

文章编号: 1007- 2985(2009) 06- 0066- 04

基于 CPLD 的 SAA7113 的初始化及其控制设计*

曾庆立, 孟凡斌, 陈善荣

(吉首大学物理科学与信息工程学院, 湖南 吉首 416000)

摘 要: 为了实现对视频图像的采集和多格式输出, 通常使用低速存储器存储所采集的图像信息. 笔者介绍了视频解码芯片 SAA7113 的特点及其应用, 研究了基于 CPLD 对 SAA7113 的硬件电路配置结构的实现, 并给出了使用 VHDL 语言通过 I²C 总线进行初始化控制的编程方法, 从而实现了视频采集和使用 8 MHz/s 的低速存储器存储所采集的图像信息, 并能根据需要由多种格式输出.

关键词: VHDL; SAA7113; I²C 总线; CPLD

中图分类号: TP391.7

文献标识码: A

SAA7113 是飞利浦公司视频解码系列芯片的一种, 在很多视频产品, 如电视卡、MPEG-2、MPEG-4 中都有应用. SAA7113 的主要作用是把输入的模拟视频信号解码成标准的 VPO 数字信号, 相当于一种 A/D 器件. 7113 兼容全球各种视频标准, 在中国应用时必须根据中国的视频标准来配置内部的寄存器, 即初始化, 否则 7113 就不能按要求输出相应信号, 可以说对 7113 进行研发的主要工作就是如何初始化. 对 7113 初始化需要通过 I²C 总线进行, 笔者给出了基于 CPLD 且使用 VHDL 语言编程控制的设计实例.

1 视频解码芯片 SAA7113 的工作原理

视频解码芯片 SAA7113, 它可以输入 4 路模拟视频信号, 通过内部寄存器的不同配置可以对 4 路输入进行转换, 输入可以为 4 路 CVBS 或 2 路 S 视频(Y/C) 信号, 输出 8 位 VPO 总线, 为标准的 ITU656、YUV 4: 2: 2 格式. 7113 兼容 PAL、NTSC、SECAM 多种制式, 可以自动检测场频适用的 50 或 60 Hz, 可以在 PAL、NTSC 之间自动切换. 7113 内部具有一系列寄存器, 可以配置为不同的参数, 对色度、亮度等的控制都是通过对相应寄存器改写不同的值, 寄存器的读写需要通过 I²C 总线进行. 7113 的模拟与数字部分均采用 + 3.3 V 供电, 数字 I/O 接口可兼容 + 5 V, 正常工作时功耗 0.4 W, 空闲时为 0.07 W. 7113 需外接 24.576 MHz 晶体, 内部具有锁相环(PLL), 可输出 27 MHz 的系统时钟. 芯片具有上电自动复位功能, 另有外部复位管脚(CE), 低电平复位, 复位以后输出总线变为三态, 待复位信号变高后自动恢复, 时钟丢失、电源电压降低都会引起芯片的自动复位.

2 SAA7113 的硬件配置电路

SAA7113 的硬件配置电路包括电源、信号输入、时钟发生、I²C 总线和数据输出 5 部分. SAA7113 的电源为数字电源和模拟电源 2 个, 两者均应在引脚附近接合适的配耦电容. 相应的地平面也分为数字地和模拟地, 布线时 2 个地平面应通过磁珠等实现单点接通, 从而减小数字信号与模拟信号的相互干扰. 信号输入部分由视频输入专用插座和相应的 RC 输入回路构成. SAA7113 支持 4 路复合视频或 2 路 S 视频输入, 本电路只使用了 1 路复合视频输入. 时钟发生电路包括外接 24.576 MHz 晶振和内部振荡电路与及内部锁相环构成, 为内部各功能电路提供符合时序要求的时钟. I²C 总线包括数据线 SDL 和时钟线 SCL, 根据 I²C 总线协议两者均应接合适的上拉电阻, 再与 CPLD 的通用 I/O 口相接. 数据输出部分由 8 位数据总线 VP[7..0]、27 MHz 的总线时钟信号 LLC 和可配置输出线 RTS1、RTCO 构成. 硬件电路如图 1 所示.

3 基于 CPLD 的 SAA7113 初始化和控制

CPLD 对 SAA7113 的控制包括 2 个部分: CPLD 通过 I²C 总线对 SAA7113 初始化, 即对寄存器配置合适的参数, 使其

* 收稿日期: 2009- 08- 11

作者简介: 曾庆立(1975-), 男, 湖南蓝山人, 吉首大学物理科学与信息工程学院副教授, 主要从事电子电路研究.

