

UART_16C554的设计

郝天发,孙轶君,胡子阳

(中国电子科技集团公司第四十七研究所,沈阳 110032)

摘要:介绍了 UART_16C554可复用 IP核的设计,通过对器件进行结构划分,RTL级代码的设计,功能仿真,FPGA原型的验证,最终形成了一个与硬件 16C554完全兼容的可复用 IP核。

关键词:设计复用;异步通信;接收;发送

中图分类号: TN47 **文献标识码:** B **文章编号:** 1002 - 2279 (2006) 03 - 0019 - 03

Design of UART_16C554

HAO Tian - fa, SUN Yi - jun, HU Zi - yang

(The 47th Research Institute of China Electronics Technology Group Corporation, Shenyang 110032, China)

Abstract: In this paper, a reuse IP core design of UART_16C554 is described With the architecture design, RTL - level design, functional simulation, and FPGA prototype verification, the UART_16C554 is a reuse IP core that fully compatible with de - facto 16C554.

Key words: Reuse; A synchronous communication; Receiver; Transmitter

1 引言

在许多微机系统和信号处理系统中,异步通信器件 16C554的应用较为广泛,而随着 SOC技术的发展,设计开发可复用的 UART_16C554模块是比较有实际意义的。UART_16C554软核是与标准器件 16C554完全相匹配的通用异步接收器和发送器。UART_16C554由四路异步通信单元(16C550)构成,每个通路都可以执行从外设或调制解调器接收数据的串行—并行转换和从 CPU发送数据的并行—串行转换,并可以通过编程控制波特率、字长、奇偶校验等。UART_16C554的接收器和发送器还有 16字节的 FIFO,可以被配置到 FIFO工作方式。

2 功能说明

UART_16C554是由四路异步通信单元加上控制逻辑、中断逻辑、调制逻辑等构成,并具有如下主要性能:

- 在 FIFO 模式,每个通道的发送器和接收器有 16字节的 FIFO缓冲,以减少 CPU中断数目。
- 在 16C450 模式,保持和移位寄存器不需要 CPU和串行数据间的精确同步。
- 可编程的波特率发生器,允许对输入时钟 $1 \sim (2^{16} - 1)$ 分频和产生 16 的倍数的内部时钟。

- 对串行数据增加或删除标准异步通信位(起始位、停止位、校验位)。

- 可编程的串口特性: 5到 8位的字符; 奇校验、偶校验或无校验; 1, 1.5或 2位停止位; 波特产生 (DC ~ 1Mbps)。

- 错误起始位检测。

- 完全状态报告性能。

- 内部诊断性能: 通信链接错误隔离的循环返回控制; 打断、校验、溢出、帧错模拟。

3 结构设计

UART_16C554的结构如图 1所示(以其中一路为例)。

3.1 内部寄存器的设计

每个通道的内部寄存器存储 3种类型的信息: 控制、状态和数据。内部寄存器的寻址方式和名称及缩写见表 1。

发送保持寄存器和接收缓冲寄存器为保持 5 ~ 8位数据的寄存器,在不足 8位的数据被传输时,数据在最低位右对齐,数据的第 0位始终是接收和发送的第一位串行数据。

使用者通过 CPU在相应寄存器写入控制字来编程,读状态寄存器反馈操作情况。串行通道可以通过控制寄存器 LCR, IER, DLL, DLM, MCR和 FCR

进行编程。这些控制字定义了字符的长度、停止位数、波特率和调制解调接口。控制寄存器可以按任

何顺序写入,但 IER 应是最后写入的,因为它控制中断使能。

表 1 寄存器选择方式

DLAB	A2	A1	A0	读 (DR = 0)	写 (DW = 0)
0	0	0	0	接收缓冲寄存器 (RBR)	发送保持寄存器 (THR)
0	0	0	1		中断使能寄存器 (IER)
X	0	1	0	中断识别寄存器 (IR)	FIFO 控制寄存器 (FCR)
X	0	1	1		线路控制寄存器 (LCR)
X	1	0	0		调制解调控制寄存器 (MCR)
X	1	0	1	线路状态寄存器 (LSR)	
X	1	1	0	调制解调状态寄存器 (MSR)	
X	1	1	1	中间寄存器 (SCR)	中间寄存器 (SCR)
1	0	0	0		除数锁存器低位 (DLL)
1	0	0	1		除数锁存器高位 (DLM)

注: X 为无关值, 0 为低电平, 1 为高电平; 串行通道在片选为低时可存取; DLAB 为除数锁存器访问位, 即 LCR7;

