{26}^2 = R_{25}^1 \wedge R_{31}^1 \wedge D_1, \tag{4}$$

$$R_{25}^1 = R_{24}^0, \tag{5}$$

$$R_{31}^1 = R_{30}^0. \tag{6}$$

将(3),(4),(5),(6)式代入(2)式,可以求得

$$R_0^8 = R_{30}^0 \wedge D_7 \wedge R_{24}^0 \wedge D_1, \tag{7}$$

即移位寄存器 R_0 经过 8 个时钟后的状态值可以表示为移位寄存器 R_{30}, R_{24} 的初始值与输入数据 D_7, D_1 的异或运算,这样就可以将 LSFR 中 8 bit 的串行输入数据转换为等效的并行输入,将其记做 CRC_0 . 按照上述的推导原则,可以求得其他 31 个寄存器经过 8 个时钟后的状态值.

8 bit 并行输入,经过并行运算单元运算之后,将运算结果存储到移位寄存器 $R_0 - R_{31}$,移位寄存中的值有 2 个用途:当前时钟周期下,直接以运算结果的形式作为输出;下一时钟周期作为保护数据反馈到并行运算单元,共同作用到下一时钟周期.

2 并行 CRC32 算法的 FPGA 验证

在算法研究的基础上,鉴于 ASIC(application specific integrated circuit)流片成本高,风险大,因此大规

模量产之前需要事先对芯片的设计进行 FPGA 验证. 通过验证,一方面可以确保系统逻辑功能的正确;另一方面可以说明系统的时序基本可以满足设计要求^[3].

依据推导出的逻辑关系,编写并行 CRC32 算法的 verilog 代码^[5]

```
module crc(
    input[7:0] data_in,
    input crc_en,
    output [31:0] crc_out,
    input    rst,
    input    clk);
    reg[31:0] lfsr_q,lfsr_c;
    assign crc_out = lfsr_q;
    always @( * ) begin
        lfsr_c[0] = lfsr_q[30] ^ data_in[7] ^ lfsr_q[24] ^ data_in[1];
        :
        lfsr_c[31] = lfsr_q[23] ^ lfsr_q[29] ^ data_in[2];
    end
    always @(posedge clk, posedge rst) begin
        if(rst) begin
            lfsr_q <= {32{1'b1}};
        end
        else begin
            lfsr_q <= crc_en ? lfsr_c : lfsr_q;
        end
    end
end
endmodule
```

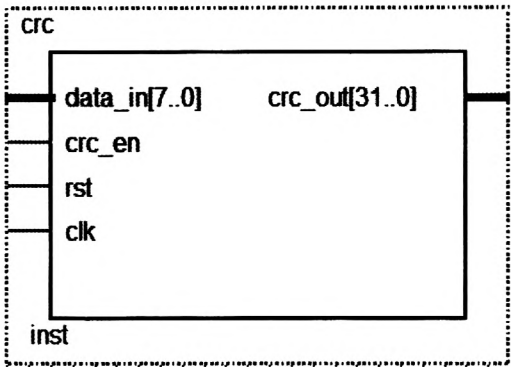


图 2 并行 CRC32 编码模块
Fig. 2 Parallel CRC32 encoder module

生成的并行 CRC32 编码模块如图 2 所示.

图 2 中 data_in 为 8 位数据输入端;crc_en 为使能端,只有当 crc_en 为高电平时,编码器才进行编码工作;rst 为复位信号(高电平有效),当电平为高时,进行初始化;clk 为时钟信号;crc_out 为 32 位 CRC 编码输出端.

通过编写 testbench,利用 Modelsim 软件对生成的编码模块进行功能仿真,其仿真结果如图 3 所示.

