

顾问 田光哲 主编 宋争辉

CETTIC 职业培训指定教材

走向职业生涯

—— 课程体系与岗位对接

中国就业培训技术指导中心 组织编写
南 阳 师 范 学 院

2

河南人民出版社

CETTIC 职业培训指定教材

走向职业生涯

——课程体系与岗位对接

(物理与电子工程专业)

中国就业培训技术指导中心 组织编写
南 阳 师 范 学 院

顾 问 田光哲
主 编 宋争辉

河南人民出版社

本分册编委会

主 编	宋太平		
副主编	刘 毓	沈献博	
编 委	姚文华	鲁道邦	林青松
	乔贵春	黄义定	刘红钊
	罗长更	张 帅	海 涛
	宋玉玲		

前言

大学生专业课程体系与就业岗位技能对接培训是劳动和社会保障部和我院为适应新形势下大学生就业工作的要求,满足社会对高素质技能劳动者的迫切需求,促进技能劳动者就业而推出的重要举措。这种培训的特点是:以就业为导向,强化课程专业体系与岗位技能之间的联系,面向社会急需岗位有目的地开展专项技能培训,从而使大学生及有关人员掌握必需的岗位技能,实现就业和再就业。

当前影响大学毕业生就业的首要因素是大学学生的学科专业体系和从业岗位技能之间有脱节,存在空白地带,所学知识不能很好地适应就业岗位的需求,因而缺乏就业竞争力。对此,南阳师范学院领导未雨绸缪,在中国就业培训技术指导中心的大力支持下,对大学生开展专业课程体系与就业岗位技能之间的对接培训。我们编写这套教材目的在于帮助大学生通过有针对性的训练,尽快实现学科专业体系的理论向岗位技能的拓展,达到岗位要求,实现学科专业与岗位技能的对接,给大学毕业生一把就业的“金钥匙”。

本套教材是建立在大学生修完高等教育专业课程的基础上,以岗位——岗位技能——岗位核心技能——岗位核心技能点为主线,通过总操作程序——操作步骤——达到标准——注意事项——相关知识五个环节呈现技能培训内容,力求使教学训练更加简便易行,功能配套,加强针对性,体现技能培训的特点。

大学生专业课程体系与就业岗位技能培训将采用案例教学、实践教学、情景教学、师生互动等方式,让大学生多体验、多实践、多动手、多操作,采用“直接下水学游泳”的培训方式,让大学生在实践和体验中学到技能。

为了最大程度满足各类学生就业需要,在编写过程中,既考虑到高等院校和各类培训机构的需要,也要考虑到中等职业学校教学的实际需要。我们希望本套教材能为促进高等教育教学改革、提高大学生从业技能水平,起到积极的推动作用。

由于开展大学生学科专业体系与就业岗位对接培训工作是一个新事物,其教材内容定有不完善的地方,诚请广大同仁指正!



2008年7月

目 录

岗位 1 DSP 硬件技师	(1)
单元 1 DSP 硬件电路设计	(1)
模块 1 最小系统设计	(1)
模块 2 DSP 常用外围接口设计	(8)
模块 3 PCB 布线	(12)
单元 2 DSP 软件开发及开发集成软件 CCS	(17)
模块 1 DSP 软件开发	(17)
模块 2 集成开发环境 CCS 的基本应用	(20)
模块 3 CCS 的 DSP/BIOS 功能应用	(27)
单元 3 DSP 应用系统设计	(32)
模块 1 DSP 开发系统介绍	(32)
模块 2 应用实例	(34)
 岗位 2 ARM 嵌入式系统设计技师	(40)
单元 1 ARM 嵌入式系统设计/开发软件概述	(40)
模块 1 ADS 软件概述	(40)
模块 2 集成开发环境 Code Warrior 的使用	(41)
模块 3 AXD 调试器的使用	(43)
单元 2 ARM 嵌入式系统常见外设程序控制	(44)
模块 1 I/O 口控制	(44)
模块 2 UART 接口控制	(46)
模块 3 RTC 控制	(47)
模块 4 Watchdog 控制	(48)
模块 5 IIC 接口控制	(49)
模块 6 ADC 控制	(50)
模块 7 PWM DAC 控制	(51)
模块 8 步进电机控制	(52)
模块 9 IIS 接口控制	(54)
模块 10 LCD 显示接口控制	(55)
模块 11 触摸屏控制	(56)