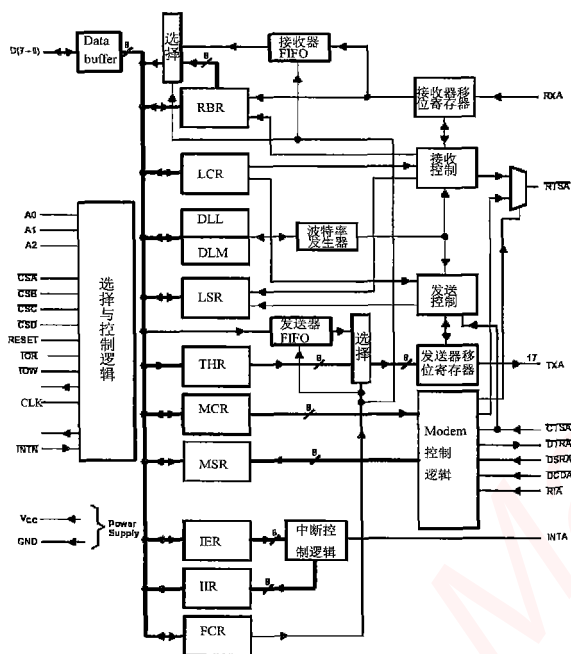


图 1 UART_16C554 的结构

3.2 可编程的波特率发生器设计

每个串行通道都包含一个可编程的波特率发生器,可以对时钟进行 $1 \sim 2^{16} - 1$ 任意分频。两个 8 位的除数锁存器存储了 16 位 2 进制格式的除数。这两个除数锁存器必须在初始化过程被载入,在任一个除数锁存器载入后,一个 16 位的波特率计数器立即被装载,这可以防止初始载入时的长计数。波特率发生器可利用三种通用的频率 (1.8432MHz, 3.072MHz 和 8MHz) 之一产生标准波特率 (50bps ~ 512kbps)。

3.3 接收器的设计

当串行异步数据输入到 RX_x 端口,接收器持续搜寻高到低的转变,当转变被检测到后,一个计数器被复位并且在最佳点即每一位的中心,对输入数据位进行采样。 RX_x 在采样点持续为低电平时起始位为有效,起始位的检验是为了防止由于 RX_x 的低噪

尖峰而导致接收器接收错误的字符。

每个字符的数据位长由 LCR0 和 LCR1 控制;校验检查、产生和奇偶由 LCR3 和 LCR4 控制。接收器的状态提供到 LSR。当一个包含校验位和停止位的完整字符被接收时,数据接收指示 LSR0 被置为 1, CPU 读取 RBR 的数据将清除 LSR0。如果在一个新字符从 RSR 传输到 RBR 之前, RBR 中的字符未被读取,将产生溢出错误,溢出指示位 LSR1 被置为 1。如果存在校验错, LSR2 被置为 1。如果停止位未检测到,帧错指示位 LSR3 被置为 1。在 FIFO 模式,数据字符和相关的错误位被存储于接收 FIFO。

3.4 中断的设计

UART_16C554 共有四个中断优先级,并可以在 IR 的位 3 ~ 位 1 读出: 011 为优先级 1——接收线路状态; 010 为优先级 2——接收数据可用; 110 为优先级 2——字符暂停指示; 001 为优先级 3——THR 空; 000 为优先级 4——调制解调状态。如果中断挂起,则 IR 的位 0 被清零。

3.5 复位的设计

在上电后,复位输入信号 (Reset) 应保持 1 微秒高电平,以使电路复位到一个空闲模式直到初始化。复位对寄存器和信号的影响见表 2。

3.6 RXRDYN 和 TXRDYN 操作

在模式 0, RXRDYN 在接收 FIFO 不空时有效 (低), 在 FIFO 空时被释放 (高), 这样, 接收 FIFO 在 RXRDYN 有效时被读取; TXRDYN 在发送 FIFO 空时有效 (低), 在 FIFO 内至少有一字节时被释放 (高), 发送 FIFO 在 TXRDYN 有效时被写入 16 字节。

在模式 1, RXRDYN 在接收 FIFO 填充达到触发级别或一个字符暂停产生时有效 (低), 在 FIFO 空时被释放 (高), 在这种模式, DMA 设备可读取多个字符, 减少了产生中断的次数; TXRDYN 在发送 FIFO 不满时有效 (低), 此时, 发送 FIFO 可写入另

