

基于 AVR 单片机的 DSP 算法

胡启龙, 冯明发, 夏继媛

(深圳职业技术学院电子工程系, 深圳 518055)

摘 要: 针对当前微处理器的高速处理性能, 提出把微处理器作为基本的运算与控制单元, 代替专用的数字信号处理芯片来执行控制动作。并以 AVR 单片机 ATMEGA16L 芯片为例, 探讨设计过程中的程序优化策略及其他注意事项, 为复杂控制算法在通信领域中的应用开辟一条新的途径。

关键词: 微处理器; AVR 单片机; 数字信号处理

DSP Algorithm Based on AVR Single-chip Processor

HU Qi-long, FENG Ming-fa, XIA Ji-yuan

(Department of Electronic Engineering, Shenzhen Polytechnic, Shenzhen 518055)

[Abstract] This article focuses on Microprocessor(MPU) as basic control and algorithm unit to replace special Digital Signal Processing(DSP) chip based on high-speed processing ability and lower cost of current MPU. And a kind of AVR, ATMEGA16L is illustrated to discuss optimized strategy of software and other attentive items during design procedure. It creates a new approach for complex control algorithm in application of communication field.

[Key words] Microprocessor(MPU); AVR single-chip processor; Digital Signal Processing(DSP)

1 概述

随着信息技术革命深入和现代电子技术的飞速发展, 出现了运算能力强大的数字信号处理(Digital Signal Processing, DSP)芯片和集成度高、速度快、功耗低、可在线编程的微处理器, 使得设计一种体积小、功耗低的数据通信系统成为可能。

DSP 芯片, 也称数字信号处理器, 是一种特别适合于进行数字信号处理运算的微处理器, 其改进的哈佛(Harvard)结构、先进的多地址/数据总线和多级流水线机制、专用的硬件乘法器、高效的指令集使芯片指令周期降到 10 ns 以下。先进的制造工艺使其功耗仅有数十毫瓦, 体积通常在 25 mm × 25 mm 左右^[1-2]。和通用微处理器相比, DSP 芯片成本高, 控制方法复杂, 在某些实时性要求适中、算法多、运算量大、控制精度要求较高的场合, 例如安全应用、灯光照明、电源控制、小型家电等便携式的应用领域, 可用某些通用微处理器代替 DSP 芯片作为基本的运算与控制单元, 提高产品的性价比, 简化设计方法。

通用微处理器被应用在电子设备中已有 30 多年的历史, 随着技术与应用市场的演进, 芯片厂商纷纷在技术上寻求新的提升, 致力于开发基于高性能、低功耗的微处理器, 并将模拟数字转换器(ADC)、数字模拟转换器(DAC)、比较器(Comparator)、放大器(Amplifier)及各种通用接口等集成到微处理器芯片中, 在简化设计电路的同时, 极大地降低了系统成本, 提高了系统的整体性能。目前在微处理器平台上编程多使用汇编语言与 C 语言。C 语言是一种较为高级的语言, 开发周期短, 可读性、可移植性等方面优于汇编指令, 但在实时性要求较高、算法多、运算量大、控制精度要求较高的场合, 用 C 语言编程无法充分发挥微处理器器件的性能优势。而汇编语言简洁高效, 能够直接操作微处理器的内部

寄存器、存储空间、外设。同样功能的 C 代码, 效率往往只有直接书写的汇编程序的几分之一甚至几十分之一, 因此在微处理器平台上实现 DSP 算法, 必须要根据微处理器的特性对 C 语言编写的程序进行进一步的优化。以 ATMEL 公司的 AVR 单片机为例, 通过采用 Harvard 结构, 能够在单一的时钟周期内执行功能强大的指令, 每兆赫兹可以实现 1 Mips 的处理能力。AVR 内核为 32 个通用工作寄存器与丰富指令集的组合, 这种组合机构具备的代码率比完成同样处理能力的常规 CISC 微处理器要快 10 倍^[3-4]。它是一种可满足许多要求, 具有高度灵活性和良好性价比的嵌入式微处理器。

