

SDRAM 在大视场 TDICCD 空间相机中的应用

李进^{*①②} 金龙旭^① 李国宁^① 张宇^①

^①(中国科学院长春光学精密机械与物理研究所 长春 130033)

^②(中国科学院研究生院 北京 100039)

摘要: 该文针对大视场空间相机中需要缓存的图像数据量大的问题,对高速大容量存储器SDRAM控制技术进行了研究。首先,说明了大视场空间相机中图像数据缓存的需要,根据CCD图像数据和图像缓存工作特点,提出了基于行使能和刷新操作驱动和基于仲裁策略的SDRAM控制器。然后,为了提高图像数据缓存可靠性,针对缓存小数据量场合,提出了(6,8)纠错算法,针对缓存大数据量场合提出了RS(143,127)和RS(142,126)码纠错算法。最后,在XX-X空间多光谱相机样机的成像系统上进行了试验验证。实验结果表明:两种控制器工作频率能够达到131 MHz,正常工作时,行使能驱动控制器存取速度达到127.08 MBps,仲裁策略控制器存取速度达到139.788 MBps,两种纠错算法在507 B/row内可以纠正32 b错误。基本满足空间相机中的稳定可靠、高效率的缓存图像数据要求。

关键词: 空间相机;行使能驱动控制器;仲裁控制器;(6,8)纠错算法;RS(143,127)码

中图分类号: TP391.4; TN919.8

文献标识码: A

文章编号: 1009-5896(2012)05-1246-06

DOI: 10.3724/SP.J.1146.2011.00906

Application of SDRAM to the Large Field of View TDICCD Space Camera

Li Jin^{①②} Jin Long-xu^① Li Guo-ning^① Zhang Yu^①

^①(Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China)

^②(Graduate University of Chinese Academy of Sciences, Beijing 100039, China)

Abstract: In order to resolve the problem of images buffering with huge amount of data quantum in large field TDICCD space camera, the control technology of SDRAM which is a high-speed and large capacity data memory is studied. First, the paper explains the image buffering demand in large field TDICCD space camera, and a SDRAM controller using line enabling driving and a SDRAM controller using arbitration policy are proposed based on the characteristics of CCD image data and that of image buffer. Then, In order to improve reliability for data buffering, a (6,8) error correction algorithm is proposed for the small data situation, and a error correction algorithm of RS (143,127) and RS (142,126) code are proposed for the large data situation. Finally, the verification experiments to CCD imaging system in the prototype machine of XX-X space multi-spectral camera are carried out. The experiments results show that the working frequenc of the two proposed controllers is able to reached 131 MHz. When the system works steadily, the data access speed of line driving controller is 127.08 MBps, and that of arbitration controller is 139.788 MBps. The two proposed error correction algorithms are able to correct 32 b errors in 507 B/row. It can satisfy the space camera data buffering requirements of non-contact, real time and rapid speed.

Key words: Space camera; Line driving controller; Arbitration controller; (6,8) error correction algorithm; RS (143, 127) code

1 引言

随着基于时间延迟积分(TDI)电荷耦合器件(CCD)成像技术的空间相机在视场和分辨率指标要求上的不断提高,导致 CCD 图像数据量急剧增加^[1]。为了对图像实时传输、压缩以及存储等处理

需要高速大容量的存储器作为图像的缓存。现有 SRAM, BRAM 等存储器无法满足要求,需要开发适应空间环境的高速大容量图像缓存器。同步动态 RAM(SDRAM)作为存储介质具有容量大、速度快、体积小、价格低等优点,使它成为各个领域电子系统数据缓存的首选存储介质。然而 SDRAM 的控制逻辑比较复杂,对时序要求也十分严格,同时还要考虑读写与刷新操作的冲突,需要设计控制器对其状态进行控制。