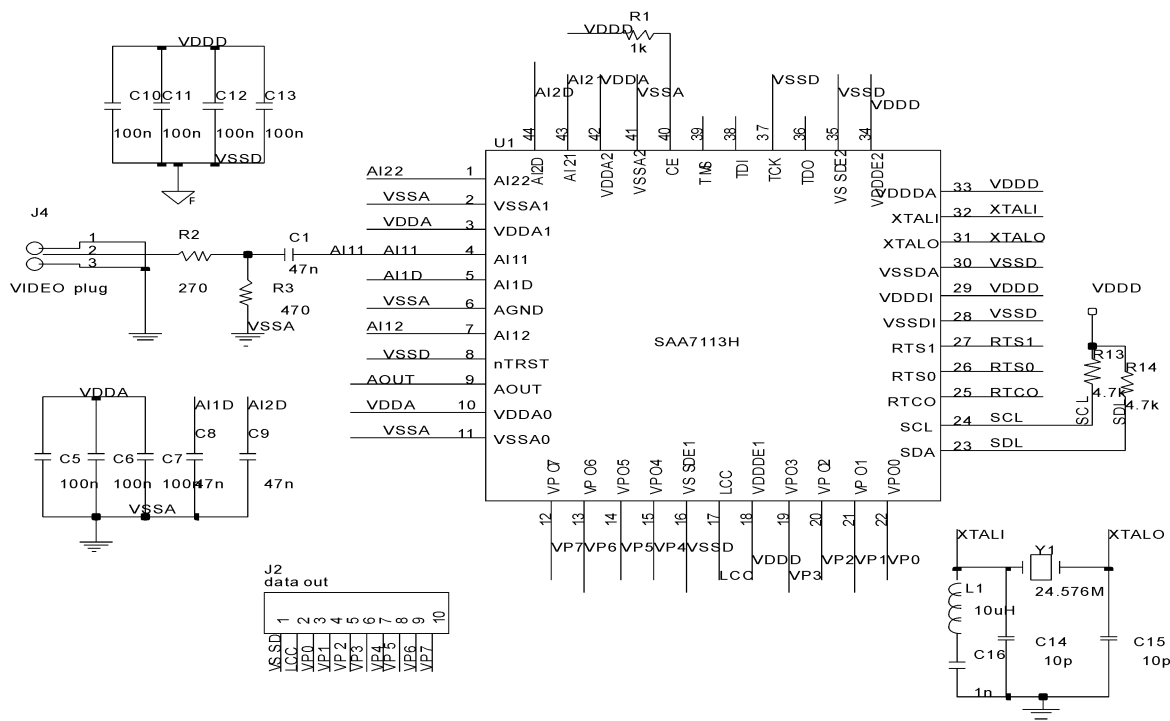


图 1 SAA7113 硬件配置

能够有符合要求的输出;接收 SAA7113 输出的原始数据进行转换后存储. SAA7113 输出数据的时钟为 27 MHz, 要完成数据转换的存储由单片机等很难达到速度要求, 使用 CPLD 通过 VHDL 编程能较好的完成此项工作.

3.1 SAA7113 初始化

CPLD 的寄存器配置通过 I²C 总线来进行, 很多可以控制 I²C 总线的器件都可以作为主器件对 7113 进行初始化, 笔者以 CPLD 为核心芯片, 对 7113 进行初始化. 使用 CPLD 通过 VHDL 语言实现对 SAA7113 的控制能准确控制各信号的时序, 并且有良好的可移植性, 适应高速控制需要. CPLD 和 7113 的硬件连接非常简单, 只要把 CPLD 的 2 个 I/O 口直接和 7113 的 SCL、SDA 管脚相连, 再加上上拉电阻即可. 用 CPLD 初始化 7113 的主要任务是程序的编写, 首先要熟悉 I²C 总线协议, 根据 I²C 总线的原理写出启动、停止、应答信号等的进程, 由子程序再写出发送、接收 1 个字节的程序, 然后根据 7113 的寄存器操作格式写出读写寄存器的程序, 最后根据以上的子程序写出初始化 7113 的程序段. 对 7113 的控制一般是改变色度、亮度等指标以及输出管脚的输出信号, 这可以通过修改相应寄存器的值来完成, 程序上写出“读写命令”即可. 程序在复位后先在进程 delay 进行必要的延时, 进程 clkgen 产生 I²C 总线的时钟信号, 进程 config 将指定数据写入 SAA7113 的相应寄存器, 从而完成对 SAA7113 的配置. 下面给出配置的关键程序:

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY saa7113 IS
    PORT ( clk:IN std_logic;
          scl:OUT std_logic;
          sda:INOUT std_logic);
END saa7113;
ARCHITECTURE translated OF saa7113 IS
    SIGNAL RST:STD_LOGIC;
    SIGNAL DATA_IN:STD_LOGIC_VECTOR(7
    DOWNT0 0);
    SIGNAL seg_data_buf:std_logic_vector(3 DOWNT0
    0);
    SIGNAL cnt_scan:std_logic_vector(13 DOWNT0 0);
    SIGNAL sda_buf:std_logic;
    SIGNAL link:std_logic;
    ...
    delay:PROCESS(clk, rst)
    ...
    END PROCESS;
    Clkgen:PROCESS( clk, rst)
    ...
    END PROCESS;
    Config1:PROCESS( clk, rst)
    BEGIN
        IF (NOT rst = 1) THEN
            start_delaycnt <= 0;
            main_state <= "00";
            i2c_state <= ini;
            inner_state <= start;
            scl_xhdl1 <= 1;

```

```

    sda_buf <= 1;
    link <= 0;
ELSIF (clk'event and clk= 1) THEN
CASE main_state IS
    WHEN "00" = > -- 等待读写要求
        writeData_reg <= data_in;
        scl_xhdl1 <= 1;
        sda_buf <= 1;
        link <= 0;
        inner_state <= start;
            i2c_state <= ini;
IF (cnt_delay = "00000000000000000000") THEN
start_delaycnt <= 1;
    ELSE
IF (cnt_delay = "11000011010100000000") THEN
    start_delaycnt <= 0;
        main_state <= "01";
    PRO <= 1;
        END IF;
    END IF;
WHEN "01" = > -- 向 EEPROM 写入数据
    IF (phase0 = 1) THEN
        scl_xhdl1 <= 1;
    ELSE
        IF (phase2 = 1) THEN
            scl_xhdl1 <= 0;
        END IF;
    END IF;
CASE i2c_state IS
WHEN ini = > -- 初始化 EEPROM
    CASE inner_state IS
        WHEN start = >
            IF (phase1 = 1) THEN
                link <= 1;
                sda_buf <= 0;
            END IF;
            IF ((phase3 AND link) = 1) THEN
                inner_state <= first;
                sda_buf <= 0;
            link <= 1;
            END IF;
        WHEN first = >
            IF (phase3 = 1) THEN
                sda_buf <= 1;
                link <= 1;
                inner_state <= second;
            END IF;
        WHEN second = >
            IF (phase3 = 1) THEN
                sda_buf <= 0;
                link <= 1;
                inner_state <= third;
            END IF;
            ...
        WHEN eighth = >
            IF (phase3 = 1) THEN
                link <= 0;
                inner_state <= ack;
            END IF;
        WHEN ack = >
            IF (phase0 = 1) THEN
                sda_buf <= sda;
            END IF;
            IF (phase1 = 1) THEN
                IF (sda_buf = 1) THEN
                    main_state <= "00";
                END IF;
            END IF;
        IF (phase3 = 1) THEN
            link <= 1;
            sda_buf <= addr(7);
            inner_state <= first;
            i2c_state <= sendaddr;
        END IF;
    WHEN OTHERS = >
        NULL;
    END CASE;
WHEN sendaddr = >
    CASE inner_state IS
        WHEN first = >
            IF (phase3 = 1) THEN
                link <= 1;
                sda_buf <= addr(6);
                inner_state <= second;
            END IF;
            ...
        WHEN eighth = >
            IF (phase3 = 1) THEN
                link <= 0;
                inner_state <= ack;
            END IF;
        WHEN ack = >
            IF (phase0 = 1) THEN
                sda_buf <= sda;
            END IF;
            IF (phase1 = 1) THEN
                IF (sda_buf = 1) THEN
                    main_state <= "00";
                END IF;
            END IF;

```

