

文章编号:1007-2780(2011)05-0640-06

RGB 格式数据向 BT. 656 视频标准转换的关键技术

张 倩, 胡健生

(武警工程学院, 陕西 西安 710086, E-mail: zhangqianjy@163.com)

摘 要: 针对当前多数场合需要将 RGB 格式图像数据转换成 BT. 656 视频标准的现状, 提出了基于 FPGA 的实现方案, 分析了系统的各组成模块和关键技术。利用状态转移机制, 产生了 BT. 656 视频标准时序; 通过线性循环补偿法, 避免了 RGB 向 YUV 格式转换的颜色失真现象; 使用带有输入输出选择控制功能的 FIFO 电路, 解决了输入输出数据的速率匹配问题。仿真验证结果表明, 所提出的方案效果良好, 具有广阔的应用前景。

关 键 词: RGB; BT. 656; 状态转移; 线性循环补偿法; FIFO

中图分类号: TP911 **文献标识码:** A **DOI:** 10.3788/YJYXS20112605.0640

Key Technology of Video Format Convert From RGB to BT. 656

ZHANG Qian, HU Jian-sheng

(Engineering Institute of CAPF, Xi'an 710086, China, E-mail: zhangqianjy@163.com)

Abstract: Aimed at the necessary of digital video convert from RGB to BT. 656 on many occasion, a realization scheme based on FPGA was put forward. The key technology and modules of system was analyzed. The status transfer mechanism was used to accomplish the production of BT. 656 video timing; The linear and circular compensation method was applied to avoid the color distortion when conversion from RGB to YUV; The FIFO circuit with function of input and output selection control was employed to resolve the speed matching between input data and output data. The emulation result shows good purpose while an application example is introduced to indicate its broad prospect in video signal conversion.

Key words: RGB; BT. 656; status transfer; linear and circular compensation method; FIFO

1 引 言

目前, 在许多嵌入式应用场合, 都需要将 RGB 格式的图像数据转换成 BT. 656 视频标准, 提供给摄像机接口进行后续的压缩、加密等功能处理^[1-2]。在 BT. 656 视频标准中, 时序代码是嵌入在数据中的, 而每个像素点的数据是按照 YUV 格式存放的, 因此转换不仅要完成视频时序的产生, 还需要完成色彩空间的转换; 而绝大多数的嵌入式处理器由于性能的限制, 不能实现这

一转换。针对此问题, 本文基于 FPGA 设计并实现了由 RGB 格式图像数据向 BT. 656 标准视频的转换。

2 系统总体设计

系统由前端输入单元、存储器单元和视频格式转换单元组成, 3 个单元通过系统的数据和地址总线进行数据传输, 如图 1 所示。前端输入单元是产生 RGB 格式图像的数据源, 可以是嵌入式处理器、专用的视频采集设备或其他可编程器件

等;存储器单元一般由两片存储器芯片(SRAM 或 SDRAM 等)组成,当前端输入单元向其中一片存储器写入 RGB 数据的同时,视频格式转换单元可以从另一片存储器中读取 RGB 数据,两片存储器的“乒乓”切换,解决了对数据同时读写冲突的问题^[3-4];视频格式转换单元是系统的核心,即 FPGA 芯片,通过对其进行编程,实现 RGB 数据向 BT. 656 格式视频的转换,并输出提供给用户使用。

视频格式转换单元采用模块化和层次化设计的方法,主要由时钟模块、存储器接口模块、RGB-YUV 转换模块和 BT. 656 视频时序产生模块组成。时钟模块主要利用 FPGA 内部的锁相环电路对输入的外部时钟源进行倍频,为系统提供一个稳定可靠的主时钟信号;存储器接口模块主要负责完成对外部 SRAM/SDRAM 等的读操作,将 RGB 格式的数据读入到 FPGA 内部;RGB-YUV 转换模块主要负责对视频数据进行 RGB 向 YUV 的转换,并存储在内部 FIFO 电路中;BT. 656 视频时序产生模块主要负责生成时序控制代码,并读取 FIFO 中的 YUV 格式的视频数据,最终产生标准的 BT. 656 视频格式的时序信号。