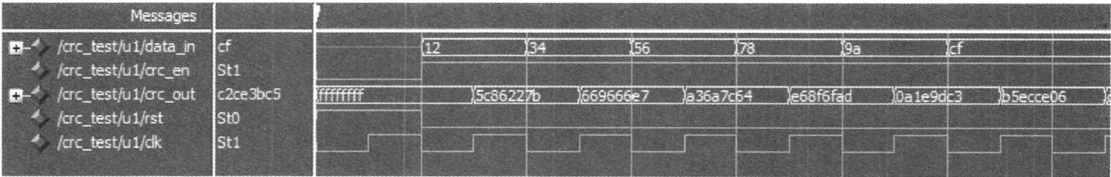


图 3 并行 CRC32 编码电路功能仿真结果
Fig. 3 Function simulation result of Parallel CRC32 encoder circuit

仿真输入数据为 8'h12, 8'h34, 8'h56, 8'h78, 8'h9a, 8'hcf; 得到的 CRC32 校验码为 16'h5c86227b, 16'h669666e7, 16'ha36a7c64, 16'he68f6fad, 16'h0a1e9dc3, 16'hb5ecce06, 通过观察仿真结果,其编码结果与理论计算的结果一致.

将编译后的工程文件下载到 DE2-70 开发板,通过实验检测,可以得到一致的输出结果,至此芯片代码的 FPGA 验证工作结束.

3 并行 CRC32 编码芯片的实现

FPGA 的使用费用太高,资源利用率较低;在功耗、设计密度以及产品的运算速度方面相对 ASIC 来说存在相当程度的劣势,因此,为了实现产品的大规模生产,必须进行系统逻辑的版图 GDS II 设计. 设计过程采用中芯国际 180 nm 的数字工艺库,主要使用 Synopsys 公司的 Design Compiler 逻辑综合软件,和 Cadence 公司的 Soc Encounter 自动布局、布线软件实现版图的 GDS II 文件设计.

3.1 编码电路的逻辑综合

使用 Design Compiler 软件,对编码电路的 verilog 代码进行逻辑综合,其主要工作包括:一方面将硬件语言描述的行为级电路、RTL 级电路转换到基于工艺库的门级网表^[6];另一方面要根据综合产生的时序报告,对综合过程中的各项参数进行修改,以保证时间余量(slack)为给定时钟的 30%以上,这是由于综合过程中设置的连线负载只是工艺库提供的延时参数,并不是实际延时,必须以足够的 slack 为后续的实际布局、布线提供充足的延时空

clock clk (rise edge)	10.00	10.00
clock network delay (ideal)	0.30	10.30
clock uncertainty	-0.01	10.29
crc32/lfsr_q_reg[11]/CK (DFFSX1)	0.00	10.29 r
library setup time	-0.20	10.09
data required time		10.09

data required time		10.09
data arrival time		-3.91

slack (MET)		6.18

图 4 逻辑综合时序报告

Fig. 4 Logic synthesis timing report

从图中可以看出给定的时钟 clk 为 10 ns,而综合后的 slack 为 6.18 ns,超过要求的最少时间余量 3 ns,说明综合过程可以提供充足的时间余量用于布局、布线.

将满足上述要求综合后的 verilog 文件(门级网表文件)和 sdf 文件(时序约束文件)进行导出,采用 Formality 形式验证工具对综合后的网表进行逻辑的验证,形式验证相对于时序仿真的优势在于,能够以较短的时间消耗比较综合前、后电路逻辑功能是否一致.

3.2 编码电路的布局、布线

使用 Soc Encounter 软件,导入逻辑综合后的 verilog 文件和 sdf 文件,完成版图 GDS II 的设计,设计过程需要针对版图的面积、时序进行分析,采取面积约束、优化布局布线、时钟数综合等方法达到合理的版图设计,随后完成版图的 LVS 比较和 DRC 规则检查,得到的版图文件如图 5 所示.

3.3 编码电路的时序仿真

上面提到过对于验证电路功能是否与原始代码一致可以采用形式验证工具 Formality. 而本设计是优化的并行电路结构,其结构简单,使用的逻辑单元较小,时序仿真的耗时可以接受,因此可以采用 Modelsim 对编码电路进行时序仿真加以验证,时序仿真的结果如图 6 所示.

