做 DSP 之前我们该弄明白哪些基本问题

做DSP之前我们该弄明白哪些基本问题 1

一、时钟和电源

问：DSP的电源设计和时钟设计应该特别注意哪些方面？外接晶振选用有源的好还是无源的好？
答：时钟一般使用晶体，电源可用TI的配套电源。外接晶振用无源的好。

问：TMS320LF2407 的A/D转换精度保证措施。
答：参考电源和模拟电源要求干净。

问：系统调试时发现纹波太大，主要是哪方面的问题？
答：如果是电源纹波大，加大电容滤波。

问：请问我用 5V供电的有源晶振为DSP提供时钟，是否可以将其用两个电阻进行分压后再接到DSP的时钟输入端，这样做的话，时钟工作是否稳定？
答：这样做不好，建议使用晶体。

问：一个多DSP电路板的时钟，如何选择比较好？DSP电路板的硬件设计和系统调试时的时序问题？
答：建议使用时钟芯片，以保证同步。硬件设计要根据DSP芯片的时序，选择外围芯片，根据时序设定等待和硬件逻辑。

二．干扰与板的布局

问：器件布局应重点考虑哪些因素？例如在集中抄表系统中？
答：可用TMS320VC5402，成本不是很高。器件布局重点应是存贮器与DSP的接口。

问：在设计DSP的PCB板时应注意哪些问题？
答：1.电源的布置；2.时钟的布置；3.电容的布置；4.终端电路；5.数字同模拟的布置。

问：请问DSP在与前向通道(比如说AD)接口的时候，布线过程中要注意哪些问题，以保证AD采样的稳定性？
答：模拟地和数字地分开，但在一点接地。

问：DSP主板设计的一般步骤是什么？需要特别注意的问题有哪些？
答：1.选择芯片；2.设计时序；3.设计PCB。最重要的是时序和布线

问：在硬件设计阶段如何消除信号干扰（包括模拟信号及高频信号）？应该从那些方面着手？

答：1.模拟和数字分开；2.多层板；3.电容滤波。

问：在电路板的设计上，如何很好的解决静电干扰问题。
答：一般情况下，机壳接大地，即能满足要求。特殊情况下，电源输入、数字量输入串接专用的防静电器件。

问：DSP板的电磁兼容（EMC）设计应特别注意哪些问题?
答：正确处理电源、地平面，高速的、关键的信号在源端串接端接电阻，避免信号反射。

问：用电感来隔离模拟电源和数字电源，其电感量如何决定？是由供电电流或噪音要求来决定吗？有没有计算公式？
答：电感或磁珠相当于一个低通滤波器，直流电源可以通过，而高频噪声被滤除。所以电感的选择主要决定于电源中高频噪声的成分。

问：讲座上的材料多是电源干扰问题，能否介绍板上高频信号布局（Layout）时要注意的问题以及数字信号对模拟信号的影响问题？
答：数字信号对模拟信号的干扰主要是串扰，在布局时模拟器件应尽量远离高速数字器件，高速数字信号尽量远离模拟部分，并且应保证它们不穿越模拟地平面。

问：能否介绍PCB布线对模拟信号失真和串音的影响，如何降低和克服？
答：有 2 个方面，1. 模拟信号与模拟信号之间的干扰：布线时模拟信号尽量走粗一些，如果有条件，2 个模拟信号之间用地线间隔。2. 数字信号对模拟信号的干扰：数字信号尽量远离模拟信号，数字信号不能穿越模拟地。

三．DSP性能
问：1.我要设计生物图像处理系统，选用那种型号较好（高性能和低价格）？2.如果选定TI DSP，需要什么开发工具？
答：1.你可采用C54x 或 C55x平台，如果你需要更高性能的，可采用C6x系列。2.需要EVMs和XDS510 仿真器。

问：请介绍一种专门用于快速富利叶变换（FFT）， 数字滤波，卷积，相关等算法的DSP，最好集成 12bit以上的ADC功能。
答：如果你的系统是马达/能量控制的，我建议你用TMS320LF240x。详情请参阅DSP选择指南：http://www.dspvillage.ti.com/dspguide。

问：有些资料说DSP比单片机好，但单片机用的比DSP广。请问这两个在使用上有何区别？
答：单片机一般用于要求低的场合，如 4/8 位的单片机。DSP适合于要求较高的场合。

问：我想了解在信号处理方面DSP比FPGA的优点。
答：DSP是通用的信号处理器，用软件实现数据处理；FPGA用硬件实现数据处

理。DSP的成本便宜，算法灵活，功能强；FPGA的实时性好，成本较高。

问：请问减小电路功耗的主要途径有哪些？
答：1.选择低功耗的芯片；2.减少芯片的数量；3.尽量使用IDLE。

问：用C55 设计一个低功耗图像压缩/解压和无线传输的产品，同时双向传输遥控指令和其他信息，要求图像 30 帧/秒，TFT显示 320*240，不知道能否实现？若能，怎样确定性能？选择周边元器件？确定最小的传输速率？能否提供开发的解决方案？软件核？
答：1.有可能，要看你的算法。2.建议先在模拟器上模拟。

问：用DSP开发MP3，比较专用MP3 解码芯片如何，比如成本、难度、周期？谢谢。
答：1.DSP的功能强，可以实现附加的功能，如ebook等；2.DSP的性能价格比高；3.难度较大，需要算法，因此周期较长，但TI有现成的方案。

问：用DSP开发的系统跟用普通单片机开发的系统相比，有何优势？DSP一般适用于开发什么样的系统？其开发周期、资金投入、开发成本如何？与DSP的接口电路是否还得用专门的芯片？
答：1.性能高；2.适合于速度要求高的场合；3.开发周期一般 6 个月，投入一般要一万元左右；4.不一定，但需要速度较高的芯片。