单元 3	ARM 嵌入式系统开发硬件模块设计	(57)
模块 1	ARM 最小系统设计	(57)
模块 2	Flash 存储器模块设计	(64)
模块 3	SDRAM 存储器模块设计	(65)
模块 4	JTAG 调试接口模块设计	(68)
模块 5	串行口模块设计	(70)
模块 6	LCD 接口模块设计	(72)
模块 7	IIC 接口模块	(74)
模块 8	AD/DA 转换模块设计	(77)
单元 4	PCB 布局/布线	(78)
模块 1	PCB 概述	(78)
模块 2	PCB 布线	(79)
模块 3	PCB 布局	(81)
单元 5	基于 ARM 和 Linux 的开发	(81)
模块 1	Linux 操作系统简介	(81)
模块 2	Linux 开发过程	(82)
模块 3	Linux 开发环境建立	(83)
模块 4	Linux 内核移植	(85)
模块 5	Linux 应用程序	(90)
模块 6	Linux 驱动程序设计	(90)
模块 7	Linux 上的图形系统	(91)
岗位 3	PLC 设计技师	(93)
单元 1	STEP7 - Micro/WIN 编程软件使用说明	(93)
模块 1	编软件的安装	(93)
模块 2	STEP7 - Micro/WIN 简介	(93)
单元 2	程序设计	(99)
模块 1	电梯控制	(99)
模块 2	交通灯控制	(102)
模块 3	装配流水线控制	(106)
模块 4	挖掘机控制	(111)
模块 5	五相步进电机控制	(117)
单元 3	常见 PLC 选型	(121)
岗位 4	射频电路硬件设计技师	(122)
单元 1	射频仿真	(122)
模块 1	概述	(122)

模块 2 软件仿真	(122)
单元 2 射频电路设计	(128)
模块 1 放大器设计	(128)
模块 2 功率放大器设计	(134)
模块 3 滤波器设计	(137)
模块 4 振荡器设计	(139)
单元 3 高速 PCB	(141)
模块 1 高速 PCB 基本知识	(141)
模块 2 高速 PCB 设计	(143)
 岗位 5 单片机硬件设计技师	(148)
单元 1 SPCE061A 单片机及 61 板简介	(148)
单元 2 硬件电路设计	(149)
模块 1 内部电路设计	(149)
模块 2 SPCE061A 的输入/输出端口及接口电路	(150)
单元 3 集成开发环境 IDE	(155)
模块 1 集成开发环境的使用方法	(155)
模块 2 应用举例	(156)
单元 4 软件设计	(157)
模块 1 汇编指令	(157)
模块 2 C 语言程序设计	(160)
单元 5 常见音频算法及 SPCE061A 音频处理功能	(162)
模块 1 常用音频算法	(162)
模块 2 凌阳 61 的语音功能	(162)
单元 6 布线制图	(163)
模块 1 原理图与 PCB 设计	(163)
模块 2 电路仿真与可编程逻辑器件设计	(163)
单元 7 SPCE061A 单片机的应用举例	(163)
模块 1 单片机报时及作息时间控制	(163)
模块 2 热敏电阻温度计	(164)
 岗位 6 开关电源设计制造商	(165)
单元 1 模电、数电电路设计	(165)
模块 1 压控振荡电路设计	(165)
模块 2 误差取样检测电路设计	(166)
模块 3 电源滤波器设计	(167)
模块 4 低压 DC/DC 设计	(168)