2011-09-02 收到, 2012-02-03 改回

国家 863 计划项目(863-2-5-1-13B)资助课题

*通信作者: 李进 664910699@qq.com

目前,国内外很多学者在不同的应用系统中对 SDRAM 控制器的设计进行了大量的研究,提出了很多 SDRM 控制策略^[2]。然而这些应用场合的 SDRAM 控制器仅能适应于自身的工作特点。对于大视场空间相机而言,CCD 输出图像数据格式不同于一般数据且各片 CCD 行频各不相同。另外,还要受单粒子翻转现象影响导致数据差错。通用 SDRAM 控制器难以满足空间相机的应用。因此,在大视场空间相机中,针对其工作特点需要进行设计专门 SDRAM 控制器。

本文在参考国内外相关技术的基础上,根据背景项目需求,提出了基于行使能驱动和基于仲裁策略的 SDRAM 控制器。重点研究了(6,8)纠错算法和 RS(143,127)+RS(142,126)码纠错算法,两种纠错算法突破了传统 SDRAM 控制器中纠错算法为零的现状,大大提高了数据缓存的可靠性。本文提出的创新性技术已经在 XX-X 空间多光谱相机样机的成像系统上获得了应用,应用表明,可以有效地解决大视场 TDICCD 空间相机中的图像缓存要求。

2 图像缓存系统总体结构

为了说明 SDRAM 在大视场遥感相机中的应用,以笔者设计的 XX-X 空间多光谱相机原理样机的处理 1 片 CCD 的图像缓存系统为例,系统总体结构如图 1 所示。在轨道高度为 500 km 下,CCD 像元转移速率为 5.5 MHz,高速低压差分信号(LVDS)发送器并行工作时钟为 55 MHz。因此需要 SDRAM 作为缓存。图像压缩时需要构造一帧图像,且压缩工作时钟为 50 MHz。因此也需要 SDRAM 作为缓存。压缩码流存储和下传时,闪存读写时钟为 30 MHz,压缩码流发送器并行工作时钟为 100 MHz,因此也需要 SDRAM 作为缓存。可见 SDRAM 能否正常高效率的工作成为整个相机能否正常工作

的瓶颈。

CCD 图像格式是以行为单位实时输出的。每行包括有效像素和消隐像素。有效像素由 3000 个图像数据和 42 个辅助数据组成。在轨工作时为了进行异速匹配消隐像素数需要调整,导致各片 CCD 行频各不相同,但同片 CCD 的 4 通道数据行频始终相同。图像捕获单元中需要缓存 4 通道一行数据后输入到高速 LVDS 发送单元中。为了保证实时性,考虑上述特点并结合 SDRAM 读写特点,本文提出一种基于行使能驱动和分次存储的 SDRAM 控制器。如图 1 中所示 SDRAM 图像行缓存控制器。控制器主要包括:行使能识别、内部 RAM, ECC 纠错编解码等模块。各通道行使能信号有效后各通道数据先以乒乓操作的方式存入内部 RAM 中,然后在刷新操作的驱动下分 6 次读取(每次读取 507 个数据)RAM 中的数据进行 ECC 编码后存入 SDRAM 中。为了提高数据读取的可靠性,考虑 SDRAM 缓存数据量相对小,但缓存输出时钟频率较高,本文采用(6,8)码纠错算法进行 ECC 编码。

图像压缩单元中,1 片压缩芯片负责处理 2 片 CCD 图像,每片压缩芯片需要 2 片 SDRAM 缓存 128 行图像。2 片 CCD 的 8 个通道行频不同,每行的有效像素数据是不同时到达的,本文采用仲裁策略 SDRAM 控制器。如图 1 中所示 SDRAM 图像帧缓存控制器。其中,中断仲裁单元包括通道中断表建立和仲裁单元。该单元主要完成负责检测不同通道的行使能信号,根据先后到的通道,把其序号存入中断表中。然后根据中断表采用分次存储和行激活仲裁处理策略将数据写入 SDRAM 中。本文根据该单元中 SDRAM 缓存数据量相对大,但缓存输出时钟频率较低,采用 RS 码纠错算法进行 ECC 编码。