问：DSP会对原来的模拟电路产生什么样的影响？
答：一方面DSP用数字处理的方法可以代替原来用模拟电路实现的一些功能；另一方面，DSP的高速性对模拟电路产生较大的干扰，设计时应尽量使DSP远离模拟电路部分。

问：请问支持MPEG-4 芯片型号是什么？
答：C55x或 C6000 或DSC2x

问：DSP内的计算速度是快的，但是它的I/O口的交换速度有多快呢？
答：主频的 1/4 左右。

四．技术性问题
问：我有二个关于C2000 的问题：1、C240 或C2407 的RS复位引脚既可输入，也可输出，直接用CMOS门电路（如 74ACT04）驱动是否合适，还是应该用OC门（集电极开路）驱动？2、大程序有时运行异常，但加一两条空指令就正常，是何原因？
答：1、OC门（集电极开路）驱动。2、是流水线的问题。

问：1.DSP芯片内是否有单个的随机函数指令？2．DSP内的计算速度是快的，但是它的I/O口的交换速度有多快呢？SP如何配合EPLD或FPGA工作呢？

答：1.没有。2.取决于你所用的I/O。对于HPI，传输速率（字节）大约为CPU的1/4，对McBSP，位速率（kbps）大约为CPU的 1/2。3.你可以级联仿真接口和一个 EPLD/FPGA 在一起。请参考下面的应用手册： http://www.ti.com/sc/docs/psheets/abstract/apps/spra439a.htm

问：设计DSP系统时，我用C6000 系列。DSP引脚的要上拉，或者下拉的原则是怎样的？我经常在设计时为某一管脚是否要设置上/下拉电阻而犹豫不定。

答：C6000 系列的输入引脚内部一般都有弱的上拉或者下拉电阻，一般不需要考虑外部加上拉或者下拉电阻，特殊情况根据需要配置。

问：我正在使用TMS320VC5402，通过HPI下载代码，但C5402 的内部只提供 16K 字的存储区，请问我能通过HPI把代码下载到它的外部扩展存储区运行吗？

答：不行，只能下载到片内。

问：电路中用到DSP，有时当复位信号为低时，电压也属于正常范围，但DSP加载程序不成功。电流也偏大，有时时钟也有输出。不知为什么？

答：复位时无法加载程序。

问：DSP和单片机相连组成主从系统时，需要注意哪些问题？

答：建议使用HPI接口，或者通过DPRAM连接。

问：原来的DSP的程序需放在EPROM中，但EPROM的速度难以和DSP匹配。现在是如何解决此问题的？

答：用BootLoad方法解决。

问：我在使用 5402DSK时，一上电，不接MIC，只接耳机，不运行任何程序，耳机中有比较明显的一定频率的噪声出现。有时上电后没有出现，但接MIC，运行范例中的CODEC程序时，又会出现这种噪声。上述情况通常都在DSK工作一段时间后自动消失。我在DSP论坛上发现别人用DSK时也碰到过这种情况，我自己参照 5402DSK做了一块板，所用器件基本一样，也是这现象，请问怎么回事？如何解决？

答：开始时没有有效的程序代码，所以上电后是随机状态，出现这种情况是正常的。

问：我使用的是TMS320LF2407，但是仿真时不能保证每次都能GO MAIN。我想详细咨询一下，CMD文件的设置用法，还有VECTOR的定义。

答：可能看门狗有问题，关掉看门狗。有关CMD文件配置请参考《汇编语言工具》第二章。

问：我设计的TMS320VC5402 板子在调试软件时会经常出现存储器错误报告，排除是映射的问题，是不是板子不稳定的因素？还是DSP工作不正常的问题？如何判别？

答：你可以利用Memoryfill功能，填入一些数值，然后刷新一下，看是不是在变，如果是在变化，则Memory 是有问题。

问：如何解决Flash编程的问题:可不可以先用仿真器下载到外程序存储RAM中，然后程序代码将程序代码自己从外程序存储RAM写到F240 的内部Flash ROM中，如何写?
答：如果你用F240，你可以用下载TI做的工具。其它的可以这样做。
问：C5510 芯片如何接入E1 信号？在接入时有什么需要注意的地方？
答：通过McBSP同步串口接入。注意信号电平必须满足要求。

问：请问如何通过仿真器把.HEX程序直接烧到FLASH中去?所用DSP为 5402 是否需要自己另外编写一个烧写程序，如何实现?谢谢!!
答：直接写.OUT。是DSP中写一段程序，把主程序写到FLASH中。

问：DSP的硬件设计和其他的电路板有什么不同的地方？
答：1.要考虑时序要求；2.要考虑EMI的要求；3.要考虑高速的要求；4.要考虑电源的要求。

问：ADS7811，ADS7815，ADS8320，ADS8325，ADS8341，ADS8343，ADS8344，ADS8345 中，哪个可以较方便地与VC33 连接，完成 10 个模拟信号的AD转换（要求 16bit，1 毫秒内完成 10 个信号的采样，当然也要考虑价格）？
答：作选择有下列几点需要考虑 1. 总的采样率：1ms、10 个通道，总采样率为100K ，所有A/D均能满足要求。2. A/D与VC33 的接口类型：并行、串行。前 2 种A/D为并行接口，后几种均为串行接口。3. 接口电平的匹配。前 2 种A/D为 5V电平，与VC33 不能接口；后几种均可为 3.3V电平，可与VC33 直接接口。

问：DSP的电路板有时调试成功率低于 50%，连接和底板均无问题，如何解决？有时DSP同CPLD产生不明原因的冲突，如何避免？
答：看来你的硬件设计可能有问题，不应该这么小的成功率。我们的板的成功率为 95%以上。

问：我们的工程有两人参与开发，由于事先没有考虑周全，一人使用的是助记符方式编写汇编代码，另一人使用的是代数符号方式编写汇编代码，请问CCS5000中这二种编写方式如何嵌在一起调试？
答：我没有这样用过，我想可以用下面的办法解决：将一种方式的程序先单独编译为.obj文件，在创建工程时，将这些.obj文件和另一种方式的程序一起加进工程中，二者即可一起编译调试了。