2 程序优化策略

由于 AVR 单片机并不是专门针对数字信号处理技术而设计, 一般不具备强大的定点数据和浮点数据处理能力, 因此在采用 C 语言编程实现 DSP 算法时存在指令效率低、执行速度慢等问题。针对这些问题, 可通过采用汇编语言编程的方法来解决。在用汇编语言编程时, 可采用如下几种策略来优化代码, 提高指令效率, 达到充分发挥 AVR 单片机器件性能优势的作用。

2.1 减少函数调用, 以空间换时间

当程序进行函数的调用和返回时, 要在进栈出栈操作上消耗一定的运算量。如果函数较为短小, 则这些操作消耗的运算量就要占去很大比例。因此, 要尽量减少函数调用, 以牺牲代码空间来达到提高指令执行效率的目的。

2.2 使用专用的特殊指令

ATMEGA16L 的汇编指令集中有很多专门用途的特殊指令, 花费极少的开销即可实现较复杂的功能。例如半字节交

作者简介: 胡启龙(1962-), 男, 讲师、硕士研究生, 主研方向: 通信技术; 冯明发, 副教授; 夏继媛, 讲师、硕士研究生

收稿日期: 2008-06-21 **E-mail:** q.hu@oa.szpt.net

换指令 SWAP、查表指令 LPM、立即数比较 CPI 等。

C 代码编译生成的汇编代码中，比较大小的方法通常是将两数相减，然后根据差的符号做出判断，包含了减法、条件跳转等很多代码，效率较低。而使用 CPI 指令，只需一个周期就可以得出结果，实现了比较大小的功能。将 CP 或 CPI 与 BRSH 等条件转移指令相配合，可极大地提高执行效率。

2.3 去除冗余代码

在 C 代码中每声明一个变量，编译器就要为这个变量开辟一个存储空间。每当变量值发生改变，编译生成的汇编代码就会将新的值存储到相应地址。可是很多情况下并不需要这样频繁地保存，例如某些公式中有很多要连续进行乘后累加，往往只需要最终结果，中间结果并不需要保存，通过汇编语言对寄存器的直接操作，可以省去这些冗余的存储操作。

2.4 多分支处理

C 程序中经常用变量作为 switch 语句的多分支选择条件，代码如下所示：

```
switch (i)
{
case 1: <语句 1>
case 2: <语句 2>
;
;
case n: <语句 n>
default: <语句 n+1>
}
```

编译生成的汇编代码在进入循环体之前，首先要判断是否符合($i=1$)的条件，如果不满足条件则跳过语句 1，执行后面的判断指令，依次判断下去。在实时性要求高的系统中，这种判断将会占用大量时间，影响系统的实时性和可靠性。如果能够保证变量 i 的值一定满足某个常量表达式，则执行语句之前的判断步骤就可以省去，这样可极大地节省时间，提高指令的执行效率。在 AVR 单片机中，可用间接跳转指令 IJMP 和直接跳转指令 RJMP 来实现，达到优化目的，优化后的汇编代码如下所示。

```
#asm
mov r30, low(brtimer)
mov r31, high(brtimer)
clr r24
add r30, _j
adc r31, r24
ijmp
brtimer: rjmp r0
rjmp r1
;
;
rjmp m
r0: ;
ret
r0: ;
ret
;
;
r0: ;
ret
#endasm
```