```

END IF;
END IF;
IF (phase3 = 1) THEN
link <= 1;
sda_buf <= writeData_reg(7);
inner_state <= first;
i2c_state <= write_data;
END IF;
WHEN OTHERS = >
NULL;
END CASE;
WHEN write_data = >
CASE inner_state IS
WHEN first = >
IF (phase3 = 1) THEN
link <= 1;
sda_buf <= writeData_reg(6);
inner_state <= second;
END IF;
WHEN second = >
IF (phase3 = 1) THEN
link <= 1;
sda_buf <= writeData_reg(5);
inner_state <= third;
END IF;
...
WHEN eighth = >
IF (phase3 = 1) THEN
link <= 0;
inner_state <= ack;
END IF;
WHEN ack = >
IF (phase0 = 1) THEN
sda_buf <= sda;
END IF;
IF (phase1 = 1) THEN
IF (sda_buf = 1) THEN
main_state <= "00";
END IF;
ELSE
IF (phase3 = 1) THEN
link <= 1;
sda_buf <= 0;
inner_state <= stop;
END IF;
END IF;
WHEN stop = >
IF (phase1 = 1) THEN
sda_buf <= 1;
END IF;
IF (phase3 = 1) THEN
main_state <= "00";
PRO <= 0;
END IF;
WHEN OTHERS = >
NULL;
END CASE;
WHEN OTHERS = >
main_state <= "00";
END CASE;
WHEN OTHERS = >
NULL;
END CASE;
END IF;
END PROCESS;
END translated;

```

3.2 SAA7113 的数据转换存储

SAA7113 输出数据的时钟为 27 MHz, 大部分 RAM 工作频率低于这个频率, 而高速 RAM 价格高. 为了能使用 HM628512 等 RAM 存储数据, 可以由多片 RAM 构成 32 位数据总线, 先由 CPLD 实时接收 SA7113H 的 8 位数据, 连续接收 4 个字节后组成 32 位数据一次存入存储单元, 这样存储系统工作频率可降为数据时钟的 1/4 也能实现实时数据存储. 同时 CPLD 逻辑较丰富, 在存储前可以根据需要实现数据格式转换和提取, 如将 YUV 数据转换为 RGB 数据, 提取偶数场或奇数场等功能, 比用其他控制方式有明显的优势.

4 结语

SAA7113 应用在很多产品中, 其初始化与控制的原理都一样, 笔者设计的程序段经过实际应用可以保证 7113 正常工作, 实现对视频通道 1 的 PAL 信号采集, 其寄存器设置参数与控制方法可以被借鉴或直接应用, 同时实现视频数据在低速 RAM 中的存储.

参考文献:

- [1] 刘永贵. 视频采集系统应用方案的研究 [J]. 现代电子技术, 2007(18): 63-64.
- [2] 王志华. 数字集成系统的结构化设计与高层次综合 [M]. 北京: 清华大学出版社, 2007.

Initialization and Control of SAA7113 Based on CPLD

ZENG Qing-li, MENG Fan-bing, CHEN Shan-rong

(College of Physics Science & Information Engineering, Jishou University, Jishou 416000, Hunan China)

Abstract: In order to achieve the right video image acquisition and multi-format output, images collected by low-speed memory are usually used. This article describes the characteristics and application of SAA 7113 video decoder chip. Implementation of the SA 7113 hardware circuit configuration structure is researched based on CPLD. Through I²C bus, the VHDL language is used to initialize the control of programming. Therefore, video acquisition, image information collection by 8 MHz/s low-speed memory, and multi-format output can be achieved.

Key words: VHDL; SAA7113; I²C bus; CPLD

(责任编辑 陈炳权)

(上接第 62 页)

- [20] DONG J J, SANKEY O F, MYLES C M. Theoretical Study of the Lattice Thermal Conductivity in Ge Framework Semiconductors [J]. Phys. Rev. Lett., 2001, 86(11): 2 361- 2 364.
- [21] DAI Z H, NI J. Electron Transport in Multi-Terminal Quantum Chain Systems Based on the Green's Functions [J]. Acta. Phys. Sin., 2005, 54(7): 3 342- 3 345.
- [22] LU X, CHU J H, SHEN W Z. Modification of the Lattice Thermal Conductivity in Semiconductor Rectangular Nanowires [J]. J. Appl. Phys., 2003, 93(2): 1 219- 1 229.

Acoustic-Phonon Transmission and Thermal Conductance in a Width-Change Quantum Waveguide

YE Fu-qiu^{1,2}, PENG Xiao-fang²

(1. College of Physics Science & Information Engineering, Jishou University, Jishou 416000, Hunan China;

2. Department of Applied Physics, Hunan University, Changsha 410082, China)

Abstract: The authors investigate the effect of the value change of L on low temperature phonon transport in a width-change quantum waveguide by using the scattering-matrix method. The calculated results show that the transmission coefficient almost doesn't change when the width of the quantum waveguide changes small, but it will become large with the value increase of L when the width of the quantum waveguide changes more. And at low temperature, the thermal conductance almost doesn't change with the width-change of the quantum waveguide, but it will become large with the value increase of L when the temperature hoists.

Key words: phonon transport; thermal conductance; scattering-matrix method

(责任编辑 陈炳权)