问：DSP数据缓冲，能否用SDRAM代替FIFO？
答：不行

问：ADC或DAC和DSP相连接时，要注意什么问题？比如匹配问题，以保证A/D采样稳定或D/A码不丢失。
答：1. 接口方式：并行 / 串行；2. 接口电平，必须保证二者一致。

问：用F240 经常发生外部中断丢失现象，甚至在实际环境中只有在程序刚开始时能产生中断，几分钟后就不能产生中断。有时只能采取查询的方式，请问有何有效的解决方法？改为F2407 是不是要好些？
答：应该同DSP无关。建议你将中断服务程序简化看一下。

五、为什么要片内RAM大的DSP效率高？

目前DSP发展的片内存储器RAM越来越大,要设计高效的DSP系统,就应该选择片内RAM较大的DSP。片内RAM同片外存储器相比,有以下优点：

1)片内RAM的速度较快,可以保证DSP无等待运行。
2)对于C2000/C3x/C5000 系列,部分片内存储器可以在一个指令周期内访问两次,使得指令可以更加高效。
3)片内RAM运行稳定,不受外部的干扰影响,也不会干扰外部。
4)DSP片内多总线,在访问片内RAM时,不会影响其它总线的访问,效率较高。

六、为什么DSP从 5V发展成 3.3V？

超大规模集成电路的发展从 1um,发展到目前的 0.1um,芯片的电源电压也随之降低,功耗也随之降低。DSP也同样从 5V发展到目前的 3.3V,核心电压发展到 1V。目前主流的DSP的外围均已发展为 3.3V,5V的DSP的价格和功耗都价格,以逐渐被3.3V的DSP取代。

七、如何选择DSP的电源芯片？

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 750mA;
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。

八、软件等待的如何使用？

DSP的指令周期较快,访问慢速存储器或外设时需加入等待。等待分硬件等待和软件等待,每一个系列的等待不完全相同。
1)对于C2000 系列： 硬件等待信号为READY,高电平时不等待。 软件等待由WSGR寄存器决定,可以加入最多 7 个等待。其中程序存储器和数据存储器及I/O可以分别设置。
2)对于C3x系列： 硬件等待信号为/RDY,低电平是不等待。 软件等待由总线控制寄存器中的SWW和WTCNY决定,可以加入最多 7 个等待,但等待是不分段的,除了片内之外全空间有效。
3)对于C5000 系列： 硬件等待信号为READY,高电平时不等待。 软件等待由SWWCR和SWWSR寄存器决定,可以加入最多 14 个等待。其中程序存储器、控制程序存储器和数据存储器及I/O可以分别设置。
4)对于C6000 系列（只限于非同步存储器或外设）： 硬件等待信号为ARDY,高电

平时不等待。 软件等待由外部存储器接口控制寄存器决定,总线访问外部存储器或设备的时序可以设置,可以方便的同异步的存储器或外设接口。

九、中断向量为什么要重定位？

为了方便DSP存储器的配置,一般DSP的中断向量可以重新定位,即可以通过设置寄存器放在存储器空间的任何地方。

注意：C2000 的中断向量不能重定位。

十、DSP的最高主频能从芯片型号中获得吗？

TI的DSP最高主频可以从芯片的型号中获得,但每一个系列不一定相同。

1)TMS320C2000 系列：

TMS320F206－最高主频 20MHz。

TMS320C203/C206－最高主频 40MHz。

TMS320F24x－最高主频 20MHz。

TMS320LF24xx－最高主频 30MHz。

TMS320LF24xxA－最高主频 40MHz。

TMS320LF28xx－最高主频 150MHz。

2)TMS320C3x系列：

TMS320C30：最高主频 25MHz。

TMS320C31PQL80：最高主频 40MHz。

TMS320C32PCM60：最高主频 30MHz。

TMS320VC33PGE150：最高主频 75MHz。

3)TMS320C5000 系列：

TMS320VC54xx：最高主频 160MHz。

TMS320VC55xx：最高主频 300MHz。

4)TMS320C6000 系列：

TMS320C62xx：最高主频 300MHz。

TMS320C67xx：最高主频 230MHz。

TMS320C64xx：最高主频 720MHz。

十一、DSP可以降频使用吗？

可以,DSP的主频均有一定的工作范围,因此DSP均可以降频使用。

十二、如何选择外部时钟？

DSP的内部指令周期较高,外部晶振的主频不够,因此DSP大多数片内均有PLL。但每个系列不尽相同。

1) TMS320C2000 系列：

TMS320C20x：PLL可以÷2,×1,×2 和×4,因此外部时钟可以为 5MHz－40MHz。

TMS320F240：PLL可以÷2,×1,×1.5,×2,×2.5,×3,×4,×4.5,×5 和×9,因此外部时钟可以为 2.22MHz－40MHz。

TMS320F241/C242/F243：PLL可以×4,因此外部时钟为 5MHz。

TMS320LF24xx：PLL可以由RC调节,因此外部时钟为 4MHz－20MHz。

TMS320LF24xxA：PLL可以由RC调节,因此外部时钟为 4MHz－20MHz。

2) TMS320C3x系列：
TMS320C3x：没有PLL,因此外部主频为工作频率的 2 倍。
TMS320VC33：PLL可以÷2,×1,×5,因此外部主频可以为 12MHz－100MHz。
3)TMS320C5000 系列：
TMS320VC54xx：PLL 可以÷4,÷2,×1-32, 因此外部主频可以为 0.625MHz－50MHz。 TMS320VC55xx：PLL可以÷4,÷2,×1-32,因此外部主频可以为 6.25MHz－300MHz。
4)TMS320C6000 系列： TMS320C62xx：PLL可以×1,×4,×6,×7,×8,×9,×10 和×11, 因此外部主频可以为 11.8MHz－300MHz。
TMS320C67xx：PLL可以×1 和×4,因此外部主频可以为 12.5MHz－230MHz。
TMS320C64xx：PLL可以×1,×6 和×12,因此外部主频可以为 30MHz－720MHz