模块 5	开关电路设计	(169)
模块 6	开关稳压电路设计	(169)
模块 7	电源散热设计	(171)
模块 8	常用电源芯片选择	(171)
单元 2	开关电源 PCB 板设计制造	(172)
模块 1	多层 PCB 板布线设计	(172)
模块 2	PCB 板的质量检测	(173)
模块 3	多层 PCB 板加工工艺	(173)
单元 3	开关电源的生产线调试	(175)
模块 1	常用仪器的使用	(175)
模块 2	电源国家标准的认识	(176)
模块 3	开关电源生产线调试	(176)
模块 4	开关电源的整机老化及测试	(177)
岗位 7	信息技术技师	(178)
单元 1	计算机和软件的安装与维护	(178)
模块 1	安装维护计算机	(178)
模块 2	系统备份和恢复	(200)
单元 2	常用软件的使用	(205)
模块 1	多媒体信息表达	(205)
模块 2	教学辅助软件的应用	(210)
模块 3	多媒体素材的制作与处理	(216)
单元 3	计算机和软件的安装与维护	(245)
模块 1	网络技术	(245)
模块 2	网络资源的检索与共享	(254)

岗位 1 DSP 硬件技师

单元 1 DSP 硬件电路设计

模块 1 最小系统设计

● 电源电路设计

【总操作程序】

主要分三个步骤,确定整个电路的电压→选择芯片→设计相应的电路图,具体操作如下:

1. 确定整个系统所需的电源

■ 根据系统开发需要考虑,一般最简单的 DSP 系统需要 3 种电源: DSP 的 CPU 电源及外设电源。不同的 DSP 以上电源电压值也不同,输出逻辑电压。

■ 确定其他外部模块所需电压,不同的其他外围芯片电源电压及驱动能力要求不同。

2. 选择芯片

■ 对于 DSP 芯片一般不进行电平转换,有专门的供电芯片。可以使用电平转换芯片得到逻辑控制电平。

■ 确定所需要的其他芯片。

3. 设计相应电源电路

■ 利用选定的芯片实现电路设计,外部模块可采用成熟的电源设计, DSP 芯片的电源最好使用专用电源芯片。

【达标标准】

使系统各个部分得到合适的供电电源和驱动电流。

【注意事项】

■ DSP 芯片的电源根据不同型号的不同需要来选择芯片。

■ 逻辑总线驱动最好用专用芯片驱动。

■ 整个电源电路的设计要考虑性价比,在满足电压需要的同时要考虑到驱动电流。

■ 合适的选用电阻阻值,实现高逻辑电平,最好使用排阻,可以减小体积。

■ DSP 输出给 5V 的电路(如 D/A),无需加任何缓冲电路,可以直接连接。

■ DSP 输入 5V 的信号(如 A/D),由于输入信号的电压 > 4V,超过了 DSP 的电源电压, DSP 的外部信号没有保护电路,需要加缓冲,如 74LVC245 等,将 5V 信号变换成 3.3V 的信号。

■ 仿真器的 JTAG 口的信号也必须为 3.3V, 否则有可能损坏 DSP。

【相关知识】

■ DSP 芯片采用双电源供电机制:1.8V 对应 CPU 内核电源(CVdd), 主要是为 CPU 内部逻辑提供电流, 包括 CPU、时钟电路和所有片上外围电路;3.3 伏对应 I/O 电源(DVdd), 为外部接口引脚提供电压, 这样可以直接与外部低压器件接口, 而不需要额外的电平转换电路。对于不同型号的 DSP 芯片采用不同的电源芯片, TI 公司的芯片有对应的专门电源芯片。

■ TMS320LF24xx:TPS7333QD, 5V 变 3.3V, 最大 500mA。

■ TMS320VC33:TPS73HD318PWP, 5V 变 3.3V 和 1.8V, 最大 750mA。

■ TMS320VC54xx:TPS73HD318PWP, 5V 变 3.3V 和 1.8V,
最大 50mA;TPS73HD301PWP, 5V 变 3.3V 和可调, 最大 750mA。