2.5 恰当选取寻址方式

ATMEGA16L 共有 11 种寻址方式，使用适当的寻址方式可以大大减少程序开销。例如，要计算某字符型序列，已知 $b[n]$ 和 $a[n]$ ，公式如下：

$$x(n)=b(n)+a(n) \quad (n=0,1,\dots,47)$$

若采用 C 语言编程，通常会用 for 循环来实现：

```
for(i = 0; i<48; i++)
{
x[i] = b[i]+a[i];
}
```

对于 AVR 单片机，计算连续存放的数据，使用带后增量或带预减量的存储器间接寻址方式极其简便高效，如 LD 和 ST 指令等。但是用于间接寻址的辅助寄存器数量有限，只有 X、Y、Z 3 个指针寄存器。当程序代码中定义的变量较多时，编译器对寻址方式的安排有可能极不合理，往往会把一些变量保存在以上 3 个指针寄存器中，无法起到间接寻址的作用。因此，在用汇编语言编程时，可用 register 和 volatile 修饰符来指定变量的存储空间不占用 X、Y、Z 指针寄存器，防止发生冲突。合理安排变量的寻址方式，也会给程序的运行效率带来很大提升。优化后的汇编代码如下所示：

```
ldi r31, high(xnaddr) ;x 序列首地址送 z 指针
ldi r30, low(xnaddr)
ldi r29, high(anaddr) ;a 序列首地址送 y 指针
ldi r28, low(anaddr)
ldi r27, high(bnaddr) ;b 序列首地址送 x 指针
ldi r26, low(bnaddr)
clr r22 ;计数器
xn: ld r20, y+ ;a[n]→r20 y+1→y
ld r21, x+ ;b[n]→r21 x+1→x
add r20, r21 ;a[n]+b[n]→r20
st z+, r20 ;a[n]+b[n]→x[n] z+1→z
inc r22
cpi r22, 47
brsh exit
rjmp xn
exit: ;
```

2.6 数据类型的灵活处理

数字系统的有限字长效应是系统实现时必须考虑的问题，相对于通用微处理器而言，DSP 芯片提供了更高精度的运算能力，而不增加运算时间。虽然通用微处理器提供了高精度的运算指令，如浮点数计算、字运算符等，但其并不是面向 DSP 技术而设计，高精度运算通常会使得运算时间大幅度增加。据笔者实际测试，一个乘法运算，分别用浮点数、字和字节类型数据实现，ATMEGA16L 芯片采用 C 语言编程，运算时间分别为 100 个、60 个、30 个时钟周期。另一方面，在任何实际应用中，并非所有 DSP 运算都要求很高的精度。通常在具体算法中，总存在一些中间数据可降低精度要求，因此，在编程时可在适当时刻对数据进行灵活处理，提高指令的执行效率。例如，用 ATMEGA16L 芯片计算一个无符号整型数 a 和无符号字符型数 b 的乘法，结果为字符型数 c ，如果不对中间数据作任何处理，采用 C 语言编程，代码很简单，即 $c=a*b$ ；但其编译后却生成了很多无用的汇编代码，如下所示：

```
;:115 max = a * b;
00007e 2fe2 mov r30, r18
00007f 2fa0 mov r26, r0x10
000080 2fb1 mov r27, r0x11
.endm
000081 27ff clr r31
000082 940e 0094 call _mulw12
000084 + _putw1r 19,20
000084 2f3e mov r0x13,r30
000085 2faf mov r0x14,r31
```

```

        _mulw12:
000094 d004 rcall _chksignw
000095 dff6 rcall _mulw12u
000096 f40e brtc _mulw12l
000097 dff0 rcall _anegw1
        _mulw12l:
000098 9508 ret
_chksignw:
000099 94e8 clt
00009a fff7 sbrs r31,7
00009b c002 rjmp _chksw1
00009c dfeb rcall _anegw1
00009d 9468 set
        _chksw1:
00009e ffb7 sbrs r27, 7
00009f c006 rjmp _chksw2
0000a0 95a0 com r26
0000a1 95b0 com r27
0000a2 9611 adiw r26,1
0000a3 f800 bld r0,0
0000a4 9403 inc r0
0000a5 fa00 bst r0,0
        _chksw2:
0000a6 9508 ret