十三、如何选择DSP的外部存储器？
DSP的速度较快,为了保证DSP的运行速度,外部存储器需要具有一定的速度,否则DSP访问外部存储器时需要加入等待周期。
1)对于C2000 系列： C2000 系列只能同异步的存储器直接相接。 C2000 系列的DSP目前的最高速度为 150MHz。建议可以用的存储器有： CY7C199-15：32K×8,15ns,5V; CY7C1021-12 ： 64K×16,15ns,5V; CY7C1021V33-12 ：64K×16,15ns,3.3V。
2)对于C3x系列： C3x系列只能同异步的存储器直接相接。 C3x系列的DSP的最高速度,5V的为 40MHz,3.3V的为 75MHz,为保证DSP无等待运行,分别需要外部存储器的速度<25ns和<12ns。建议可以用的存储器有： ROM： AM29F400-70：256K×16,70ns,5V, 加入一个等待； AM29LV400-55(SST39VF400) ：256K×16,55ns,3.3V, 加入两个等待（目前没有更快的Flash）。 SRAM：CY7C199-15：32K×8,15ns,5V; CY7C1021-15：64K×16,15ns,5V; CY7C1009-15：128K×8,15ns,5V; CY7C1049-15 ： 512K×8,15ns,5V; CY7C1021V33-15 ： 64K×16,15ns,3.3V; CY7C1009V33-15 ： 128K×8,15ns,3.3V; CY7C1041V33-15 ：256k×16,15ns,3.3V。
3)对于C54x系列： C54x系列只能同异步的存储器直接相接。 C54x系列的DSP的速度为 100MHz或 160MHz,为保证DSP无等待运行,需要外部存储器的速度<10ns或<6ns。建议可以用的存储器有： ROM： AM29LV400-55(SST39VF400)：256K×16,55ns,3.3V,加入 5 或 9 个等待（目前没有更快的Flash）。 SRAM：CY7C1021V33-12 ： 64K×16,12ns,3.3V, 加入一个等待； CY7C1009V33-12 ：128K×8,12ns,3.3V,加入一个等待。
4)对于C55x和C6000 系列： TI的DSP中只有C55x和C6000 可以同同步的存储器相连,同步存储器可以保证系统的数据交换效率更高。 ROM：AM29LV400-55(SST39VF400) ： 256K×16,55ns,3.3V 。 SDRAM ：HY57V651620BTC-10S： 64M,10ns。 SBSRAM： CY7C1329-133AC,64k×32; CY7C1339-133AC,128k×32。 FIFO：CY7C42x5V-10ASC,32k/64k×18。

做DSP之前我们该弄明白那些基本问题 2

十四.DSP芯片有多大的驱动能力？

DSP的驱动能力较强,可以不加驱动,连接 8 个以上标准TTL门。

十五.调试TMS320C2000 系列的常见问题？

1)单步可以运行,连续运行时总回 0 地址： Watchdog没有关,连续运行复位DSP回到 0 地址。

2)OUT文件不能load到片内flash中： Flash不是RAM,不能用简单的写指令写入,需要专门的程序写入。CCS和CSource Debugger中的load命令,不能对flash写入。OUT文件只能load到片内RAM,或片外RAM中。

3)在flash中如何加入断点： 在flash中可以用单步调试,也可以用硬件断点的方法在flash中加入断点,软件断点是不能加在ROM中的。硬件断点,设置存储器的地址,当访问该地址时产生中断。

4)中断向量： C2000 的中断向量不可重定位,因此中断向量必须放在 0 地址开始的flash内。在调试系统时,代码放在RAM中,中断向量也必须放在flash内。

十六.调试TMS320C3x系列的常见问题？

1)TMS320C32 的存储器配置：TMS320C32 的程序存储器可以配置为 16 位或 32 位;数据存储器可以配置为 8 位、16 位或 32 位。

2)TMS320VC33 的PLL控制： TMS320VC33 的PLL控制端只能接 1.8V,不能接 3.3V或 5V。

十七.如何调试多片DSP？

对于有MPSD仿真口的DSP（TMS320C30/C31/C32）,不能用一套仿真器同时调试,每次只能调试其中的一个DSP; 对于有JTAG仿真口的DSP,可以将JTAG串接在一起,用一套仿真器同时调试多个DSP,每个DSP可以用不同的名字,在不同的窗口中调试。

注意：如果在JTAG和DSP间加入驱动,一定要用快速的门电路,不能使用如LS的慢速门电路。

十八.在DSP系统中为什么要使用CPLD？

DSP的速度较快,要求译码的速度也必须较快。利用小规模逻辑器件译码的方式,已不能满足DSP系统的要求。 同时,DSP系统中也经常需要外部快速部件的配合,这些部件往往是专门的电路,有可编程器件实现。 CPLD的时序严格,速度较快,可编程性好,非常适合于实现译码和专门电路。

十九.DSP系统构成的常用芯片有哪些？

1)电源： TPS73HD3xx,TPS7333,TPS56100,PT64xx...

2)Flash： AM29F400,AM29LV400,SST39VF400...

3)SRAM： CY7C1021,CY7C1009,CY7C1049...

4)FIF CY7C425,CY7C42x5...

5)Dual port： CY7C136,CY7C133,CY7C1342...

6)SBSRAM： CY7C1329,CY7C1339...

7)SDRAM： HY57V651620BTC...