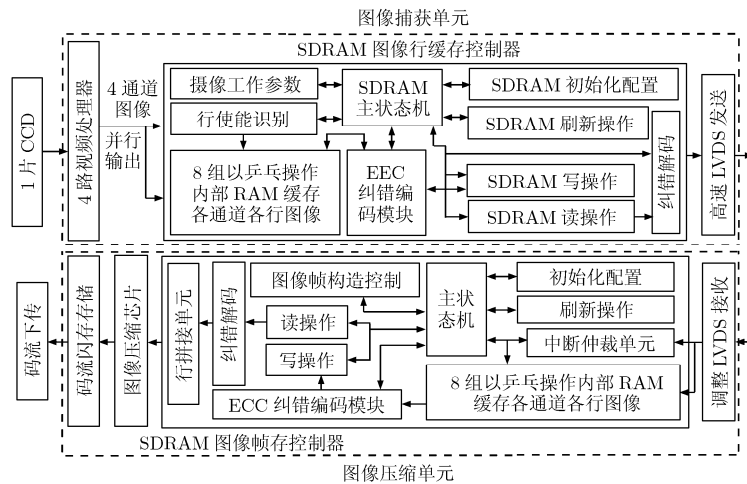


图 1 图像缓存系统

3 关键技术

3.1 (6,8)码纠错算法

空间环境的辐射可能导致存储器翻转,即单粒子翻转导致差错。SDRAM 缓存一行图像数据后某比特出现单粒子翻转现象时,如果没有纠错编码,将导致图像发生差错。为了提高数据单粒子翻转纠错能力,权衡纠错码算法计算速度,本文提出一种(6,8)码的 SDRAM 纠错算法。

(6,8)码纠错方法是采用小步长数据范围进行纠错编码策略。本文采用每 16 像素数据为一组进行纠错编码。12 bit 的 CCD 图像数据添“0”成为 16 bit,将 16 bit 分为高字节和低字节分别进行纠错编码。这样可以把 16 byte 数据看成 16×8 的矩阵,矩阵的每个元素表示一个比特位。矩阵进行编码后分别生成 6 bit 的列校验信息和 8 bit 的行校验信息,本文称为(6,8)纠错码。(6,8)码的生成规则如图 2 所示。

1st byte	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	E_1	$P8'$	$P16'$	$P32'$	$P64'$
2nd byte	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	E_2	$P8$			
3rd byte	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	E_3	$P8'$	$P16$		
4th byte	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	E_4	$P8$			
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		
12th byte	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	E_{13}	$P8'$	$P16'$	$P32$	$P64$
13th byte	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	E_{14}	$P8$			
14th byte	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	E_{15}	$P8'$	$P16$		
15th byte	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	E_{16}	$P8$			
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0					
	P_1	P_1'	P_1	P_1'	P_1	P_1'	P_1	P_1'					
	P_2	P_2'		P_2	P_2'		P_2	P_2'					
	P_4				P_4'								

图 2 (6,8)纠错校验码生成方法

图中 $P_1, P_1', P_2, P_2', P_4, P_4'$ 为列校验码, $P_8, P_8', P_{16}, P_{16}', P_{32}, P_{32}', P_{64}, P_{64}'$ 为行校验码。(6,8)编码算法思想为:设待输入编码器的第 n 字节 S 为

$$S(n) = (\alpha_n^7, \alpha_n^6, \alpha_n^5, \alpha_n^4, \alpha_n^3, \alpha_n^2, \alpha_n^1, \alpha_n^0) \quad (1)$$