```

若采用汇编语言编程，及时将中间无用数据舍弃，可极大地降低运算时间，其优化后的代码如下所示：

```

mul    r22,    r20 ;low(a)*b→r1:r0
mov    r24,    r0 ;结果暂存 r25:r24 中
mov    r25,    r1
mov    r22,    r21 ;high(a)*b→r1:r0
add    r0,    r25 ;舍弃低位 r24
clr    r23
adc    r1,    r23 ;结果存 r1 中

```

编程思想如图 1 所示。将 a 的高位和低位分别存放在寄存器 $r21$ 和 $r20$ 中， b 存放在寄存器 $r22$ 中，其中寄存器 $r24$ 中的内容就是中间变量，对最终结果没有贡献，因此，在计算时可直接舍弃。

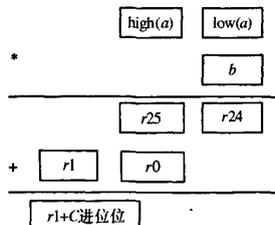


图 1 乘法编程思想

另外，对于整数除法，在已知被除数、除数和商的数据类型，即已知计算精度的前提下，可使被除数在适当时刻通过舍弃低位字节或采用移位指令 LSL, LSR, ROL, ROR 来达到程序优化的目的，移位指令的执行时间只有一个周期。

3 程序优化过程中的注意事项

3.1 合理安排寄存器

在用汇编语言实现上述算法时，要定义很多变量，包括中间变量，这些都要合理分配在不同寄存器及 SRAM 中。计算过程中直接对这些寄存器和 SRAM 进行修改，如果安排有冲突会直接影响到程序的运行结果，不同变量一定不能占用相同的 SRAM 空间。

3.2 监测程序运行结果

用汇编语言编程的目的是提高程序的运行效率，而不能影响系统的精度和稳定性，因此，要保证程序运行结果在合理范围内，可以制作一段测试序列用来监测运行结果是否正确。构造测试序列时要尽量涵盖各种典型情况，还要有足够的长度。如果测试序列构造不当，对这部分代码的正确性就无从检验了。

4 应用实例

在某无线数据通信控制系统中，接收机包括 2 个射频接收通道、1 个发射通道、2 个频率合成器、人机接口以及控制与信号处理单元，其系统框图如图 2 所示。

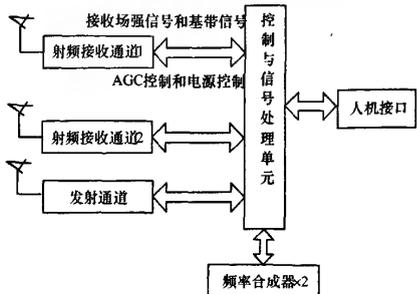


图 2 用于 MCU/DSP 的系统结构

频率合成器为射频发射和接收通道提供本振频率，控制单元主要实现人机界面操作，同时进行 AGC 控制和电源电压检测控制。信号处理单元接收来自于 2 个射频通道的场强信号和基带信号，根据场强信号计算出发射机与接收机之间的方向角度并进行校正；根据两路基带信号，利用初始位同步和跟踪位同步算法恢复出数据比特流^[5]，并根据数据通信协议执行相应的命令。同时为保证系统的稳定性，要实现自适应均衡控制算法。

笔者利用 ATMEGA16L 单片机，通过上述的程序优化策略，成功地实现了比特恢复、初始位同步、跟踪位同步和自适应均衡控制等算法，总的执行时间为 166 μ s。其中，在实现比特恢复算法时，笔者曾分别用 C 语言和汇编语言来编程，执行时间分别为 610 μ s 和 166 μ s，利用在某 I/O 引脚取反的测试程序抓拍得到的图像分别如图 3、图 4 所示。