8)CPLD： CY37000 系列,CY38000 系列,CY39000 系列...
9)PCI： PCI2040,CY7C09449...
10)USB： AN21xx,CY7C68***...
11)Codec：TLV320AIC23,TLV320AIC10...
12)A/D,D/A：ADS7805,TLV2543... 具体资料www.ti.comwww.cypress.com

二十.什么是boot loader？

DSP的速度尽快,EPROM或flash的速度较慢,而DSP片内的RAM很快,片外的RAM也较快。为了使DSP充分发挥它的能力,必须将程序代码放在RAM中运行。为了方便的将代码从ROM中搬到RAM中,在不带flash的DSP中,TI在出厂
时固化了一段程序,在上电后完成从ROM或外设将代码搬到用户指定的RAM中。此段程序称为"boot loader"。

二十一.TMS320C3x如何boot？

在MC/MP管脚为高时,C3x进入boot状态。C3x的boot loader在reset时,判断外部中断管脚的电平。根据中断配置决定boot的方式为存储器加载还是串口加载,其中ROM的地址可以为三个中的一个,ROM可以为 8 位。

二十二.Boot有问题如何解决？

1)仔细检查boot的控制字是否正确。
2)仔细检查外部管脚设置是否正确。
3)仔细检查hex文件是否转换正确。
4)用仿真器跟踪boot过程,分析错误原因。

二十三.DSP为什么要初始化？

DSP在RESET后,许多的寄存器的初值一般同用户的要求不一致,例如：等待寄存器,SP,中断定位寄存器等,需要通过初始化程序设置为用户要求的数值。
初始化程序的主要作用：
1)设置寄存器初值。
2)建立中断向量表。
3)外围部件初始化。

二十四.DSP有哪些数学库及其它应用软件？

TI公司为了方便客户开发DSP,在它的网站上提供了许多程序的示例和应用程序,如MATH库,FFT,FIR/IIR等,可以在TI的网页免费下载。

二十五.如何获得DSP专用算法？

TI有许多的Third Party可以通过DSP上的多种算法软件。可以通过TI的网页搜索你所需的算法,找到通过算法的公司,同相应的公司联系。注意这些算法都是要付费的。

二十六.eXpressDSP是什么？

eXpressDSP是一种实时DSP软件技术,它是一种DSP编程的标准,利用它可以加快

你开发DSP软件的速度。 以往DSP软件的开发没有任何标准,不同的人写的程序一般无法连接在一起。DSP软件的调试工具也非常不方便。使得DSP软件的开发往往滞后于硬件的开发。 eXpressDSP集成了CCS(Code Composer Studio)开发平台,DSP BIOS实时软件平台,DSP算法标准和第三方支持四部分。利用该技术,可以使你的软件调试,软件进程管理,软件的互通及算法的获得,都便的容易。这样就可以加快你的软件开发进程。

1)CCS是eXpressDSP的基础,因此你必须首先拥有CCS软件。

2)DSP BIOS是eXpressDSP的基本平台,你必须学会所有DSP BIOS。

3)DSP算法标准可以保证你的程序可以方便的同其它利用eXpressDSP技术的程序连接在一起。同时也保证你的程序的延续性。

二十七.为什么要用DSP?

3G技术和internate的发展,要求处理器的速度越来越高,体积越来越小,DSP的发展正好能满足这一发展的要求。因为,传统的其它处理器都有不同的缺陷。MCU的速度较慢;CPU体积较大,功耗较高;嵌入CPU的成本较高。DSP的发展,使得在许多速度要求较高,算法较复杂的场合,取代MCU或其它处理器,而成本有可能更低。

二十八.如何选择DSP?

选择DSP可以根据以下几方面决定:

1)速度： DSP速度一般用MIPS或FLOPS表示,即百万次/秒钟。根据您对处理速度的要求选择适合的器件。一般选择处理速度不要过高,速度高的DSP,系统实现也较困难。

2)精度： DSP芯片分为定点、浮点处理器,对于运算精度要求很高的处理,可选择浮点处理器。定点处理器也可完成浮点运算,但精度和速度会有影响。

3)寻址空间： 不同系列DSP程序、数据、I/O空间大小不一,与普通MCU不同,DSP在一个指令周期内能完成多个操作,所以DSP的指令效率很高,程序空间一般不会有问题,关键是数据空间是否满足。数据空间的大小可以通过DMA的帮助,借助程序空间扩大。

4)成本： 一般定点DSP的成本会比浮点DSP的要低,速度也较快。要获得低成本的DSP系统,尽量用定点算法,用定点DSP。

5)实现方便： 浮点DSP的结构实现DSP系统较容易,不用考虑寻址空间的问题,指令对C语言支持的效率也较高。

6)内部部件：根据应用要求,选择具有特殊部件的DSP。如：C2000 适合于电机控制;OMAP适合于多媒体等。

二十九.DSP同MCU相比的特点?

1)DSP的速度比MCU快,主频较高。

2)DSP适合于数据处理,数据处理的指令效率较高。

3)DSP均为 16 位以上的处理器,不适合于低档的场合。

4)DSP可以同时处理的事件较多,系统级成本有可能较低。

5)DSP的灵活性较好,大多数算法都可以软件实现。

6)DSP的集成度较高,可靠性较好。

三十.DSP同嵌入CPU相比的特点？

1)DSP是单片机,构成系统简单。
2)DSP的速度快。
3)DSP的成本较低。
4)DSP的性能高,可以处理较多的任务。

三十一.如何编写C2000 片内Flash？

DSP中的Flash的编写方法有三中：
1.通过仿真器编写：在我们的网页上有相关的软件,在销售仿真器时我们也提供相关软件。其中LF240x的编写可以在CCS中加入一个插件,F24x的编写需要在windows98 下的DOS窗中进行。具体步骤见软件中的readme。有几点需要注意：
a.必须为MC方式;
b.F206 的工作频率必须为 20MHz;
c.F240 需要根据PLL修改C240_CFG.I文件。建议外部时钟为 20MHz;
d.LF240x也需要根据PLL修改文件;
e.如果编写有问题,可以用BFLWx.BAT修复。
2.提供串口编写：TI的网页上有相关软件。注意只能编写一次,因为编写程序会破坏串口通信程序。
3.在你的程序中编写：TI的网页上有相关资料。