列校验码 P 和 P' 的生成计算方法为

$$D(m) = \sum_{n=1}^{16} \alpha_n^m, \quad m = 0, 1, 2, \dots, 7 \quad (2)$$

$$P_1 = D(7) + D(5) + D(3) + D(1) \quad (3)$$

$$P_1' = D(6) + D(4) + D(2) + D(0) \quad (4)$$

$$P_2' = D(5) + D(4) + D(1) + D(0) \quad (5)$$

$$P_2 = D(7) + D(6) + D(3) + D(2) \quad (6)$$

$$P_4 = D(7) + D(6) + D(5) + D(4) \quad (7)$$

$$P_4' = D(3) + D(2) + D(1) + D(0) \quad (8)$$

行校验码 P 和 P' 的生成计算方法为

$$E(n) = \sum_{i=0}^7 \alpha_n^i, \quad n = 1, 2, 3, \dots, 16 \quad (9)$$

$$P_8 = \sum_{k=1}^8 E(2k) \quad (10)$$

$$P_8' = \sum_{k=1}^8 E(2k-1) \quad (11)$$

⋮

$$P_{64} = \sum_{k=9}^{16} E(k) \quad (12)$$

$$P_{64}' = \sum_{k=1}^8 E(k) \quad (13)$$

其中“+”表示“位异或”操作。

下面说明纠错算法如何检测错误并进行纠正。每向 SDRAM 写入 16 个像素数据(以低字节为例)时,生成 14 bit 校验码,写入 507 个像素数据则生成 56 byte 的校验码,称为写校验码。保存到 Block RAM 中。当从 SDRAM 读取数据时,每读取 16 个像素数据对其进行(6,8)编码,生成 14 bit 校验码,称为读校验码。将从 Block RAM 中读出的 14 bit 写校验码与读校验码按位异或操作。结果会出现 3 种情况:(1)全为“0”,则数据不存在错误;(2)存在 7 bit 为“1”,则存在一个比特错误,且可以纠正;(3)只存在 1 bit 为 1,则生成的校验码出错,数据本身无错;(4)其他情况则出现多个错误,不可纠正。通常空间环境会使 SDRAM 的某比特发生翻转,而校验码没有保存其中,因此通常会出现(1)和(2)两种情况。当出现情况(2)时,由于发生单粒子翻转的数据位在数据组中的字节偏移量为 $P_{64}, P_{32}, P_{16}, P_8$ 。发生单粒子翻转的数据位在所处的字节中的位偏移量为 P_4, P_2, P_1 。因此,根据字节的偏移量和字节中的位偏移量就可以确定数据中发生的翻转位,然后对其进行纠正。

3.2 基于 RS 码的纠错算法

本文提出的(6,8)纠错算法对于缓存小数据场合时,计算速度快、实现简单、纠错能力强。对于缓存大数据场合时(6,8)纠错算法生成的校验码占用内存较大,如缓存 3000×128 的图像时生成 84 kB 的校验码。本文针对缓存大数据场合提出一种基于 RS(143,127)和 RS(142,126)码的 SDRAM 纠错算法。

通常 RS 码^[3]定义为:在有限域 $GF(q)(q \neq 2)$ 上,码长 $n=q-1$ 的本原 BCH 码。其码元符号与生成多项式的根都是 $GF(q)$ 上的元素。具有以下特性:

$$n = 2^m - 1, \quad n - k = 2t, \quad d_{\min} = 2t + 1 \quad (14)$$

式中 m 为符号比特数; n 为 RS 码长; k 为信息长度; t 为可纠错码元数; d_{\min} 最小码距离。此 RS 码是 1 个可在 n 码字上纠正 t 码字或更少错误的 RS 码。根据 SDRAM(以 MT48LCM16A2 为例)特点及缓存图像大小要求,将 SDRAM 每列缓存的数据分