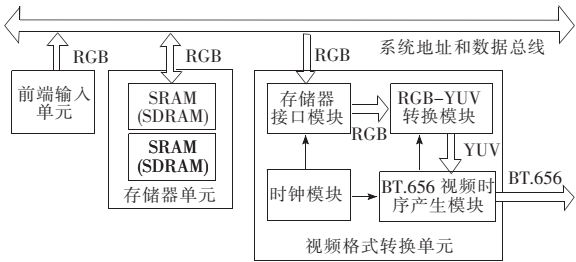


图 1 系统整体结构组成
Fig. 1 System configuration

3 关键问题的解决

3.1 BT. 656 视频格式时序的产生

BT. 656 为 YUV422 格式的视频数据定义了并行或串行接口,分辨率为 720×486 或 720×576 。BT. 656 并行接口使用 8 bit 或 10 bit 的复合 YUV 数据和 27 MHz 的时钟,不需使用传统的视频时序信号(HSYNC, VSYNC 和 BLANK),将时序代码嵌入到视频流中,可以减少芯片接口的引脚数量^[5]。本文采用的是 720×576 的 8 bit 格式,时序如图 2 所示。

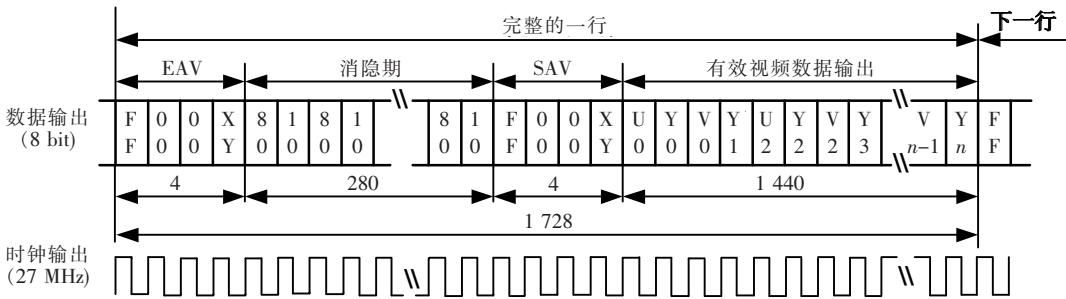


图 2 BT. 656 信号时序
Fig. 2 Timing signal of BT. 656

其中,SAV(Start active video)是有效视频数据的开始标识字段,EAV(End active video)是有效视频数据的结束标识字段,均为 4 个字节,前 3 个字节相同,最后 1 个字节(XY)的内容与当前时序相关,具体组成如表 1 所示。

选用 20 MHz 的外部时钟源,利用 FPGA 内部锁相环生成系统的主时钟 27 MHz,即 BT. 656 视频标准的像素时钟频率。在设计视

频时序信号时,设置一个 20 bits 像素时钟计数器的(pclk_counter),对像素时钟信号(pclk)进行计数;设置一个 4 bit 的状态转移标识变量status,根据 pclk_counter 的值确定状态值,在 10 个状态之间进行切换;设置一个 10 bit 的行计数器(line_counter),对输出视频的行进行计数。各状态维持的时间和执行的操作如表 2 所示。

表 1 SAV 和 EAV 最后一个字节组成
Table 1 The last byte of SAV and EAV

XY	D8	D7	D6	D5		D4	D3	D2	D1
Line	1	F	V	H(EAV)	H(SAV)	P3	P2	P1	P0
1~22	1	0	1	1	0				
23~310	1	0	0	1	0			$P3 = V \oplus H$	
311~312	1	0	1	1	0			$P2 = F \oplus H$	
313~335	1	1	1	1	0			$P1 = F \oplus V$	
336~623	1	1	0	1	0			$P0 = F \oplus V \oplus H$	
624~625	1	1	1	1	0				

表 2 时序控制代码产生的状态转移过程
Table 2 Status transfer course of timing control code production

状态	状态名称	持续时间	执行的操作
0	EAV 的第 1 个 Byte	0	(1)禁止对外部 SRAM 和内部 FIFO 的操作 (2)根据 line_counter 的大小,设置 XY 中“F”位 (3)code[7:0]输出“0xFF”
1	EAV 的第 2 个 Byte	1	(1)根据 line_counter 的大小,设置 XY 中“V”位 (2)code[7:0]输出“0x00”
2	EAV 的第 3 个 Byte	2	(1)根据 line_counter 的大小,判断完整一帧是否结束 (2)如果是完整的一帧,line_counter 清 0,否则,加 1 (3)Code[7:0]输出“0x00”
3	EAV 的第 4 个 Byte	3	(1)根据状态 0~3 计算的“F”、“V”和“H”的结果 (2)code[7:0]输出“0xXY”
4	生成随机数	4~283	code[7:0]输出“0x80,0x10,0x80,0x10”
5	SAV 的第 1 个 Byte	284	code[7:0]输出“0xFF”
6	SAV 的第 2 个 Byte	285	code[7:0]输出“0x 00”
7	SAV 的第 3 个 Byte	286	code[7:0]输出“0x00”
8	SAV 的第 4 个 Byte	287	(1)根据状态 0~3 计算的“F”、“V”和“H”的结果 (2)code[7:0]输出“0xXY”
9	有效视频数据输出	288~1 727	(1)读取内部 FIFO 中的 YUV 数据,按次序输出 (2)根据 pixel_counter 大小,判断完整一行是否结束