做DSP之前我们该弄明白那些基本问题 3

三十二.如何编写DSP外部的Flash？

DSP的外部Flash编写方法：
1.通过编程器编写：将OUT文件通过HEX转换程序转换为编程器可以接受的格式,再由编程器编写。
2.通过DSP软件编写：您需要根据Flash的说明,编写Flash的编写程序,将应用程序和编写Flash的程序分别load到RAM中,运行编写程序编写。

三十三.对于C5000,大于 48K的程序如何BOOT？

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的程序不能对于 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。

三十四.DSP外接存储器的控制方式？

对于一般的存储器具有RD、WR和CS等控制信号,许多DSP（C3x、C5000）都没

有控制信号直接连接存储器,一般采用的方式如下：
1.CS有地址线和PS、DS或STRB译码产生;
2./RD=/STRB+/R/W;
3./WR=/STRB+R/W。

三十五.GEL文件的功能？

GEL文件的功能同emuinit.cmd的功能基本相同,用于初始化DSP。但它的功能比emuinit的功能有所增强,GEL在CCS下有一个菜单,可以根据DSP的对象不同,设置不同的初始化程序。
以TMS320LF2407 为例：

```
#define SCSR1 0x7018 ;定义scsr1 寄存器
#define SCSR2 0X7019 ;定义scsr2 寄存器
#define WDKEY 0x7025 ;定义wdkey寄存器
#define WDNTR 0x7029 ;定义wdntr寄存器
StartUp() ; 开始函数
{ GEL_MapReset(); 存储空间复位
GEL_MapAdd(0x0000,0,0x7fff,1,1); 定义程序空间从 0000－7fff 可读写
GEL_MapAdd(0x8000,0,0x7000,1,1); 定义程序空间从 8000－f000 可读写
GEL_MapAdd(0x0000,1,0x10000,1,1); 定义数据空间从 0000－10000 可读写
GEL_MapAdd(0xffff,2,1,1,1); 定义i/o 空间 0xffff可读写
GEL_MapOn(); 存储空间打开
GEL_MemoryFill(0xffff,2,1,0x40); 在i/o空间添入数值
40h *(int *)SCSR1=0x0200; 给scsr1 寄存器赋值
*(int *)SCSR2=0x000C; 给scsr2 寄存器赋值,在这里可以进行mp/mc方式的转换
*(int *)WDNTR=0x006f; 给wdntr寄存器赋值
*(int *)WDKEY=0x055; 给wdkey寄存器赋值
*(int *)WDKEY=0x0AA; 给wdkey寄存器赋值 }
```

三十六.使用TI公司模拟器件与DSP结合使用的好处。

1)在使用TI公司的DSP的同时,使用TI公司的模拟可以和DSP进行无缝连接。器件与器件之间不需要任何的连接或转接器件。这样即减少了板卡的尺寸,也降低了开发难度。
2)同为TI公司的产品,很多器件可以固定搭配使用。少了器件选型的烦恼
3)TI在CCS中提供插件,可以用于DSP和模拟器件的开发,非常方便。

三十七.C语言中可以嵌套汇编语言？

可以。在ANSI C标准中的标准用法就是用C语言编写主程序,用汇编语言编写子程序,中断服务程序,一些算法,然后用C语言调用这些汇编程序,这样效率会相对比较高

三十八.在定点DSP系统中可否实现浮点运算？

当然可以,因为DSP都可以用C,只要是可以使用c语言的场合都可以实现浮点运算。

三十九.JTAG头的使用会遇到哪些情况？

1)DSP的CLKOUT没有输出,工作不正常。
2)Emu0,Emu1 需要上拉。
3)TCK的频率应该为 10M。
4)在 3.3V DSP中,PD脚为 3.3V 供电,但是仿真器上需要 5V电压供电,所以PP仿真器盒上需要单独供电。
5)仿真多片DSP。在使用菊花链的时候,第一片DSP的TDO接到第二片DSP的TDI即可。注意当串联DSP比较多的时候,信号线要适当的增加驱动。

四十.include头文件

（.h)的主要作用头文件,一般用于定义程序中的函数、参数、变量和一些宏单元,同库函数配合使用。因此,在使用库时,必须用相应的头文件说明。

四十一.DSP中断向量的位置

1)2000 系列dsp的中断向量只能从 0000H处开始。所以在我们调试程序的时候,要把DSP选择为MP（微处理器方式）,把片内的Flash屏蔽掉,免去每次更改程序都要重新烧写Flash工作。
2)3x系列dsp的中断向量也只能在固定的地址。
3)5000,6000 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。

四十二.有源晶振与晶体的区别,应用范围及用法

1)晶体需要用DSP片内的振荡器,在datasheet上有建议的连接方法。晶体没有电压的问题,可以适应于任何DSP,建议用晶体。 2)有源晶振不需要DSP的内部振荡器,信号比较稳定。有源晶振用法：一脚悬空,二脚接地,三脚接输出,四脚接电压。

四十三.程序经常跑飞的原因

1)程序没有结尾或不是循环的程序。
2)nmi管脚没有上拉。
3)在看门狗动作的时候程序会经常跑飞。
4)程序编制不当也会引起程序跑飞。
5)硬件系统有问题。

四十四.并行FLASH引导的一点经验

最近BBS上关于FLASH和BOOT的讨论很活跃,我也多次来此请教。前几天自制的DSP板引导成功,早就打算写写这方面的东西。我用的DSP是 5416,以其为核心,做了一个相对独立的子系统（硬件、软件、算法）,目前都已基本做好。 下面把在FLASH引导方面做的工作向大家汇报一下,希望能对大家有所帮助。本人经验和文笔都有限,写的不好请大家谅解。