为4块,每块的高低字节分别独立编码,前3块每块为127 B,剩余126 B作为1块。为了使纠错能力与(6,8)纠错算法相同。所以设计 $t_1=8$,待编码元 $k_1=127$,总长 $n_1=143$ 的RS(143,127)码,以及 $t_2=8$,待编码元 $k_2=126$,总长 $n_2=142$ 的RS(142,126)码。由于 $2^7-1 < n_1, n_2 < 2^8-1$,所以,RS(143,127)和RS(142,126)其实是有限域GF(2^8)上标准码RS(255,239)的缩短码。RS(143,127)和RS(142,126)分别是删去(255,239)码的高112 bit和113 bit信息符号(全取0)。本文采用添0和去0的办法使其变为标准形式的RS码。这时,在设计编码器时,当输入126 B和127 B数据时,编码器会自动分别补113 B和112 B的0,将长度转为239,然后生成16 B的校验码元和255 B的RS(255,239)码。当编码结果输出时,在将所加入的0去掉,变成RS(143,127)和RS(142,126)码输出,然后存入SDRAM中。译码时,把原始数据和校验码一并读出组合输入译码器,译码器也会自动填补0,译码输出结果后,去掉填补的0即可。

RS(255,239)编码算法思想为:根据有限域性质^[4]求得GF(2^8)上本原多项式 $F(x)$,由 $F(x)$ 求出GF(2^8)域的全部元素编码表。由编码表得出生成多项式为

$$g(x) = x^{16} + 118x^{15} + \dots + 44x + 79 \quad (15)$$

设待输入编码器的码字多项式为 $D(x)$ (共 k 位),产生的校验元为 $R(x)$ (共 $n-k=16$),则编码后码多项式 $C(x)$ 可表示为 $C(x) = x^{n-k}D(x) + R(x)$,求出 $R(x)$ 便完成了编码。由于码多项式 $C(x)$ 是 $g(x)$ 的倍式,易得 $R(x) = x^{n-k}D(x) \bmod g(x)$ 。

3.3 SDRAM 纠错译码算法

RS译码算法的整体结构图参考文献[5],这里不再赘述。RS(255,239)译码算法思想如下:

(1)由输入译码器的 $R(x)$ 求伴随式 $S(x)$ 。设 $R(x) = \sum_{i=1}^{255} c_{255-i}x^{255-i}$,伴随矩阵 $S = RH^T$,其中, H 为校验矩阵。得到伴随式系数

$$s_k = \sum_{i=0}^{254} c_i \alpha^{ik}, k = 1, 2, \dots, 6 \quad (16)$$

若采用式(16)计算任意 s_k 要进行254次加法和 $255 \times (255-1)/2 = 32385$ 次乘法,计算量较大。本文采用一种改进办法即将式(16)转换为

$$s_k = \{\dots[(c_{n-1}\alpha^k + c_{n-2})\alpha^k + c_{n-3}]\alpha^k + \dots\}\alpha^k + c_0 \quad (17)$$

则乘法运算减为254次。

(2)由伴随式求错误位置多项式 $\sigma(x)$ 和错误值多项式 $\omega(x)$ 。目前比较成熟的方法有:BM算法^[6],Euclid算法^[7]和PGZ算法^[8]。PGZ算法无快速递归

法,不易于硬件实现。Euclid算法需要保存大量的中间量,占用大量存储空间。BM算法是迭代算法,极大地加快了求 $\sigma(x)$ 的速度,且易于用FPGA实现。权衡考虑速度和资源问题,本文采用BM算法并对其进行改进。

在BM算法中要用到有限域元素求逆运算,会大大增加逻辑资源用量,并使系统速度受限。所以对该算法进行改进。具体改进如下:

初始化: $\delta^{(0)}(x) = 1, \omega^{(0)} = 1, l^{(0)} = 0, \gamma^{(j)} = 1$ 且 $j > 0$ 开始迭代;

循环迭代:定义

$$\delta^{(j+1)}(x) = \gamma^{(j)}\delta^{(j)}(x) - d^{(j+1)}\omega^{(j)}(x) \quad (18)$$