■ TMS320VC55xx:TPS73HD301PWP, 5V 变 3.3V 和可调, 最大 750mA。

■ TMS320C6000:PT6931, TPS56000, 最大 3A。

● 复位电路设计

【总操作程序】

主要分三个步骤, 确定复位要求→选择芯片→设计相应的电路图, 具体操作如下:

1. 确定复位要求

■ 不同的系统要求, 复位要求不同。对于可靠性和稳定性要求比较高的系统, 对复位电路较高, 由于 DSP 系统的时钟频率较高, 在运行时难以避免发生干扰和被干扰的现象, 严重时系统会现死机或程序“跑飞”现象。为了克服这种情况, 除了在软件上作一些保护措施外, 硬件上也必须作相应的处理。硬件上最有效的保护措施就是采用所谓的“看门狗”(Watch dog)电路。“看门狗”电路就是具有监视功能的自动复位电路, 需要监督系统的运行, 而一般的系统只要求能够复位即可。

■ 是否要求对 DSP 电压进行监督。

2. 选择芯片

■ 对于简单要求的复位电路, 只要采用 RC 电路充放电回路即可实现。

■ 对于要求“看门狗”电路的复位电路, 确定所需要的“看门狗”电路芯片。

3. 设计相应复位电路

■ 利用确定的芯片型号实现电路设计。

【达标标准】

能够满足复位实现的要求, 能够实现 DSP 工作状态监督。

【注意事项】

■ 使用分立元件实现时要注意电路参数的计算, 以及使用常用的电路参数。

■ 编写复位中断服务程序, 根据系统的要求初始化 DSP 芯片。

【相关知识】

■ 为使芯片初始化正确, 一般应保证 RS 为低至少持续 5 个 CLKOUT 周期, 即当速度为 25ns 时约为 125ns。但是, 由于在上电后, 系统的晶体振荡器往往需要几百毫秒的稳定

时间,所以,RS 为低的时间主要由系统的稳定时间确定,一般为 100ms—200ms。其复位时间由 R 和 C 确定。取 $R=100k$, $C=4.7\mu F$,复位时间约为 167ms,满足复位要求。

■ 常用看门狗芯片 MAX705/706 介绍:

1 引脚 MR:手动复位输入,低电平有效(要求低于 0.8 伏)。

2 引脚 VCC 电源。3 引脚 GND,4 引脚 PFI 掉电输入检测,输入低于 1.25 伏时,5 引脚 PFO 高输出。6 引脚 WDI 看门狗输入检测,输入超过 1.6 秒不变时引脚 8 看门狗输出 WDO 输出低电平复位信号。7 引脚输出复位信号。

● 时钟电路设计

【总操作程序】

主要分四个步骤,确定外部时钟还是内部时钟→确定时钟频率,选择芯片→设计相应的电路图→确定硬件配置时钟参数并用电路实现。具体操作如下:

1. 确定使用外部有源晶振时钟还是内部时钟(外接晶体)

■ 系统中要求多个不同频率的时钟信号时,首选可编程时钟芯片,这样有利于时钟信号的同步。

■ 单一时钟信号时,一般的应用建议选择晶体时钟电路。

■ 多个同频时钟信号时,可选择有源的晶振作为时钟电路。

■ 尽量使用 DSP 片内的 PLL,降低片外时钟频率,提高系统的稳定性。

2. 选择芯片

■ 需要倍频的 DSP 需要配置好 PLL 周边配置电路。

■ 20MHz 以下的晶体晶振基本上都是基频的器件,稳定度好,20MHz 以上的大多是谐波的(如 3 次谐波、5 次谐波等等),稳定度差,因此强烈建议使用低频的器件,毕竟倍频用的 PLL 电路需要的周边配置主要是电容、电阻、电感,其稳定度和价格方面远远好于晶体晶振器件。