硬件环境：
DSP：TMS320VC5416PGE160

FLASH：SST39VF400A-70-4C-EK 都是贴片的,FLASH映射在DSP数据空间的0x8000-0xFFFF

软件环境：
CCS v2.12.01 主程序（要烧入FLASH的程序）： DEBUG版,程序占用空间0x28000-0x2FFFF（片内SARAM）,中断向量表在 0x0080-0x00FF（片内DARAM）,数据空间使用 0x0100-0x7FFF（片内DARAM）。 因为FLASH是贴片的,所以需要自己编一个数据搬移程序,把要主程序搬移到FLASH中。在写入FLASH数据时,还应写入引导表的格式数据。最后在数据空间的 0xFFFF处写入引导表的起始地址（这里为 0x8000）。
搬移程序： DEBUG版,程序空间 0x38000-0x3FFFF（片内SARAM）,中断向量表在 0x7800-0x78FF（片内DARAM）,数据空间使用 0x5000-0x77FF（片内DARAM）。搬移程序不能使用与主程序的程序空间和中断向量表重合的物理空间,以免覆盖。烧写时,同时打开主程序和搬移程序的PROJECT,先LOAD主程序,再LOAD搬移程序,然后执行搬移程序,烧写OK!
附：搬移程序（仅供参考）

```
volatile unsigned int *pTemp=(unsigned int *)0x7e00; unsigned int iFlashAddr; int iLoop; /* 在
引导表头存放并行引导关键字 */ iFlashAddr=0x8000; WriteFlash(iFlashAddr,0x10aa); iFlashAddr++;
/* 初始化SWWSR值 */ WriteFlash(iFlashAddr,0x7e00); iFlashAddr++; /* 初始化BSCR值 */ WriteFlash
(iFlashAddr,0x8006); iFlashAddr++; /* 程序执行的入口地址 */ WriteFlash(iFlashAddr,0x0002);
iFlashAddr++; WriteFlash(iFlashAddr,0x8085); iFlashAddr++; /* 程序长度 */ WriteFlash
(iFlashAddr,0x7f00); iFlashAddr++; /* 程序要装载到的地址 */ WriteFlash(iFlashAddr,0x0002);
iFlashAddr++; WriteFlash(iFlashAddr,0x8000); iFlashAddr++; for (iLoop=0;iLoop<0x7f00;iLoop++) {
/* 从程序空间读数据,放到暂存单元 */ asm(" pshm al"); asm(" pshm ah"); asm(" rsbx cpl"); asm("
ld #00fch,dp"); asm(" stm #0000h, ah"); asm(" MVDM _iLoop, al"); asm(" add #2800h,4,a"); asm("
reada 0h"); asm(" popm ah"); asm(" popm al"); asm(" ssbx cpl"); /* 把暂存单元内容写入FLASH */
WriteFlash(iFlashAddr,*pTemp); iFlashAddr++; } /* 中断向量表长度 */ WriteFlash
(iFlashAddr,0x0080); iFlashAddr++; /* 中断向量表装载地址 */ WriteFlash(iFlashAddr,0x0000);
iFlashAddrlash(iFlashAddr,0x0080); iFlashAddr++; for (iLoop=0;iLoop<0x0080;iLoop++) { /* 从程序
```

空间读数据,放到暂存单元 */ asm(" pshm al"); asm(" pshm ah"); asm(" rsbx cpl"); asm(" ld
#00fch,dp"); asm(" stm #0000h, ah"); asm(" MVDM _iLoop, al"); asm(" add #0080h,0,a"); asm("
reada 0h"); asm(" popm ah"); asm(" popm al"); asm(" ssbx cpl"); /* 把暂存单元内容写入FLASH */
WriteFlash(iFlashAddr,*pTemp); iFlashAddr++; } /* 写入引导表结束标志 */ WriteFlash
(iFlashAddr,0x0000); iFlashAddr++; WriteFlash(iFlashAddr,0x0000); /* 在数据空间的 0xFFFF写入引导
表起始地址 */ iFlashAddr=0xffff; WriteFlash(iFlashAddr,0x8000);

四十五.关于LF2407A的FLASH烧写问题的几点说明

TI现在关于LF24x写入FLASH的工具最新为c2000flashprogsw_v112。可以支持LF2407、LF2407a、LF2401 及相关的LF240x系列。建议使用此版本。在 http://focus.ti.com/docs/tool/to ... Number=C24XSOFTWARE上可以下载到这个工具。我们仿真器自带的光盘中也有此烧写程序。 在使用这个工具时注意：

一,先解压，再执行setup.exe。

二、进入cc中，在tools图标下有烧写工具；

1、关于FLASH时钟的选择，此烧写工具默认最高频率进行FLASH的操作。根据目标系统的工作主频重新要进行PLL设置。方法：先在advance options下面的View Config file中修改倍频。存盘后，在相应的目录下（tic2xx\\algos\\相应目录）运行buildall.bat就可以完成修改了。再进行相应的操作即可。

2、若是你所选的频率不是最高频率，还需要设定你自己的timings.xx来代替系统默认的最高频率的timings.xx。例如LF2407a的默认文件是timings.40。Timings.xx可以利用include\\timings.xls的excel工作表来生成。然后在advance options下面的View Config file中修改相应的位置。存盘后，在相应的目录下运行buildall.bat就可以完成修改了。

3、对于TMS320LF240XA系列，还要注意：由于这些DSP的FLASH具有加密功能，加密地址为程序空间的 0x40-0X43H，程序禁止写入此空间，如果写了，此空间的数据被认为是加密位，断电后进入保护FLASH状态，使FLASH不可重新操作，从而使DSP报废，烧写完毕后一定要进行Program passwords的操作，如果不做加密操作就默认最后一次写入加密位的数据作为密码。

4、2407A不能用DOS下的烧写软件烧写，必须用c2000flashprogsw_v112 软件烧写；

5、建议如下：

1)、一般调试时，在RAM中进行；

2)、程序烧写时，避开程序空间 0x40-0x43H加密区，程序最好小于 32k；

3)、每次程序烧写完后，将word0，word1,word2，word3 分别输入自己的密码，再点击 Program password，如果加密成功，提示Program is arrayed，如果 0x40－0x43h中写入的是ffff，认为处于调试状态，flash不会加密；