其中 $d^{(j+1)}(x) = \sum_{i=0}^l \delta_i^j S_{m+j-1}^o$

$$\omega^{(j+1)}(x) = \begin{cases} x\omega^{(j)}(x), & d^{(j+1)} \neq 0, 2l^{(j)} > j \\ \delta^{(j)}(x), & d^{(j+1)} \neq 0, 2l^{(j)} \leq j \end{cases} \quad (19)$$

$$l^{(j+1)}(x) = \begin{cases} l^{(j)}(x), & d^{(j+1)} \neq 0, 2l^{(j)} > j \\ x + 1 - l^{(j)}, & d^{(j+1)} \neq 0, 2l^{(j)} \leq j \end{cases} \quad (20)$$

$$\gamma^{(j+1)}(x) = \begin{cases} \gamma^{(j)}, & d^{(j+1)} \neq 0, 2l^{(j)} > j \\ d^{(j+1)}, & d^{(j+1)} \neq 0, 2l^{(j)} \leq j \end{cases} \quad (21)$$

以上循环迭代经过 $2t(t=3)$ 次迭代后得到 $\sigma^{(2t)}(x)$ 和 $\omega^{(2t)}(x)$,即为所求的 $\sigma(x)$ 和 $\omega(x)$ 。改进的BM算法中没有用到有限域求逆运算,这样就能简洁快速地完成求解运算,能减少大量的时钟周期。

(3)通过计算 $\sigma(x)$ 在GF(2^8)中所有元素的值从而找出错误位置多项式的根,即依次将 $\alpha^i(0 \leq i \leq 254)$ 代入 $\sigma(x)$ 。若 $\sigma(\alpha^i) = 0$,则第 i 位出现误码, α^{255-i} 为错误位置。将错误位置 α^{255-i} 和错误估值多项式 $\omega(x)$ 代入Forney^[8]公式求出错误值 e_i 。算出错误位置和错误值之后,错误值与对应错误位置的码元异或便可得到纠错之后的码元。

4 测试实验与结果

4.1 实验方案

为了测试SDRAM控制器工作性能使用了自主研发的地面检测设备,实验系统如图3所示。服务器向图像模拟源注入各种测试图像,图像模拟源对图像进行调整后发送至图像缓存系统。图像缓存系统为图1所示的系统,图像缓存系统对测试图像进行缓存后发送至地面相机检测设备中,地面相机检测设备进行观察图像,并进行缓存性能分析。另外,为了测试和仿真SDRAM控制器中相应模块性能,本文采用Verilog HDL设计整个SDRAM控制器和SDRAM控制器的寄存器传输级(RTL)模型。仿真

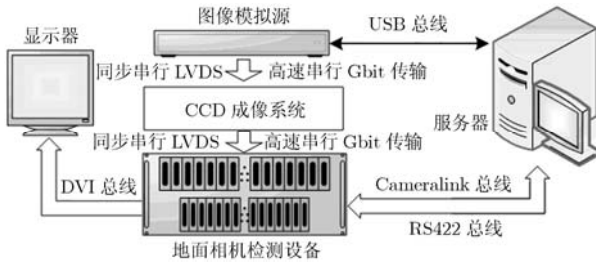


图3 实验系统结构

试验在 Xilinx ISE 8.2i 软件平台上并调用 ModelSim SE 6.1f 完成。

4.2 SDRAM 缓存实验验证

为了验证 SDRAM 控制器整体工作性能, 图像模拟源向图像缓存系统发送自校测试图形, 并加入单粒子翻转事件, 测试自校图形经编码、SDRAM 缓存, 然后读取 SDRAM 图像、纠错译码后通过 Camera Link 传到 PC 机上, 得到如图 4(c)和 4(d)所示, 测试自校图形不加入和加入单粒子翻转事件, 不经过缓存直接传到 PC 机上得到如图 4(a)和 4(b)所示。