3.2 色彩空间的转换

BT. 656 视频标准采用的是 YUV422 格式,而输入的视频数据采用的是 RGB 格式,这里以 RGB565 为例,需要进行转换。常规的方法是采用低位补 0 的方法,先将 RGB565 格式转换成 RGB888 格式,再对 RGB888 进行矩阵运算,转换成 YUV422 格式。这种直接对低位进行补 0 的

方法未进行补偿数据,在精度上会有损失,容易造成色彩的失真。针对上述问题,这里采用循环线性补偿方法。首先,将原始数据填充至高位;对于未填充的低位,用原始数据的低位进行补偿;如果仍有未填充的位,继续使用原始数据的低位进行循环补偿。此方法可以很好地解决由低深度 RGB 格式向高深度 RGB 格式转换时的色彩失真

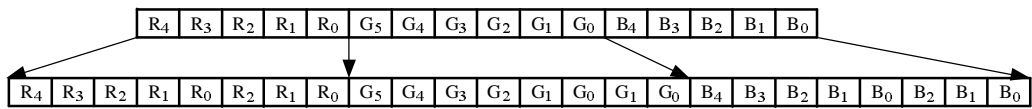


图 3 RGB565 向 RGB888 的线性循环补偿

Fig. 3 Linear and circular compensation method of RGB565 to RGB888

问题。本系统中 RGB565 向 RGB888 的线性循环补偿如图 3 所示。

由 RGB 转换成 YUV 的公式如下所示^[6]：

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ U = -0.147R - 0.289G + 0.436B \\ V = 0.615R - 0.515G - 0.100B \end{cases} \quad (1)$$

在公式(1)中,3 个分量的计算均需要乘法和加法运算,并且公式中均用到了小数,为了在 FP-GA 中实现数据转换,首先对系数进行放大,然后再对结果进行缩小,如公式(2)所示。

$$\begin{cases} Y = (306 + 601G + 117B)/1024 \\ U = (-151R - 296G + 446B)/1024 \\ V = (630R - 527G - 103B)/1024 \end{cases} \quad (2)$$

这样对公式处理后,去除了小数运算,并且最后的除法运算,可以通过移位操作来实现,完全可以在 FPGA 中实现视频数据色彩空间的转换。

3.3 输入输出数据速率的匹配

存储器接口模块每次读取一个 RGB565 格式的像素点数据,送到 RGB-YUV 模块进行色彩空

间的转换,而对于输出端的 YUV422 格式,在每个时钟周期内只能输出某个像素点的 YUV 中的一个分量,且相邻的两个像素点 U 分量和 V 分量是共享的,因此需要解决输入输出数据速率的匹配问题。FIFO 电路是解决异步时钟同步的方法之一^[7-8]。

系统采用带有输入输出选择功能的 FIFO 组来实现,即像素点的 Y 分量、U 分量和 V 分量分别存放在 3 个 FIFO 队列中,FIFO 组的输入和输出都具有选择判断功能,如图 4 所示。一方面,由于在输出的 YUV422 格式的视频数据中相邻两个像素点的 U 分量和 V 分量是共享的,每个像素点的 Y 分量是独立的,因此可以通过选择控制,对输入的 RGB565 格式的两个相邻像素点只需进行一次 U 分量和 V 分量的计算,这样不仅可以节约 FPGA 内部的资源,同时也提高了处理器速率。另一方面,对转换后的 3 个 FIFO 队列中 YUV 分量,在每个输出时钟内,也通过一个选择判断模块,按照 YUV422 的顺序依次输出。

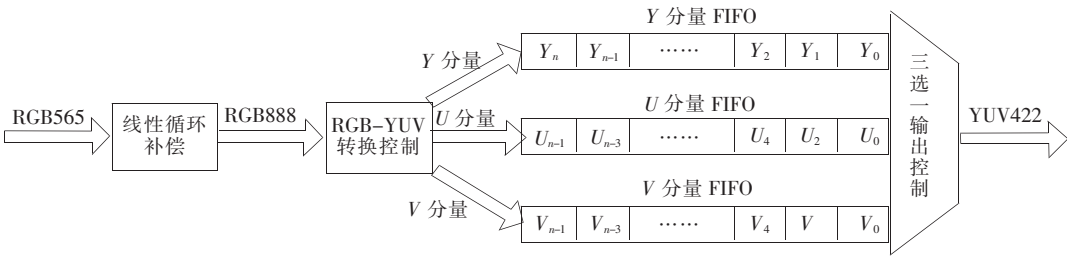


图 4 输入输出选择控制功能的 FIFO 组

Fig. 4 FIFO group with selection and control of input and output

RGB-YUV 的转换控制实现了对每个输入像素点进行 Y 分量计算,每隔一个像素点进行一次 U 分量和 V 分量的计算,在程序设计时,只需设置一个二进制的标识变量 yuv_pixel。当 yuv_pixel=0 时,对当前像素点进行 Y、U 和 V 3 个分量的计算;当 yuv_pixel=1 时,只进行 Y 分量的