一字节。

L16C554 内部四个 ACE 的 RXRDYN 和 TXR-

DYN 输出进行与操作后的组合信号为外部的 RXR-
DYN 和 TXRDYN 输出。

表 2 复位对寄存器和信号的影响

寄存器 信号	复位控制	复位状态
IER	Reset	所有位均清零 (0—3 强制, 4—7 永久)
IR	Reset	位 0 置 1; 位 1, 2, 3, 6, 7 清零; 位 4, 5 永久清零
LCR	Reset	所有位均清零
MCR	Reset	所有位均清零 (5—7 永久)
FCR	Reset	所有位均清零
LSR	Reset	位 5, 6 置 1; 其余位均清零
MSR	Reset	位 0—3 清零; 位 4—7 为输入信号
TXx	Reset	高
中断 (接收器错)	读 LSR / Reset	低
中断 (接收数据准备好)	读 RBR / Reset	低
中断 (THR 空)	读 IR / 写 THR / Reset	低
中断 (modem 状态改变)	Reset	低
请求发送	Reset	高
数据终端准备好	Reset	高

4 验证方案

在 RTL 级设计完成后,对 UART_16C554 进行了功能验证和 FPGA 的原型验证。

4.1 功能验证

利用 KRISC 的 IP 核作为 CPU,加上 RAM、ROM 模块搭建了一个对 UART_16C554 进行功能验证的软测试平台。测试程序采用 KRISC 的汇编语言进行编写,覆盖了全部的 16C554 功能。对 UART_16C554 的功能验证是利用 ModelSim 仿真工具进行的。对 RTL 级代码利用 FCII 综合工具进行综合,并对综合后的网表进行了带 SDF 时序延迟反标的仿真(后仿真),验证了 UART_16C554 的功能和时序。

4.2 FPGA 原型验证

首先在 Aptix 硬件仿真器上搭建测试平台,测试平台由 KRISC 的 IP 核,16C554 芯片,双口 ROM (用于存放程序)和 RAM 构成,通过编写的测试程序对 16C554 的芯片进行测试采样,并记录下测试波形和码点。然后将 UART_16C554 下载到 FPGA 中,在 Aptix 可编程通道中修改连接关系,使 UART_16C554 代替 16C554 芯片,通过测试程序进行对比

(上接第 18 页)

的铝丝。根据计算该直径的铝丝可以承载 12 安培的直流电流,完全可以满足本电路的需要。

4 结束语

通过以上的电路设计和工艺设计,使该专用大功率驱动电路能够满足高可靠使用的需要,实现了小型化、轻型化、集成化的目的。

验证。在我们的测试平台上,建立以 NC - VLOG 为仿真平台的系统仿真环境,向子模块提供必要的激励信号。信号由 Aptix 的 MVP 向硬件及 FPGA 提供,并由 MVP 回采以波形方式显示在波形窗口,实现了软硬件的无缝互连。最终证明我们设计的 UART_16C554 与 16C554 芯片完全兼容。

5 总结

UART_16C554 IP 核通过了功能验证和 FPGA 的原型验证,可以与实际的 16C554 芯片完全匹配。在设计过程中,我们从体系结构入手,对电路进行了功能块的划分,RTL 级设计遵循了设计复用的原则,使得我们最终的设计能够作为可复用的 IP 核,验证过程使用了软硬件协同仿真技术,为今后在设计复用和软硬件协同仿真上积累了大量有益的经验。

参考文献:

- [1] TL16C554 A synchronous Communications Element [Z]. Texas Instruments, 2001 - 08
- [2] Rochit Rajsuman. SoC 设计与测试 [M]. 北京:北京航空航天大学出版社, 2003
- [3] 任艳颖,王彬. IC 设计基础 [M]. 西安:西安电子科技大学出版社, 2003

参考文献:

- [1] 中国集成电路大全编委会. 电力电子技术与运动控制系统 [M]. 北京:国防工业出版社, 1995
- [2] 沈耀忠,任志纯,罗毅. CMOS 功率场效应晶体管原理及应用 [M]. 北京:电子工业出版社, 1995
- [3] 田民波. 电子封装工程 [M]. 北京:清华大学出版社, 2003
- [4] 郑福员,周立飞,虎轩东. 厚薄膜混合集成电路 - 设计、制造和应用 [M]. 北京:科学出版社, 1984

嵌入式资源免费下载

总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)

VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)

11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)

Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 CC++ 语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)
17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)

Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)

RT Embedded <http://www.kontronn.com>

5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)

PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)

ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)

WeChat ID: kontronn

10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)

Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)