■ 确定所需要的其他芯片,主要是隔离和滤波。

3. 设计相应时钟电路

■ 利用选定的芯片实现电路设计,外接时钟电路模块可采用成熟的电路设计,内部时钟电路时只用一片晶体和两个电容,按照 DSP 资料连接电路即可。

【达标标准】

使 DSP 得到工作需要的时钟。

【注意事项】

■ DSP 芯片的工作时钟不能超过其最大主频,可以降频使用。

■ 软件和硬件电路配置结合起来使用,可以使较低的外部时钟,倍频得到 DSP 的高频时钟。

■ 整个时钟电路的设计要考虑性价比,大范围用的产品开发一般考虑用内部时钟。

■ 合适的选用电阻阻值,实现硬件管脚高逻辑电平配置,最好使用排阻,可以减小体积。

■ C6000、C5510、C5409A、C5416、C5420、C5421 和 C5441 等 DSP 片内无振荡电路,不

能用晶体时钟电路。

■ VC5401、VC5402、VC5409 和 F281x 等 DSP 时钟信号的电平为 1.8V,建议采用晶体时钟电路。

【相关知识】

■ 无源晶体与有源晶振的区别、应用范围及用法:

无源晶体——无源晶体需要用 DSP 片内的振荡器,在 datasheet 上有建议的连接方法。无源晶体没有电压的问题,信号电平是可变的,也就是说是根据起振电路来决定的,同样的晶体可以适用于多种电压,可用于多种不同时钟信号电压要求的 DSP,而且价格通常也较低,因此对于一般的应用如果条件许可建议用晶体,这尤其适合于产品线丰富批量大的生产者。无源晶体相对于晶振而言其缺陷是信号质量较差,通常需要精确匹配外围电路(用于信号匹配的电容、电感、电阻等),更换不同频率的晶体时周边配置电路需要做相应的调整。建议采用精度较高的石英晶体,尽可能不要采用精度低的陶瓷晶体。

有源晶振——有源晶振不需要 DSP 的内部振荡器,信号质量好,比较稳定,而且连接方式相对简单(主要是做好电源滤波,通常使用一个电容和电感构成的 PI 型滤波网络,输出端用一个小阻值的电阻过滤信号即可),不需要复杂的配置电路。有源晶振通常的用法:一脚悬空,二脚接地,三脚接输出,四脚接电压。相对于无源晶体,有源晶振的缺陷是其信号电平是固定的,需要选择好合适输出电平,灵活性较差,而且价格高。对于时序要求敏感的应用,一般认为还是有源的晶振好,因为可以选用比较精密的晶振,甚至是高档的温度补偿晶振。有些 DSP 内部没有起振电路,只能使用有源的晶振,如 TI 的 6000 系列等。有源晶振相比于无源晶体通常体积较大,但现在许多有源晶振是表贴的,体积和晶体相当,有的甚至比许多晶体还要小。

●外部存储器电路设计

【总操作程序】

主要分三个步骤,确定外部存储器容量及最小所需的读写速度→选择芯片→设计相应的接口电路,具体操作如下:

1. 确定外部存储器容量及最小所需的读写速度

■ 根据系统开发需要考虑,确定外部程序存储器和外部数据存储器,外部程序存储器容量取决于系统开发的代码的大小,可通过仿真得到,外部数据存储器容量,由系统运行所需数据量所占的空间确定,一般要有一定的富裕量。

