基于 ARM和 CPLD的嵌入式数字图像处理系统

xxx[1] xxx[2]
(xxx大学 电气工程学院，xxx xxx xxx)

摘要：介绍一种嵌入式数字图象处理平台的实现方案，通过 ARM和 CPLD技术，构造一个具有通用性、可扩充性、灵活的数字图像处理平台作为嵌入式机器人控制系统的一个子系统。

关键词：ARM；PLD；数字图像处理
中图法分类号：TP391.41 文献标识码：A

ARM and CPLD -Based embedded digital image processing system

xxx[1], xxx[2]

Abstract: The purpose of this paper was to design and implement a scheme for embedded digital image processing .Depend on ARM and the CPLD technology, construct a versatile, scalable and flexible platform for digital image processing, as a part of embedded robot control system.

Key words: ARM; PLD; Digital image processing

0 引言

得益于数字信号处理技术，半导体光电技术和微处理器快速发展，作为人类主要信息来源之一的视觉系统，正越来越多的受到重视。越来越多的领域需要一种具备大数据量处理能力的快速和实时系统；同时对系统的体积、功耗、稳定性等也有较严格的要求。本文介绍一种嵌入式数字图象处理平台的实现方案，通过 ARM和 CPLD技术，构造一个数字图像处理平台作为嵌入式机器人控制系统的一个子系统。

1. 系统需求及总体方案设计

1.1. 系统需求分析

图像采集和分析方面，考虑到作为移动机器人系统应用，因为机器人移动速率比较低的情况下，每秒 10到 15帧的图像刷新速率基本可以满足系统要求。同时，对图像的分辨率的要求可以在 640 480以内的范围内可调，一般情况下选择 640 480和 320 240两种主要的分辨率模式。

作为移动机器人系统，一种无线通信机制应该是必备的组成部分。作为子系统的图像系统来说，一定的无线通信能力也可以满足独立构成控制器的需要。以一种高速通信协议为基础实现图像处理结果上传和指令下载成为了系统需求的一部分。

最后，考虑到图像系统有可能单独构成控制器，对其控制能力就提出来更多的要求，比如提供一定数量的 GPIO和多种外设功能。

1.2. 总体方案实现

从总体来说，目前在嵌入式图像处理方面，主要有两类思路：

第一类，采用基于 DSP技术的数字图像处理应用。这类系统的突出优势在于可以在完成复杂数据处理算法的同时，保证图像的实时特性。相关的控制能力相对较弱。

第二类，以 MCU为基础构建系统，相对 DSP的强大运算能力来说，MCU的处理复杂运算的能力明显不足。同时，因为这类 MCU如果没有 DMA能力而通过中断技术进行数据的采集，很多情况下会因为中断响应情况而导致数据的丢失。[1] [2]

基于以上的分析，我们采用一种 ARM + CPLD + RAM结合的结构，在提供一定的控制能力的基础上，保证了数据采集的快速和完整性。如图 1所示 .该结构主要有以下特点：

第一， 以 ARM为主控制器，利用这类 32位的微控制器完成对图像数据的简单处理，

判断并作出决策，同时还具备构成一个较强的独立控制系统能力。

第二， 为了兼顾采样速率，采用 CPLD+RAM的结构，构造了一个 类 DMA 方式的控制器。构成数据缓存的方式还有其他多种，比如采用双口 RAM, FIFO等方式，但大多限制于容量和价格因素而未予采用。

第三， 在图像数据来源方面，主要有两类方案，其一，是采用模拟信号接入并进行编码，将标准的全电视信号转换为数字信号以及对应的同步信号。另外一类就是采用 COMS或 CCD技术的数字摄像头。在本系统中，因为不需要扩展到对模拟信号的处理，故采用数字摄像头方案作为图像数据来源。[3]