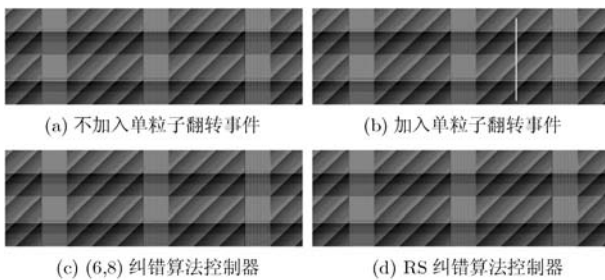


图4 系统实验结果

取 20 组如图 4 所示的图像, 并对各组图像的数码值进行比较。计算方法借鉴数学上衡量保真度的均方根误差公式:

$$e_{\text{rms}} = \sqrt{\sum_{x=1}^M \sum_{y=1}^N (p(x,y) - q(x,y))^2 / (MN)} \quad (22)$$

其中 $p(x,y)$ 为图 4(a) 数码值, $q(x,y)$ 为图 4(c) 和 4(d) 数码值, M 为 CCD 像元数, N 为扫描行数。由式 (22) 算得 20 组图像的均方根误差 e_{rms} 均等于 0, 可见 SDRAM 控制器可以稳定可靠的工作。

本文在 SDRAM 控制器中引入了提出的两种纠错算法, 与目前现有 SDRAM 控制器纠错能力比较如表 1 所示。可见。本文提出的两种纠错算法可以大幅度地提高 SDRAM 环存储数据的可靠性。

本文提出的两种控制器在 Xilinx 生产的 Virtex-2 Pro 平台的 XC2VP40 下进行综合得到: (1)(6,8) 算法控制器 LUT 占用率为: 5122/38784(13%), BRAM 占用率为 27/192(14%); (2)RS 码算法控制

表1 纠错能力比较

范围	文献[9]	文献[10]	本文方法
512 B/row	0 b	0 b	32 b

器 LUT 占用率为: 8116/38784(20%), BRAM 占用率为 46/192(23%)。另外, 笔者针对本文提出的两种控制器进行了实际的读写强度测试, 得出在 SDRAM 最大工作频率为 133 MHz 下, 控制器工作频率能够达到 131 MHz。在轨道平均高度下, 正常工作时, 设行频为 f , 处理通道数为 n , 每通道像元数为 p , A/D 转换位数为 k , 则 SDRAM 控制器存储数据吞吐率 s 为

$$s = nkpf / 8 \quad (23)$$

其中 $k=12$, $p=3000$, $f=7.06$ kHz, 对于行使能驱动控制器 $n=4$, 求得 $s=127.08$ MBps, 对于仲裁 SDRAM 控制器由于受压缩芯片输入速率限制, 只需缓存 2 通道的数据, 即 $n=2$, 求得 $s=63.54$ MBps。当不受限制时该控制器平均可以处理 4.4 通道数据, 即数据吞吐率可达到 139.788 MBps。与其他 SDRAM 控制器实际最大工作频率和数据吞吐率比较如表 2 所示。

表2 SDRAM 控制器工作性能比较

方法	实际最大工作频率 (MHz)	正常工作数据吞吐率 (MBps)
文献[11]	120	30
文献[2]	108	62.208
本文行驱动法	131	127.08
本文仲裁法	131	63.54

5 结论

本文提出的基于行使能驱动和基于仲裁策略的 SDRAM 控制器, 提高了单行 CCD 图像缓存操作效率, 解决了由于行频可变性导致图像缓存复杂的问题。提出的(6,8)纠错算法和 RS 码纠错算法突破了传统 SDRAM 控制器中纠错算法为零的现状, 不仅计算速度快, 而且不影响 SDRAM 操作速度, 提高了 SDRAM 缓存数据的可靠性。结果表明, 两种控制器工作频率能够达 131 MHz, 正常工作时, 行使能驱动控制器存取速度达到 127.08 MBps, 仲裁策略控制器存取速度达到 139.788 MBps, 两种纠错算法在 507 B/row 内可以纠正 32 b 错误。基本满足空间相机中的稳定可靠、高效率的缓存图像要求。为空间相机的图像缓存提供了一种很好的解决方案。