■ 如果系统要求的容量超过了 DSP 的外部寻址能力,则需要考虑进行扩展外部存储空间。

■ 根据系统实时性要求及系统硬件仿真(在开发板上进行)的运行时间来确定自己系统的外部存储器的最小读取速度。

2. 选择芯片

■ 根据第一步选择芯片,考虑芯片的成本及性能。

■ 确定所需要的其他芯片,包括确定是否需要硬件等待电路,如需要相应的芯片型号等。

■ 如果需要扩展电路,则需要确定所需的译码控制电路的主要芯片。

3. 设计相应接口电路

■ 利用选定的芯片,根据 DSP 接口原理,实现电路设计。

■ 根据 CPU 主频,确定需要插入的软件等待个数。

【达标标准】

使系统的数据空间和程序空间能够满足需要。

【注意事项】

■ 系统中没有的存储区,设定等待状态为 0 或 1。以防止引脚受干扰引出问题。

■ 与内部有效的 RAM 相同的地址,外部 RAM 无效。

■ 当只插入 2~7 个软件等待周期时,引脚 MSC 与 READY 相连。

■ 如果所有外部器件均不插等待周期,或仅插入 1 个等待周期,则将 READY 引脚接固定高电平。

■ 硬件等待周期是在 2~7 个软件等待周期基础上插入的。

■ 硬件等待周期是利用 READY、MSC 以及外部电路产生的。

■ 若 DSP 芯片的机器周期为 T, CPU 运行速度为 $1/TMIPS$,则要求外部器件的存取时间小于 T 的 60%。

■ 如果没有扩展存储器,那么要 DSP 相应控制管脚可以和存储器直接相连,注意存储器如果数据输出电平和 CPU 核电压电平不符则要加电平转换电路。

【相关知识】

■ DSP 无论是运算还是存取数据,速度都很快,但外部存储器或其他设备的读写周期都较长。因此经常用等待方式访问外存储器。

■ DSP 有软等待(内等待)、硬等待(外等待)访问控制以便于与不同速度的外围器件交换数据,同时 DSP 自身的运行速度又可以保持很高。软、硬件等待都可以分别对不同类型、不同地址范围的外设产生不同的等待状态数。

■ 复位时, $SWWSR = 7FFFh$,各空间均插入 7 个等待周期。

● 并行 I/O 接口电路设计

【总操作程序】

主要分四个步骤,确定通过并行 I/O 接口连接的设备→选择芯片→设计相应的接口电路→编写设备的驱动程序,具体操作如下:

1. 确定通过并行 I/O 接口连接的设备

■ 根据系统开发需要考虑,确定系统与 DSP 并行交换数据的外部设备。比如键盘,显示器等设备。

■ 也可通过并行 I/O 接口实现 DSP 与并行的 ADC 或者 DAC 连接。

■ 根据系统实时性要求及设备读写速度来确定自己系统的外部 I/O 设备的最小读取速度。

2. 选择芯片

■ 根据第一步不同的外部设备选择芯片,考虑芯片的成本及性能。

■ 确定所需要的其他芯片,包括确定是否需要硬件等待电路,如需要相应的芯片型号等。

■ 如果需要扩展电路更快的实现接口,则需要确定所需的译码控制电路的主要芯片。

3. 设计相应接口电路

■ 利用选定的芯片,根据 DSP 接口原理,实现电路设计。

■ 根据 CPU 主频,确定需要插入的软件等待个数。

4. 编写设备的驱动程序

根据硬件的连接,及设备工作原理,设计 DSP 对设备的驱动程序:即通过软件编程,利用硬件连接实现对外部设备的控制,使之正常工作。

【达标标准】

系统实现对外部设备控制,使外部设备能够正常工作。。

【注意事项】

■ 系统中没有的用到的 I/O 空间,设定等待状态为 0 或 1。以防止引脚受干扰引出问题。

■ 当只插入 2~7 个软件等待周期时,引脚 MSC 与 READY 相连。

■ 如果所有外部器件均不插等待周期,或仅插入 1 个 T,则将 READY 引脚接固定高电平。

■ 硬件等待周期是在 2~7 个软件等待周期基础上插入的。

■ 硬件等待周期是利用 READY、MSC 以及外部电路产生的。

【相关知识】

■ 插入一个等待周期,每次 I/O 读写操作都延长一个机器周期。

■ 如果 I/O 读/写操作紧跟在存储器读/写操作之后,则 I/O 读/写操作至少 3 个机器周期,如果存储器读操作紧跟在 I/O 读/写操作之后,则存储器读操作至少 2 个机器周期。