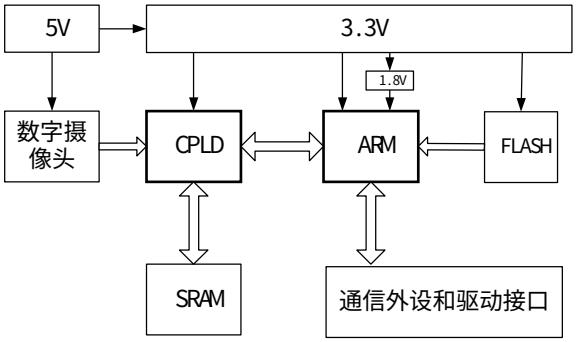

图 1: 系统结构图

在图像传输通道以外，系统还包括的其他扩展外设。2MByte的 FLASH存储器用于保存用户程序，扩展 USB接口是预留 USB网卡的接入通道，SPI接口则是为 2.4GHz无线数据传输模块预留通道，以实现小量数据的高速传输。为了便于调试而扩展 UART接口，CAN接口是为了和机器人主控制器各模块交换少量数据和命令而设计的。另外，为了提供独立控制能力扩展的少量通用 IO端口可以为以后直接控制功率驱动模块而准备。

2. 硬件设计

在已经形成的总体方案下，对各部分进行详细权衡和设计以形成具体的原理设计。下面就本系统与图像数据流相关的硬件部分的设计进行比较详细说明。

2.1. 数字图像数据格式说明

首先对数字图像数据格式情况做一个简单说明。如图 2所示，来自 OV7620的图像数据，分为 Y0～Y7和 UV0～UV7共 16位数据宽度，另外同时输出的还包括四个同步信号：像素同步信号（PCLK），垂直场同步信号（VSYNC），水平行同步（HREF）和奇偶场同步信号（FODD）。凭借以上各同步信号的协调动作，就可以为每一个像素点的 16位灰度数据提供精确的定位。下面以 320 240分辨率的图像为例说明各同步信号和数据的关系，VSYNC输出表示垂直场 扫描 的开始，FODD输出为高表示当前输出数据为奇场数据，HREF输出高表示对一行像素的 扫描 ，在一行中，每一个像素点的输出，伴随着 PCLK的一次脉冲过程。[4]

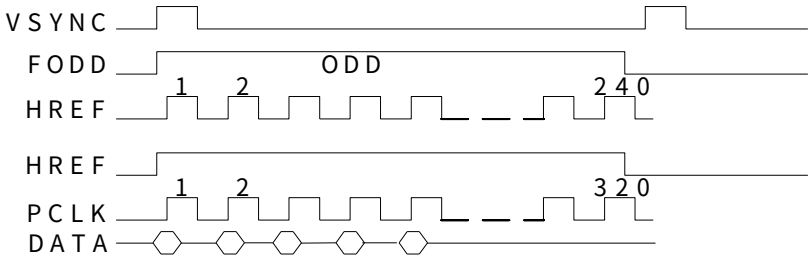

图 2: 同步信号时序

根据以上的数据传输过程，构建了以下图像采集电路，如图 3所示。

首先，在系统完成初始化过程之后，由作为主控制器的 ARM发出指令，控制 CPLD启动数字摄像头，开始接受图像。来自 OV7620的同步信号通过 CPLD，以 VSYNC,FODD和 HREF作为使能信号，对 PCLK进行计数以产生写入到 SRAM的地址，在相关的写控制信号作用下，将同时获得的数据写入到 SRAM中，直到完整的一场或一帧数据被全部写入。这个过程 ARM可以完成其他任务而不需要任何参与，在图像完成以后，由 CPLD停止图像采集并向 ARM提出中断申请。