参考文献

- [1] 王磊, 吴家骥, 白静. 基于整型可逆时域交叠变换的遥感图像压缩[J]. 光子学报, 2010, 39(12): 2251-2256.
Wang Lei, Wu Jia-ji, and Bai Jing. Remote sensing image compression based on reversible integer time domain lapped transform[J]. *Acta Photonica Sinica*, 2010, 39(12): 2251-2256.
- [2] 赵强, 罗嵘, 王蕙, 等. 用于HDTV视频解码器的高性能SDRAM控制器[J]. 电子与信息学报, 2007, 29(6): 1332-1337.
Zhao Qiang, Luo Rong, Wang Hui, et al. High performance SDRAM controller design for HDTV video decoder[J]. *Journal of Electronics & Information Technology*, 2007, 29(6): 1332-1337.
- [3] Xiao Y, Luo C, and Yang C. The comparative analysis of LDPC and RS code[C]. IEEE Conference on Consumer Electronics, Communications and Networks(CECNET), Xianning, China, April 2011: 4510-4513.
- [4] 于湘珍, 刘军, 王秋娟. RS 码在汉信码中的纠错性能分析及仿真[J]. 南京理工大学学报(自然科学版), 2011, 35(1): 42-46.
Yu Xiang-zhen, Liu Jun, and Wang Qiu-juan. Analysis and simulation on error correcting performance of RS code in chinese sensible code[J]. *Journal of Nanjing University of Science and Technology (Natural Science)*, 2011, 35(1): 42-46.
- [5] Lee H. High speed VLSI architecture for Reed Solomon decoder[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) System*, 2003, 11(2): 288-294.
- [6] Berlekamp E R. Algebraic Coding Theory [M]. New York: McGraw Hill, 1968: 118-125.
- [7] Baek J H and Sunwoo M H. New degree computation-less modified Euclid algorithm and architecture for Reed Solomon decoder [J]. *IEEE Transactions on Very Large Scale Integration (VLSI) System*, 2006, 14(8): 915-920.
- [8] 刘悦, 刘明业, 尚振宏. RS(255,223)码的编译码软件实现[J]. 计算机应用与软件, 2006, 23(11): 47-49.
Liu Yue, Liu Ming-ye, and Shang Zhen-hong. A software implementation of encoding and decoding for RS (255,223) [J]. *Computer Applications and Software*, 2006, 23(11): 47-49.
- [9] 马灵, 杨俊峰, 宋克柱, 等. 地震数据采集集中基于FPGA的多DDR SDRAM控制器设计[J]. 中国科学技术大学学报, 2010, 40(9): 939-945.
Ma Ling, Yang Jun-feng, Song Ke-zhu, et al. A design of multiple DDR SDRAM controllers based on FPGA in a seismic data acquisition system[J]. *Journal of University of Science and Technology of China*, 2010, 40(9): 939-945.
- [10] 司孝平. 基于SDRAM的FPGA实时图像采集系统的设计[J]. 西南大学学报, 2011, 33(1): 128-132.
Si Xiao-ping. Design of an SDRAM-based FPGA real-time image acquisition system[J]. *Journal of Southwest University (Natural Science Edition)*, 2011, 33(1): 128-132.
- [11] 刘正, 黄战华, 刘书桂. 基于 SDRAM 的高分辨力高速图像缓存[J]. 光学仪器, 2006, 28(6): 49-53.
Liu Zheng, Huang Zhan-hua, and Liu Shu-gui. High-resolution high-speed image buffering using SDRAM[J]. *Optical Instruments*, 2006, 28(6): 49-53.
- 李进: 男, 1985年生, 博士生, 研究方向为光电成像和图像压缩存储技术.
- 金龙旭: 男, 1965年生, 研究员, 博士生导师, 研究方向为空间相机智能控制技术.
- 李国宁: 男, 1979年生, 副研究员, 研究方向为星上光电成像与图像处理.