计算;每完成一个像素点的处理后,对 yuv_pixel 进行取反运算。

而对于三选一输出控制,需要设置两个标识变量,yuv_select0 取值范围为 0~2,yuv_select1 取值范围是 0~1。选择控制程序流程如图 5 所示。

真现象;使用带有输入输出选择控制功能的 FIFO 电路,解决了输入输出数据的速率匹配问题。该系统具有很强的可移植性和可扩展性,与其他平台相结合,可以实现多路视频的采集和远传的功能。

参 考 文 献:

[1] 向守坤,黄启俊,江冠群,等. 基于 FPGA 的 ITU-RBT. 656 数字视频转换接口系统 [J]. 电子测量技术,2009,32(4): 113-117.

[2] 刘潇,高峻. 采用 FPGA 通过 BT. 656 接口实现传输 4 路视频流的方法 [J]. 电子工程师,2006,32(4):34-38.

[3] 朱耀东,经亚枝,张焕春. 基于 FPGA 的 LCD&VGA 控制器设计 [J]. 电子技术应用,2002,28(11):44-46.

[4] 李广军,佟国香,谢岑芬. 图形式 LCD 控制器的 VHDL 设计与实现 [J]. 半导体技术,2001,26(5):46-49.

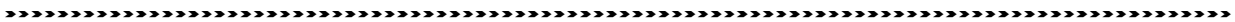
[5] Intersil Corporation. BT. 656 Video Interface for ICs Application Note [S]. Milpitas, USA: Intersil Corporation,2002.

[6] 刘清堂,王忠华,陈迪. 数字媒体技术导论 [M]. 北京:清华大学出版社,2008.

[7] 付先成,邹雪城,雷镒铭. LCD 控制器中央控制单元的设计 [J]. 微电子学与计算机,2006,23(6):1-4.

[8] 于海,樊晓桢. 基于 FPGA 异步 FIFO 的研究与实现 [J]. 微电子学与计算机,2007,24(3):210-216.

[9] 胡健生,臧晓昱. 嵌入式多路视频采集显示系统设计 [J]. 液晶与显示,2010,25(12):833-835.



物理量量值表示法

物理量量值必须用阿拉伯数字表示。按规定,阿拉伯数字后的单位必须使用我国法定计量单位,在非普及性的科技书刊中,还应使用单位的国际符号。例如:1 m, 5 d,12 m²,4 A,2 000 m³/s,273 K,100~150 kg,30 多 V,400 余 t 等。

在非普及性的科技书刊中,非量值情况下也使用单位的国际符号。例如:“速度的单位为 m/s”,“每 mL 溶液……”等,只要不会引起误解,单位的国际符号可以用于一切使用单位的场合。

对于非物理量的量词前面的数字,一般也应使用阿拉伯数字,包括整数一至十。例如:1 支铅笔,2 台电视,3 组数据。

组合单位加词头的规则

a. 通过单位相乘构成的组合单位,通常词头加在第 1 个单位前面。例如:N·s 加词头 k,应为 kN·s。但常见的法定单位中有 2 个单位例外。一个是电阻率的单位 $\Omega \cdot m$,通常写成 $k\Omega \cdot m$, $M\Omega \cdot m$,但当加词头 c 时,则要写成 $\Omega \cdot cm$;另一个是货运量单位“吨公里”写成 $t \cdot km$ 。

b. 通过单位相除或相乘除(分数形式)构成的组合单位,词头一般加在分子的第 1 个单位前,分母一般不加词头。例如:摩尔熵单位 $J/(mol \cdot K)$ 加词头 k,应写成 $k J/(mol \cdot K)$,不能写成 $J/(mmol \cdot K)$,虽然 $kJ/(mol \cdot K) = J/(mmol \cdot K)$,但 $J/(mmol \cdot K)$ 是不规则的。

- c. 一般不在组合单位的分子分母同时加词头。
- d. 组合单位的分母为长度、面积、体积单位或分子为 1 时,分母可按习惯与方便选用某些词头。
- e. 质量单位 kg 例外。