■ 复位时,SWWSR = 7FFFh,各空间均插入 7 个 T。

■ 对于常用的并行 I/O 设备,TI 公司给出的典型的硬件接口电路,以及典型的驱动程序设计,用户可以免费得到。在 DSP 的集成开发环境 CCS 中,可以通过 DSP/BIOS API 调用,实现驱动程序,及其调度。

● BOOT 设计

【总操作程序】

主要分四个步骤,根据系统及连接的外部 ROM 类型确定 BOOT 方式→编写初始化程序→设计 BOOTLOADER 程序→烧写程序。具体操作如下:

1. 根据系统及连接的外部 ROM 类型确定 BOOT 方式

■ 根据系统开发需要考虑,确定系统是否采用主机接口引导,可实现由主机控制系统的启动。

■ 从主处理器通过以下途径加载:

■ HPI 总线。

■ 8 位或 16 位并行 I/O 口,从外部 8 位或 16 位 EPROM 加载。

■ 任何一个串行口。

■ 从用户定义的地址自举。

■ 如果代码存储在 I/O 空间时可采用异步引导模式。

2. 编写初始化程序

■ 在上电复位时,如果引脚 MP/MC 为低电平,则 C54x 从片内 ROM 的 FF80h 处开始执行程序。在这里有一条跳转指令,转向 ROM 中已掩模的加载程序。在此之前必须先初始化 CPU 的状态器:全局禁止中断($INTM = 1$),内部 DARAM 和 SRAM 均映射到程序/数据存储器空间($OVLY = 1$)。

3. 设计 BOOTLOADER 程序

■ 根据系统运行需要编写 BOOTLOADER。

■ 变换序存储格式。

4. 烧写程序

根据硬件的连接,及设备工作原理,编写烧写程序。

【达标标准】

系统能够成功实现自启动。

【注意事项】

■ 仔细检查 BOOT 的控制字是否正确。

■ 仔细检查外部管脚设置是否正确。

■ 仔细检查 hex 文件是否转换正确。

■ 用仿真器跟踪 BOOT 过程,分析 BOOT 过程,判断其正确与否。

■ $MP/MC = 0$ (片上 ROM 有效),复位后程序从 FF80 开始运行,将跳转到 BOOT-LOADER 代码开始出。

■ 对于 C5000,片内的 BOOT 程序在上电后将数据区的内容,搬移到程序区的 RAM 中,因此 FLASH 必须在 RESET 后放在数据区。

■ 由于 C5000,数据区的空间有限,一次 BOOT 的程序不能大于 48K。解决的方法如下:

1. 在 RESET 后,将 FLASH 译码在数据区,RAM 放在程序区,片内 BOOT 程序将程序 BOOT 到 RAM 中。

2. 用户初试化程序发出一个 I/O 命令(如 XF),将 FLASH 译码到程序区的高地址。开放数据区用于其它的 RAM。

3. 用户初试化程序中包括第二次 BOOT 程序(此程序必须用户自己编写),将 FLASH 中没有 BOOT 的其它代码搬移到 RAM 中。

4. 开始运行用户处理程序。

【相关知识】

■ 为了保证引导成功,EPROM 的数据组织必须遵守 BOOTLOADER 的规定的数据存放格式。利用 DSP 的编译工具可以很方便的生成符合引导规定的数据存放格式。这个工具是十六进制转换工具,该工具使用方法如下:

HEX500 -e 1000h -boot -i -bootorg 08000H
- memwidth 8 filemane

■ 其中最主要的是选项的正确使用。决定引导程序识别的数据格式的主要选项包括:

-e VALUE 指定引导程序装入 DSP 的程序存储器的起始地址。

■ -boot 转换生成引导表。

■ -i 生成 intel 格式的 EPROM 数据格式。

■ -bootorg VALUE 说明引导程序从外部数据存储器引导数据的起始地址。

■ -memwidth VALUE 定义外部数据存储器的字宽。

■ 另外还可以使用 -swsr VALUE 来设置插入等待周期;使用 -bscr VALUE 来设置存储器块大小;使用 romwidth VALUE 来设置 EPROM 大小。

模块 2 DSP 常用外围接口设计

● AD 和 DA 接口电路设计

【总操作程序】

主要分四个步骤,确定整个系统取样频率→选择芯片→设计相应的接口电路图→接口软件编程,具体操作如下:

1. 确定整个系统取样频率

■ 根据系统开发需要考虑,确定整个系统的数模或模数转换的频率。

■ 确定整个系统的数模或模数转换的动态范围。

2. 选择芯片

■ ADC 的种类:

逐次比较 ADC

双线性 ADC

FLASH ADC

$\Sigma - \delta$ ADC

■ ADC 选择的标准:芯片的价格、分辨率、速度、输入动态范围、POWER,串/并行输出等。

■ DAC 的选择的标准:芯片的价格、速度、输入动态范围、POWER,串/并行输出等。

■ DAC 内部是否包含滤波器,如果没有,则要考虑滤波器设计。

■ 确定所需要的其他芯片。

3. 根据所确定的 AD/DA 芯片搭建和 DSP 的接口电路

■ 对于串行的 AD/DA 一般连接 DSP 的串口,根据标准串口的电路原理以及 AD/DA 的电路特性进行握手通信设计。

■ 对于并行的 AD/DA 一般作为 I/O 空间的一个地址连接 DSP 的并行 I/O 口上。并行采样系统进行采样时,每次数据采样完,都向处理器 INT 管脚(任意一个都可以)发出 DMA 同步信号,处理器接到同步信号后自动调用 DMA 功能,把采样数据自动放入内存。

4. 接口软件编程

串口连接的情况:

■ DSP 串口的初始化。

■ AD50C 初始化。

■ 用户代码的编写。

I/O 空间并口连接的情况:

■ 首先对 DMA 通道进行初始化,在初始化过程中应确定所占用的 DMA 通道,引起 DMA 工作的同步事件,DMA 所要处理的数据的源地址和目的地址,源地址以及目的地址所处的空间是在程序空间、数据空间还是 I/O 空间,其源地址、目的地址是否累加等。

■ 编写中断服务程序,及修改中断向量表。

【达标标准】

使系统能够正常的接收离散数据和输出模拟信号。

【注意事项】

■ C5000 系列 DSP 的中断向量可以重新定位。但是它只能被重新定位到 Page0 范围内的任何空间。

■ DSP 输出给 5V 的电路(如 D/A),无需加任何缓冲电路,可以直接连接。

■ DSP 输入 5V 的信号(如 A/D),由于输入信号的电压 > 4V,超过了 DSP 的电源电压,DSP 的外部信号没有保护电路,需要加缓冲,如 74LVC245 等,将 5V 信号变换成 3.3V 的信号。

【相关知识】

■ 定点 DSP 的定标

1) 一般定点 DSP 有两种定标表示法:Q、S

Q15—S0.15— $0 \leq X \leq 0.9999695$

Q5—S10.5— $1024 \leq X \leq 1023.96875$

2) 浮点数(x)转换成定点数(xq):

$$xq = (\text{int}) X * 2Q$$

3) 定点数(xq)转换成浮点数(x):

$$x = (\text{float}) xq * 2Q$$

4) C54x 通过归一化支持定标操作:

相关指令 EXP、NORM

● 串口通信电路设计

【总操作程序】

主要分三个步骤,选择芯片→设计相应的接口电路图→接口软件编程,具体操作如下:

1. 选择芯片

■ 根据系统功能需要,选定要通信的设备芯片。

■ 确定所需要的其他芯片。