4)、断电后，下次重新烧写时需要往word0～word3 输入已设的密码，再unlock，成功后可以重新烧写了；

6、VCPP管脚接在＋5V上，是应直接接的，中间不要加电阻。

7、具体事宜请阅读相应目录下的readme1,readme2 帮助文件。

8.注意*.cmd文件的编写时应该避开 40-43H单元，好多客户由于没有注意到这里而把FALSH加密。

四十六.如何设置硬件断点？

在profiler －>profile point -> break point

四十七.c54x的外部中断是电平响应还是沿响应？

是沿响应，准确的说，它要检测到 100(一个clk的高和两个clk的低)的变化才可以。

四十八.参考程序，里面好象都要 disable wachdog,不知道为什么？

watchdog是一个计数器，溢出时会复位你的DSP，不disable的话，你的系统会动不动就reset。

四十九.时钟电路选择原则

1,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;

2,单一时钟信号时，选择晶体时钟电路;

3,多个同频时钟信号时，选择晶振;

4,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;

5,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;

6,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

五十.C程序的代码和数据如何定位

1,系统定义:

.cinit 存放C程序中的变量初值和常量;
.const 存放C程序中的字符常量、浮点常量和用const声明的常量;
tch 存放C程序tch语句的跳针表;
.text 存放C程序的代码;
.bss 为C程序中的全局和静态变量保留存储空间;
.far 为C程序中用far声明的全局和静态变量保留空间;
.stack 为C程序系统堆栈保留存储空间，用于保存返回地址、函数间的参数传递、存储局部变量和保存中间结果;
.sysmem 用于C程序中malloc、calloc和realloc函数动态分配存储空间
2,用户定义:
#pragma CODE_SECTION (symbol, "section name");
#pragma DATA_SECTION (symbol, "section name")

五十一.cmd文件

由 3 部分组成：
1)输入 / 输出定义：.obj文件：链接器要链接的目标文件;.lib文件：链接器要链接的库文件;.map文件：链接器生成的交叉索引文件;.out文件：链接器生成的可执行代码;链接器选项
2)MEMORY命令：描述系统实际的硬件资源
3)SECTIONS命令：描述"段"如何定位

五十二.为什么要设计CSL?

1,DSP片上外设种类及其应用日趋复杂
2,提供一组标准的方法用于访问和控制片上外设
3,免除用户编写配置和控制片上外设所必需的定义和代码

五十三.什么是CSL?

1,用于配置、控制和管理DSP片上外设
2,已为C6000 和C5000 系列DSP设计了各自的CSL库
3,CSL库函数大多数是用C语言编写的，并已对代码的大小和速度进行了优化
4,CSL库是可裁剪的：即只有被使用的CSL模块才会包含进应用程序中
5,CSL库是可扩展的：每个片上外设的API相互独立，增加新的API，对其他片上

外设没有影响

五十四.CSL的特点

1,片上外设编程的标准协议：定义一组标准的APIs：函数、数据类型、宏;

2,对硬件进行抽象，提取符号化的片上外设描述:定义一组宏，用于访问和建立寄存器及其域值

3,基本的资源管理:对多资源的片上外设进行管理;

4,已集成到DSP/BIOS中:通过图形用户接口GUI对CSL进行配置;

5,使片上外设容易使用:缩短开发时间，增加可移植.

五十五.为什么需要电平变换?

1)DSP系统中难免存在 5V/3.3V混合供电现象;

2)I/O为 3.3V供电的DSP，其输入信号电平不允许超过电源电压 3.3V;

3)5V器件输出信号高电平可达 4.4V;

4)长时间超常工作会损坏DSP器件;

5)输出信号电平一般无需变换

五十六.电平变换的方法

1,总线收发器（Bus Transceiver）:

常用器件： SN74LVTH245A（8 位）、SN74LVTH16245A（16 位）

特点：3.3V供电，需进行方向控制，

延迟：3.5ns，驱动：-32/64mA，

输入容限：5V

应用：数据、地址和控制总线的驱动

2,总线开关（Bustch）

常用器件：SN74CBTD3384（10 位）、SN74CBTD16210（20 位）

特点：5V供电，无需方向控制

延迟：0.25ns，驱动能力不增加

应用：适用于信号方向灵活、且负载单一的应用，如McBSP等外设信号的电平变换

3,2 选 1 切换器（1 of 2 Multiplexer）

常用器件：SN74CBT3257（4 位）、SN74CBT16292（12 位）

特点：实现 2 选 1，5V供电，无需方向控制

延迟：0.25ns，驱动能力不增加

应用：适用于多路切换信号、且要进行电平变换的应用，如双路复用的McBSP

4,CPLD

3.3V供电，但输入容限为 5V，并且延迟较大：＞7ns，适用于少量的对延迟要求不高的输入信号

5,电阻分压

10KΩ和 20KΩ串联分压，5V×20÷（10＋20）≈3.3V

五十七.未用的输入／输出引脚的处理

1,未用的输入引脚不能悬空不接，而应将它们上拉活下拉为固定的电平

1)关键的控制输入引脚，如Ready、Hold等，应固定接为适当的状态,Ready引脚应固定接为有效状态,Hold引脚应固定接为无效状态

2)无连接（NC）和保留（RSV）引脚,NC 引脚：除非特殊说明，这些引脚悬空不接,RSV引脚：应根据数据手册具体决定接还是不接

3)非关键的输入引脚,将它们上拉或下拉为固定的电平，以降低功耗

2,未用的输出引脚可以悬空不接

3,未用的I/O引脚:如果确省状态为输入引脚，则作为非关键的输入引脚处理，上拉或下拉为固定的电平;如果确省状态为输出引脚，则可以悬空不接。