



基于C6000的应用开发

- DSP历史：特点
- DSP学习：基础知识
- 数据链路的层次
- 中断、DMA和时钟
- C6000的开发环境：CCS
- C6000的软件开发：
CPU结构、汇编、C、线性汇编
C开发环境、优化
传统的DSP软件开发模式：中断驱动
- C6000的实时操作系统：DSP/BIOS



DSP的历史

- DSP历史:

实时系统对数据处理的要求促进DSP的出现和发展;

70年代末, 第一片DSP出现, Intel2920 , 然后是Upd7720 。

第一代DSP的标志是**TMS32010**, 其它代表还有AMD2900、NEC7720。

- 80年代末, DSP开始高速发展, DSP器件内部使用流水线, 并行指令和多核结构



DSP的特点

- DSP的目的和核心：大量的数据处理。带来多处理单元和多数据链路。
其CPU结构的特点一般有
- 1. 运算能力强，在单指令周期类完成乘加运算。
(靠并行实现)
- 2. 采用哈佛结构和流水线技术。
- 3. 芯片具有满足数字信号算法特殊要求的寻址方式。
- 4. 数据交换能力高。
- 5. 多处理单元，支持并行处理指令等。



DSP的重要指标

- IO数据率
- 处理能力
- 片内存储容量, cache
- Benchmark: 1024点FFT的时间
C30,1.5ms, C6701,120us, C6416,10us
- 其它片上资源:
中断, DMA, timer, 串口
- IO接口: memory接口, 总线接口



DSP 的学习

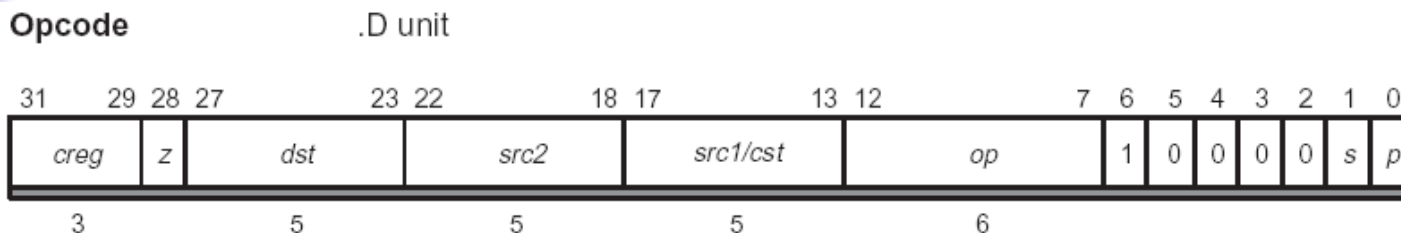
- 基础知识：
EE：微机原理； CS：计算机组成原理
- 几个概念：
CPU结构：处理核，数据通路（寻址），
控制，资源
寄存器：通用寄存器，控制寄存器，
状态寄存器
指令和机器码：
寻址：CPU访问DRAM, 获取数据的手段
流水线：
- C6000学习的重点和难点：优化，并行，流水



一条C6000的指令和其机器码

- C6000: ADD .D2 B5,B4,B4 ADD (.D2 or v.D1) src2,src1,src1
- 00000010000101001000100001000010
- 000 0 00100 00101 00100 010000 10000 1 0
- (1) (2) (3) (4) (5) (6) (7) (8) (9)
- (1) 条件寄存器
- (2) z, 指定条件寄存器的判断条件
- (3) dst, 目的
- (4) src2, 源2
- (5) src1, 源1
- (6) 操作码: 设定唯一指令的码, sint, 2个源和目标都为有符号整数且功能单元为D时的操作码就是010000 ;
- (7) 固定值
- (8) s, 选择A边寄存器还是B边寄存器
- (9) p, 是否并行

手册上对ADD .D指令的讲解



Description for .D1, .D2 Opcodes

src1 is added to *src2*. The result is placed in *dst*.

Execution for .D1, .D2 Opcodes

if (cond) *src2* + *src1* → *dst*
 else nop

Pipeline

Pipeline Stage	E1
Read	<i>src1, src2</i>
Written	<i>dst</i>
Unit in use	.L, .S, or .D