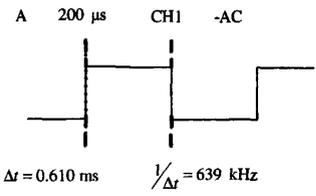


图 3 优化前运行时间

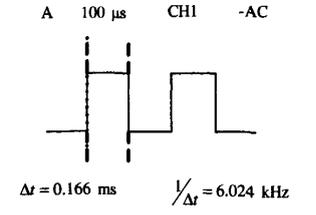


图 4 优化后运行时间

(下转第 229 页)

(5)wait1: FIFO 类型目标设备的等待读。如果 FIFO 中有了待传输的数据,状态机跳转至 wait2;否则一直等待在状态 wait1。

(6)wait2: FIFO 类型目标设备等待读。FIFO 中已经有了所要读取的数据,状态机跳转至 multiple。

(7)abort: 控制器回应放弃。一直维持该状态直到主设备声明传输结束,状态机跳转至 idle。

(8)single: 单数据传输。如果主设备声明最后一个数据传输,状态机跳转至 idle;如果主设备继续对该目标设备进行读写操作,状态机跳转至 retry。

(9)multiple: 数据突发传输。如果剩下最后一个字节传输,状态机跳转到 idle;如果地址信息无效或地址越界,状态机跳转到 retry;如果接收 FIFO 几乎满了而 PCI 主设备继续往其中写入数据,状态机跳转到 retry;其他情况状态机保持 multiple 状态不变。

(10)retry: 控制器回应重试。一直维持该状态直到主设备声明传输结束,状态机转至 idle。

目标设备通道通过状态机的状态和当前数据传输的情况产生控制信号,协调完成 PCI 总线上的数据传输,PCI 总线上的传输控制信号也是这样产生的。

DEVSEL#: 只要状态机没有处在 idle 或者 abort 状态下,该信号就有效。

TRDY#: 状态机处于 single 或者 multiple 状态下,该信号有效。

STOP#: 状态机处于 abort 或 retry 状态下,该信号有效。

3 目标控制器的验证

在完成目标控制器的 RTL 级功能设计后,必须对设计进行功能验证。据统计功能验证的工作量和投入的力度已经占到项目总投入的 60%~70%,它已经成为整个设计过程中最重要的一个环节^[4]。

验证使用了总线功能模型模拟 PCI 目标控制器在实际工作过程中所处的环境。在验证过程中,生成测试数据并将其输出到待验证的总线控制器中,监测模型对激励的响应过程,并报告结果。验证环境的组织结构如图 3 所示。

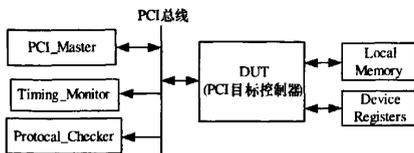


图 3 PCI 目标控制器验证环境

(上接第 226 页)

5 结束语

综上所述,利用通用微处理器实现 DSP 算法,实际上就是充分利用通用微处理器的片内资源,发挥通用微处理器潜在的计算能力,降低代码冗余度的过程。可运用上述优化策略,在提高速度的同时,极大地提高产品的性价比。虽然这些经验都是针对 ATMEGA16L 芯片的,但也同样适用 AVR 系列单片机的其他芯片,对于其他系列的通用微处理器也有一定的借鉴意义。

参考文献

[1] 纪宗南. DSP 实用技术和应用实例[M]. 北京: 航空工业出版社,

在图 3 中, PCI_Master 模拟总线主设备的功能,对 PCI 目标控制器(Design Under Test, DUT)施加各种传输类型的激励。Local Memory 和 Device Registers 2 个模型分别模拟本地的存储器和寄存器型目标,它们挂接在对应的 FIFO 型目标接口和寄存器型目标接口上。PCI 总线上还有 2 个行为模型, Timing_Monitor 和 Protocol_Checker 是总线上的时序和协议规范监测模块。