通过比较功能仿真和时序仿真结果,可以证明所设计芯片完全能够实现 CRC32 的并行编码功能.

4 结语

在研究 CRC 算法及编码原理的基础上,从代码编写到 GDS II 版图生成,完成了数字集成电路前端到后端的整个设计流程;所实现的编码芯片能够以高效的速率完成编码工作. 此外,本文介绍的芯片设计方法对了解和掌握数字芯片设计思想起到一定的指导作用.

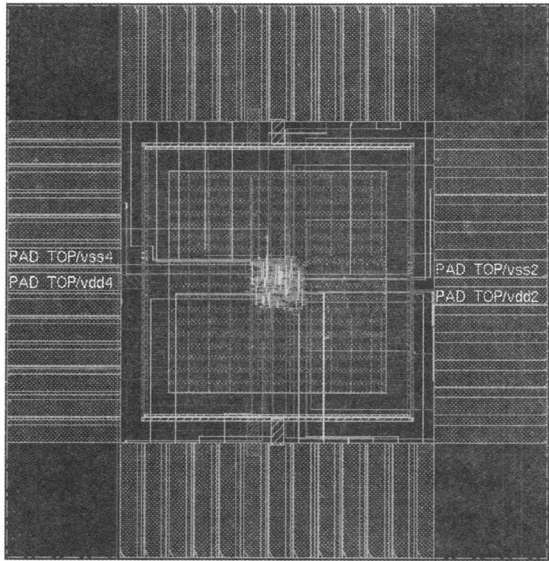


图 5 并行 CRC32 编码芯片版图

Fig. 5 Parallel CRC32 encoder chip layout

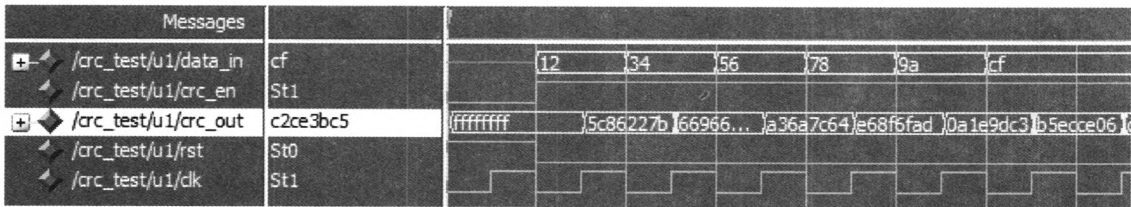


图 6 并行 CRC32 编码芯片时序仿真结果

Fig. 6 Timing simulation result of Parallel CRC32 encoder chip

参 考 文 献:

[1] 张树刚,张遂南,黄士坦. CRC 校验码并行计算的 FPGA 实现[J]. 计算机技术与发展,2007,17(2):56 - 62.
ZHANG Shugang,ZHANG Suinan,HUANG Shitan. CRC Parallel Computation Implementation on FPGA[J]. Computer Technology and Development,2007,17(2):56 - 62.

[2] 樊昌信,张甫翊,徐炳祥,等. 通信原理[M]. 北京:国防工业出版社,2001.

[3] SHIEH M D,SHEU M H,CHEN C H,et al. A systematic approach for parallel CRC computations [J]. Journal of Information Science and Engineering,2001,17:445 - 461.

[4] RAMABADRAN T V,GAITONDE S S. A tutorial on CRC computations[J]. IEEE Micro,1988(8):62 - 75.

[5] 夏宇闻. Verilog 数字系统设计教程[M]. 北京:北京航空航天大学出版社,2003.

[6] 韩雁,洪慧,马绍宇,等. 集成电路设计制造中 EDA 工具实用教程[M]. 杭州:浙江大学出版社,2007.

(责任编辑:孟素兰)