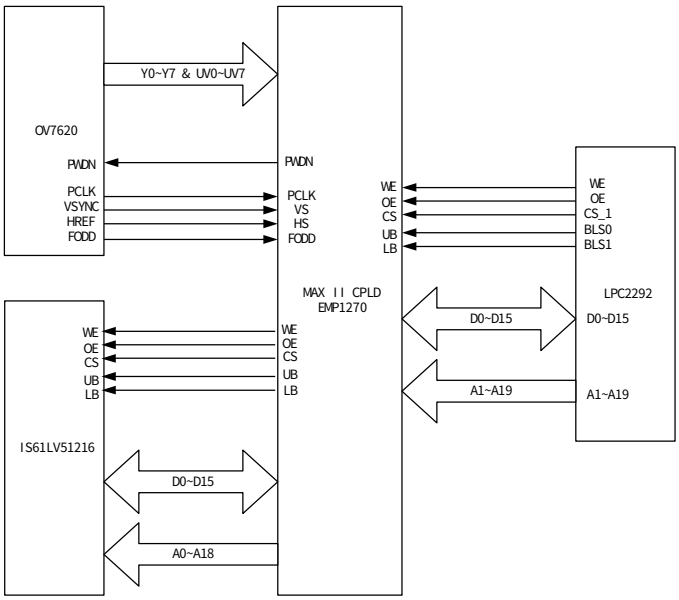

图 3：图像采集系统原理图

在图像写入完成后，在中断响应中 ARM声明对 SRAM享有完全的访问权。ARM按照常规方式对 SRAM进行访问，对数据完成处理。处理过程完成之后，ARM从新交出 SRAM的访问权给 CPLD，并启动新一轮的图像采集和写入过程。与此同时，ARM就可以将分析处理的结果通过通讯外设发送到相关系统。

3. 软件设计

软件部分涉及到三个方面，ARM部分的应用程序，基于 VHDL的 CPLD的硬件功能描述设计以及在 PC端的 C#的上位机程序设计 [5]。下面，主要就前两个部分分别说明。

3.1 CPLD的 VHDL设计

在本系统中，CPLD主要完成总线切换的功能，类似于实现一种简易的 DMA 控制器一样，主要思路就是由一个统一的信号控制，对两侧的三套总线（地址，数据和控制）进行切换。设计过程中，采用混合设计方式，采用 VHDL方式分别设计三套总线切换模块和地址发生模块，最后在通过在顶层的原理图中调用所设计的相关模块构建该控制器，如左图 4所示。

其中，以数据切换模块为例，对 VHDL设计进行说明。[6][7]

```
entity data_switch is
port( globe_en : in std_logic ; --全局控制信号
      dir      : in std_logic ; --方向控制，决定是由 CM写 RM还是 ARM访问 RM
      am_oe    : in    std_logic; --ARM控制的读信号
      am_we    : in    std_logic; --ARM控制的写信号
      am_d     : inout std_logic_vector(15 downto 0); --来自 ARM的数据总线
      cm_d     : inout std_logic_vector(15 downto 0); --来自摄像头的数据总线
      rm_d     : inout std_logic_vector(15 downto 0) --来自 SRAM的数据总线
     );
```

```
end ;
architecture one of data_switch is
begin
PROCESS(globe_en,dir,am_oe,am_we)
  BEGIN
  IF globe_en = '1' THEN
    IF dir ='1' THEN              --arm acccess ram
      IF am_oe = '0' THEN       -- arm read data from ram
        am_d <= rm_d;  --如果全局使能有效，方向控制为 ARM-RAM，且为 读
      ELSE                --则可以将 RAM的数据总线连接到 ARM数据总线
        am_d <= "ZZZZZZZZZZZZZZZZ" ;
      END IF;
      IF am_we='0' THEN       --如果全局使能有效，方向控制为 ARM-RAM，且为 写
        rm_d <= am_d;  --则可以将 ARM的数据总线写到 RAM数据总线
      ELSE
        rm_d <= "ZZZZZZZZZZZZZZZZ" ;
      END IF;
    ELSE                --如果全局使能有效，方向控制为 CM-RAM ，直接写数据到 RAM
      rm_d <= cm_d ;
    END IF;
  ELSE
    rm_d <= "ZZZZZZZZZZZZZZZZ" ;
  END IF ;
  END PROCESS ;
END ONE ;
```