验证涉及的验证点有以下几个方面:配置寄存器读写,IO 空间立即读写,存储器空间单次、突发延迟读写,目标设备请求重试,目标设备放弃,主设备放弃。使用 Modelsim SE6.2 完成整个仿真过程,最终 PCI 目标控制器通过验证实现了目标设备功能,完全符合 PCI v2.2 规定。图 4 为 PCI 主设备突发读的仿真结果。

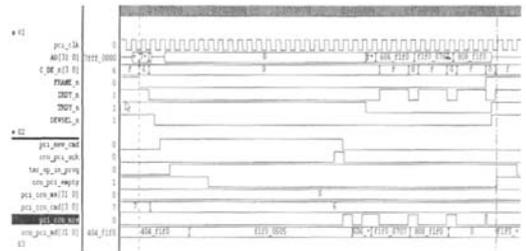


图 4 PCI 总线主设备对本地存储器的突发读

4 结束语

本文在深入研究 PCI 总线规范后,使用 Verilog 硬件描述语言完成了通用 PCI 目标控制器的 RTL 级设计和功能验证。经验证,该设计完全实现了目标控制器的功能,根据目标设备的不同类型,在目标控制器内实现不同的目标接口完成数据传输,使整个设计取得了较高的效率。目前本设计已成功应用于微电子中心多个 SoC 系统设计中。

参考文献

[1] Shanley T, Anderson D. PCI 系统结构[M]. 林 辉,译. 北京: 电子工业出版社, 2000.
 [2] 沈涵飞,甘 萌. PCI 总线目标控制器的设计与实现[J]. 计算机工程与设计, 2004, 25(11): 2101-2104.
 [3] 刘 红,李 勃,常 青,等. 基于 IP 核的 PCI 总线接口设计与实现[J]. 电子技术应用, 2006, 32(6): 6-9.
 [4] Bergeron J. Writing Testbenches——Functional Verification of HDL Models[M]. Boston, USA: Kluwer Academic Publisher, 2000.

编辑 顾逸斐

2006.

[2] 张雄伟,陈 亮,徐光辉. DSP 芯片的原理与开发应用[M]. 北京: 电子工业出版社, 2003.
 [3] Barnett R, Cull L, Cox S. Embedded C Programming and the Atmel AVR[M]. 北京: 清华大学出版社, 2003.
 [4] 丁化成,耿德根,李君凯. AVR 单片机应用设计[M]. 北京: 北京航空航天大学出版社, 2002.
 [5] Glover L, Grant P M. Digital Communications[M]. 北京: 机械工业出版社, 2006.

编辑 顾逸斐

基于AVR单片机的DSP算法

作者: [胡启龙](#), [冯明发](#), [夏继媛](#), [HU Qi-long](#), [FENG Ming-fa](#), [XIA Ji-yuan](#)
 作者单位: [深圳职业技术学院电子工程系, 深圳, 518055](#)
 刊名: [计算机工程](#) **ISTIC|PKU**
 英文刊名: [COMPUTER ENGINEERING](#)
 年, 卷(期): 2009, 35(5)
 引用次数: 0次

参考文献(5条)

1. 纪宗南 [DSP实用技术和应用实例](#) 2006
2. 张雄伟, 陈亮, 徐光辉 [DSP芯片的原理与开发应用](#) 2003
3. Barnett R, Cull L, Cox S [Embedded C Programming and the Atmel AVR](#) 2003
4. 丁化成, 耿德根, 李君凯 [AVR单片机应用设计](#) 2002
5. Glover L, Grant P M [Digital Communications](#) 2006

相似文献(9条)

1. 期刊论文 [张涛, 赵向阳. ZHANG Tao, ZHAO Xiangyang 基于AVR单片机的CAN总线设计](#) -现代电子技术2007, 30(10)
 CAN总线是几种最有前途的现场总线之一, 但是CAN总线开发需要一定的技术基础, 如何在较短的时间内开发出一套可靠性高、实时性好的CAN系统, 成为众多电子开发人员十分关注的问题. 采用AVR单片机作为微处理器, 设计一个用于气体监测的CAN总线网络, 给出了基本CAN节点的硬件原理图和程序流程图, 阐述CAN总线调试过程一些细节和经验.
2. 期刊论文 [刘光祖, 王建新, 邵名波. Liu Guangzhu, Wang Jianxin, Shao Mingbo TMS320C54XX DSP与AVR单片机的SPI及其编程](#) -电子工程师2005, 31(3)
 在微处理器之间的各种通信接口中, sPI(串行外围接口)以其显著的优点得到越来越广泛的应用. 文中通过将TMS320C5410的McBSP口配置为从SPI形式, 实现DSP与AVR单片机AT-mega128之间的串行通信. 经检验, 通信能可靠运行.
3. 学位论文 [臧宇恒 AVR 8位嵌入式单片机在车载全球定位系统显示终端中的应用](#) 2004
 全球定位系统最早应用在军事方面. 随着科学技术的发展和人们对全球定位系统的了解, 全球定位系统逐渐并且越来越广泛地应用到社会生产的各个领域. 全球定位系统的产品很多, 从手持全球定位系统到飞机、船只上使用的自动导航系统, 可以说是应有尽有. 在各种应用产品中, 车载全球定位系统无疑是发展最快的. 车载全球定位系统是伴随着人们对汽车理念的不断提高而产生的. 它向驾驶员提供准确、全面的交通信息, 使驾驶员摆脱以往缺乏交通信息指引的困扰, 大大提高了行车效率, 因此越来越受到驾驶人员的青睐. 车载全球定位系统主要是由GPS信号接收器和显示终端构成. 一般情况下只能通过购买专门的信号接收器来获得GPS信号. 显示终端通常是使用32位单片机或者DSP、CPLD等功能强大的微处理器处理接收到的GPS信号, 并控制液晶显示器输出电子地图和空间位置等信息. 不过这类微处理器及其开发平台价格比较高, 该设计探索一种全新的显示终端解决途径: 使用相对廉价的AVR 8位嵌入式单片机作为处理核心, 控制液晶显示器显示电子地图, 达到显示终端的基本要求. 该论文主要介绍AVR 8位嵌入式单片机在车载全球定位系统显示终端中的应用. 其中包括的主要内容有: ●介绍全球定位系统的原理、特点及其主要应用. ●车载全球定位系统显示终端的分析和方案设计. 包括车载全球定位系统的结构分析和显示终端各组成模块的选择. ●显示终端硬件电路系统设计. 包括硬件系统的总体设计和各个功能模块的具体实现. ●显示终端的系统软件设计. 包括电子地图的生成和各个功能模块的程序设计. ●系统调试. 通过调试, 使系统正确显示电子地图并实现图形操作. 该设计为车载全球定位系统显示终端提供了一种新颖、廉价的实现方式, 也为AVR 8位单片机的应用提供一种新思路, 具有较强的实用性、通用性和较高的市场价值.
4. 学位论文 [黄海泳 一种面向AVR单片机的嵌入式虚拟机](#) 2008
 随着电子技术和计算机技术的快速发展, 嵌入式系统已经被广泛地应用于人类生活的方方面面. 目前的嵌入式微处理器种类繁多, 体系结构也各不相同, 这降低了嵌入式软件的通用性, 进而阻碍了嵌入式系统的发展与普及. 如何能够合理重用现有的软件代码来提高软件开发效率已经成为嵌入式软件开发中的一个主要研究方向. 目前, 广泛应用的方法是采用嵌入式虚拟机技术来解决不同嵌入式平台的软件重用问题. 本文主要研究了一种基于ATmega128单片机的嵌入式虚拟机. 其中研究了嵌入式硬件平台, 包括原理图设计、PCB设计制作、硬件调试等; 通过对JAVA虚拟机的深入分析后, 借鉴其核心思想给出了基于AVR单片机的虚拟机规范定义; 重点研究了虚拟机的设计与实现, 且针对系统软件的高效性等特点, 使用C语言和汇编语言混合编程的方法开发出了嵌入式堆栈虚拟机(ESVM)原型; 研究了主机端开发环境的设计问题, 开发了基于MFC应用程序框架的可视化主机端软件, 该软件为用户提供了应用程序的编辑、编译等基本功能. 另外, 本文还开发了一个目标代码下载器, 使用该下载器把编译生成的目标代码通过串口下载到目标机端. 最后, 本文对系统的运行效果进行了验证.
5. 期刊论文 [王勇. Wang Yong 基于AVR单片机的一体化GPS接收机设计](#) -自动化仪表2008, 29(2)
 针对GPS接收机国产化低的现状, 以AVR ATmega128微处理器为核心处理, 基于GPS接收机、OEM板的基本原理, 应用一体化的设计思路, 开发研制了一体化GPS接收机. 经过样机的测试, 证明一体化GPS接收机具有操作简单、性能稳定、数据通信抗干扰能力强等特点, 其精度完全可以满足各种工程测量以及勘察测量的需求.
6. 期刊论文 [顾耀林, 翁丽华, 王正山, GU YAOLIN, WENG LIHUA, WANG ZHENGSHAN AVR单片机在多回路数据采集器中的应用](#) -微计算机信息2007, 23(35)
 介绍了AVR高档单片机MEGA系列的性能和特点, 提出了一种以ATmega128微处理器为核心器件的多回路、高精度、快速数据采集器的设计方案. 采集器系统通过模拟开关对8路互感器输入的电流以及开关量输入的报警和状态信号进行采集, 充分利用微处理芯片内A/D转换器进行数据处理和存储等, 并给出了AD采集处理的硬件接口电路和软件编程实现.
7. 期刊论文 [张洪, 董秀成. ZHANG Hong, DONG Xiu-cheng 基于AVR单片机的通用USB接口模块设计](#) -国外电子元器件2008(5)
 以AVR单片机ATmega8和USB接口器件PDIUSB12为核心, 基于标准的USB1.1协议, 设计一种通用USB接口模块, 以满足嵌入式系统中对USB接12的需求. 对模块的硬件电路或单片机固件程序的硬件接口层稍加修改即可用于其他各种微处理器. 该模块可为各种嵌入式系统增加USB接口, 实现与USB主机系统通信.