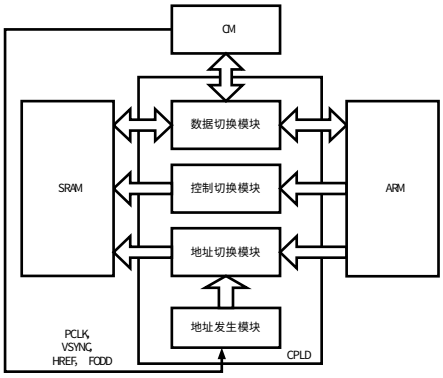

图 4：CPLD原理图设计

其他两个切换模块也是类似的处理方法。另外，对地址发生模块，直接采用在有效使能信号下，直接对输入的 PCLK进行循环计数，产生 19位地址。

3.2 ARM应用程序设计

在 ARM平台，采用 C语言作为主要的开发设计方式。主要的工作流程在硬件设计部分以初步介绍，下面以 320 240图像为例，简单说明数据访问过程的代码框架设计。

将 SRAM中的一维线性数据转换为二维图像，就成为了空间域图像处理的框架，采用 外层循环+内层循环 的框架，可以通过坐标方式访问图像中每个像素点数据。外层循环完成 行寻址 ，内层循环完成 行内寻址 。因为是 16位寻址方式，故在内层循环中，来自 ARM

的地址应该是 16位对齐，也就是 +2 的方式访问。[8][9]

```
for (y_coordinate = 0; y_coordinate < 240; y_coordinate++){
  for (x_coordinate = 0; x_coordinate < 642; x_coordinate+=2){
    add_bank_0 = RAM_BANK_0_BASE+(y_coordinate*642+x_coordinate) ;
    temp16 = *((unsigned short *)add_bank_0) ;
    //用户算法代码，读取的像素灰度进行处理之后，在回写 SRAM
    *((unsigned short *)add_bank_1) = temp16 ;
  }
}
```

4. 总结

经过试验，证明本方案基本可以实现系统需求，具备一定的图像处理分析能力和较强的控制能力。可以满足较低速率图像处理应用要求。图 5是上位机获取的原始图像数据和经过ARM处理的图像，通过串口上传辅助调试。图 6是完成的硬件平台

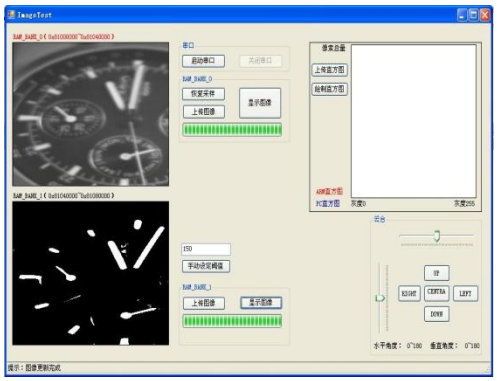

图 5: 原始图像与处理图像

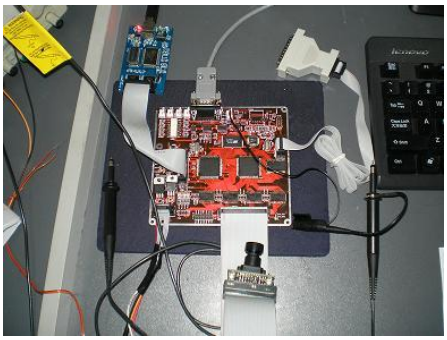

图 6: 系统硬件平台

5 主要参考文献：

[1] 任贵伟，张海．基于 ARM的紧凑型图像采集系统：[J]．单片机与嵌入式系统应用，2007，（02）：35-37.

[2] Anthony Rowe, Adam Goode and Carnegie Mellon University.CMUcam3 Datasheet, [EB/OL]．[2007-04-2]．http:// www.seattlerobotics.com

[3]庄婵飞，盛仲英．基于 OV7620的视频信息采集系统的设计与实现：[J]．浙江工业大学学报，2006，34（4）：249-252.

[4] OmniVision公司．Serial Camera Control Bus Functional Specification [EB/OL]．[2007-02-26]．http://www.ovt.com.

[5]李兰友，王学彬．Visual C#图像处理程序设计实例：[M]．北京：国防工业出版社，2003.

[6]吴继华，王诚．Altera FPGA/CPLD设计：[M]．北京：人民邮电出版社，2005.

[7] Altera公司．MAX II Device Handbook [EB/OL]．[2007-06-26]．http://www.altera.com/literature/hb/max2/max2_mii5v1.pdf

[8]杜春雷．ARM体系结构与编程：[M]．北京：清华大学出版社，2003.

[9]周立功．ARM嵌入式系统基础教程 [M]．北京：北京航空航天大学出版社，2004.

[10]陈书海，傅录祥．实用数字图像处理 [M]．北京：科学出版社，2005.