8. 期刊论文 [吴炯洋, 谢剑英, WU Jiong-yang, XIE Jian-ying 无刷直流电动机控制器设计 -控制工程](#)2008, 15(6)

针对无刷直流电动机的结构特点与控制要求, 设计了基于专用芯片MC33035和MC33039的无刷直流电动机控制器. 通过研究发现, 对控制信号中的转速信号进行预处理可以提高控制器的转速控制精度. 于是, 采用AVR单片机ATmega48对控制信号进行预处理, 同时使控制操作更为安全. 相对于完全使用微处理器或数字信号处理器的控制方案, 此方案可以简化软件的设计与调试, 并且降低成本. 实验结果表明, 转速控制精确. 该控制器实现了高性价比, 可以广泛地应用于无刷电动机控制领域.

9. 期刊论文 [何永泰, 肖丽仙, HE Yongtai, XIAO Lixian AVR单片机中移植 \$\mu\$ C/OS-II的研究 -微计算机应用](#)

2007, 28(1)

随着电子系统的复杂化和系统实时性需求的提高及应用软件朝着系统化方向发展的加速, 在单片机系统设计中引入操作系统是必然的. μ C/OS-II操作系统已在多种微处理器中移植成功, 在本文中以AVR单片机ATmega128为例详细介绍 μ C/OS-II操作系统在ATmega128中移植的过程.

本文链接: http://d.g.wanfangdata.com.cn/Periodical_jsjgc200905077.aspx

下载时间: 2009年12月7日