Instruction Type

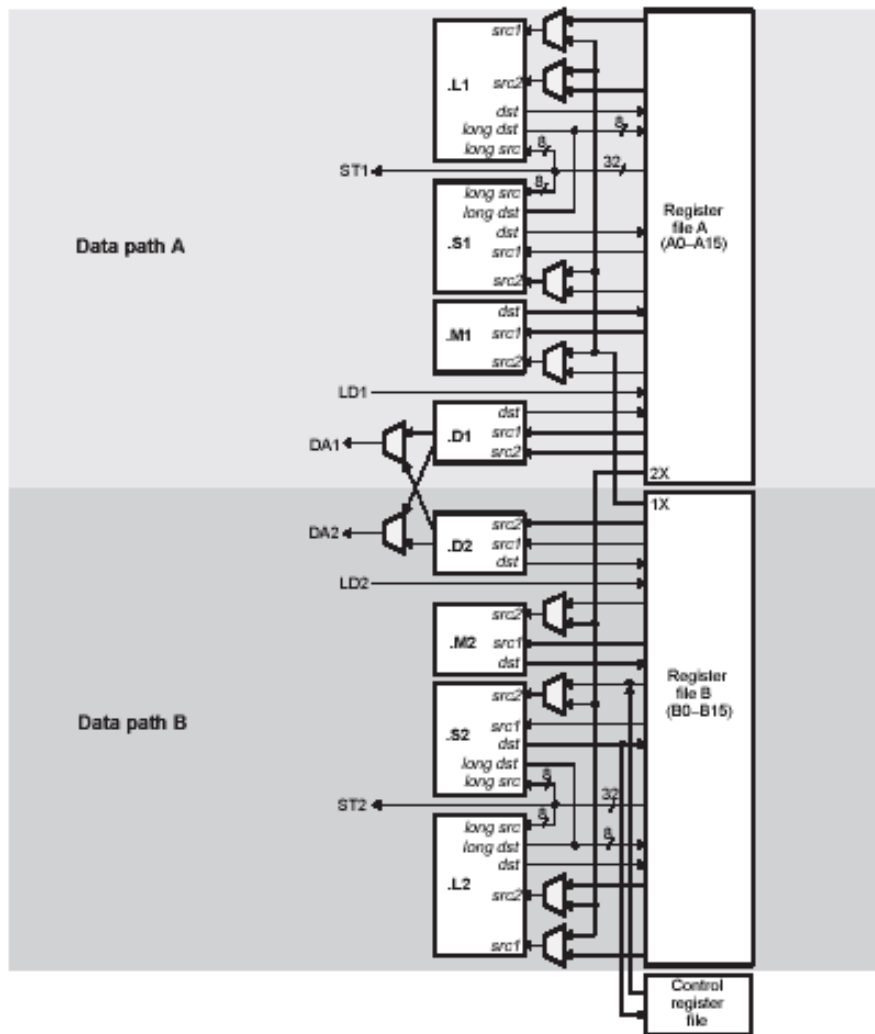
Single-cycle

Delay Slots

0

数据链路层次 (1) CPU核

Figure 2-1. TMS320C62x CPU Data Paths

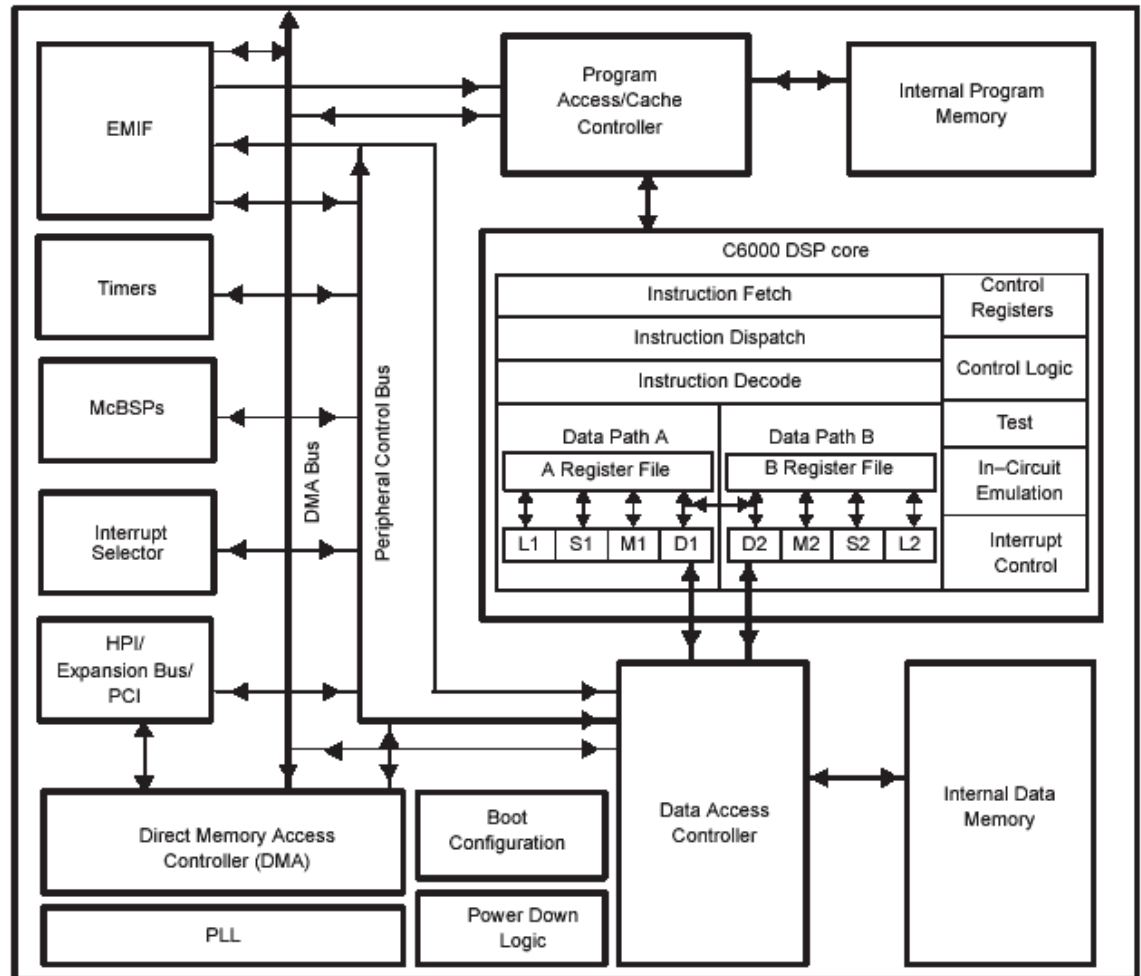


- DSP的寄存器和运算单元之间的数据链路多

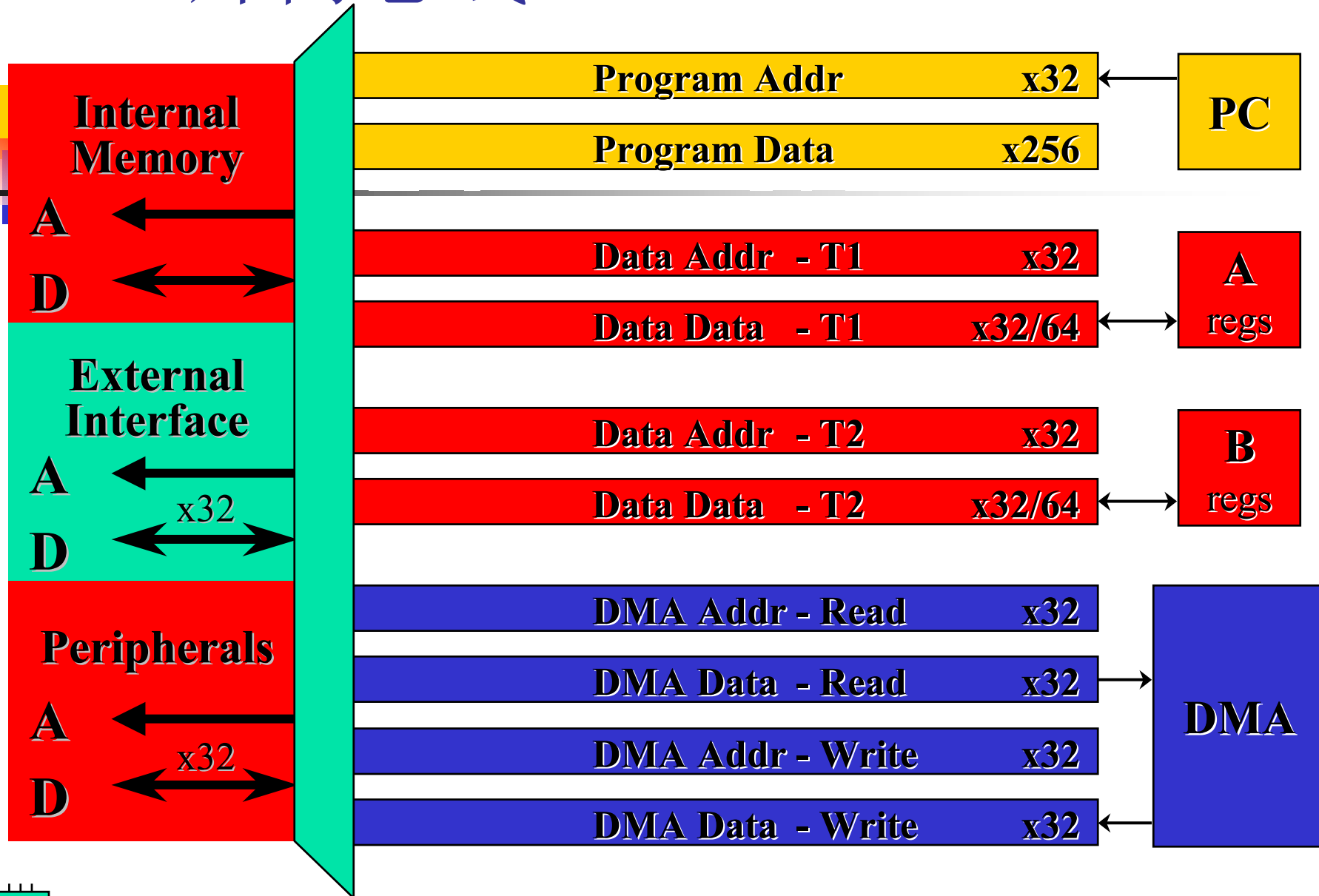
数据链路层次 (2) DSP

Figure 1-1. TMS320C620x/C670x Block Diagram

- 片内存储器
- 片外存储器接口:
EMIF, HPI/XBUS
- DMA
- 中断
- 其它外设:
时钟, 串口
- 补: PLL时钟倍频
BOOT模式, 低功耗模式



C6x 片内总线



'C67x can perform 64-bit data loads.

数据链路层次 (3) 板级

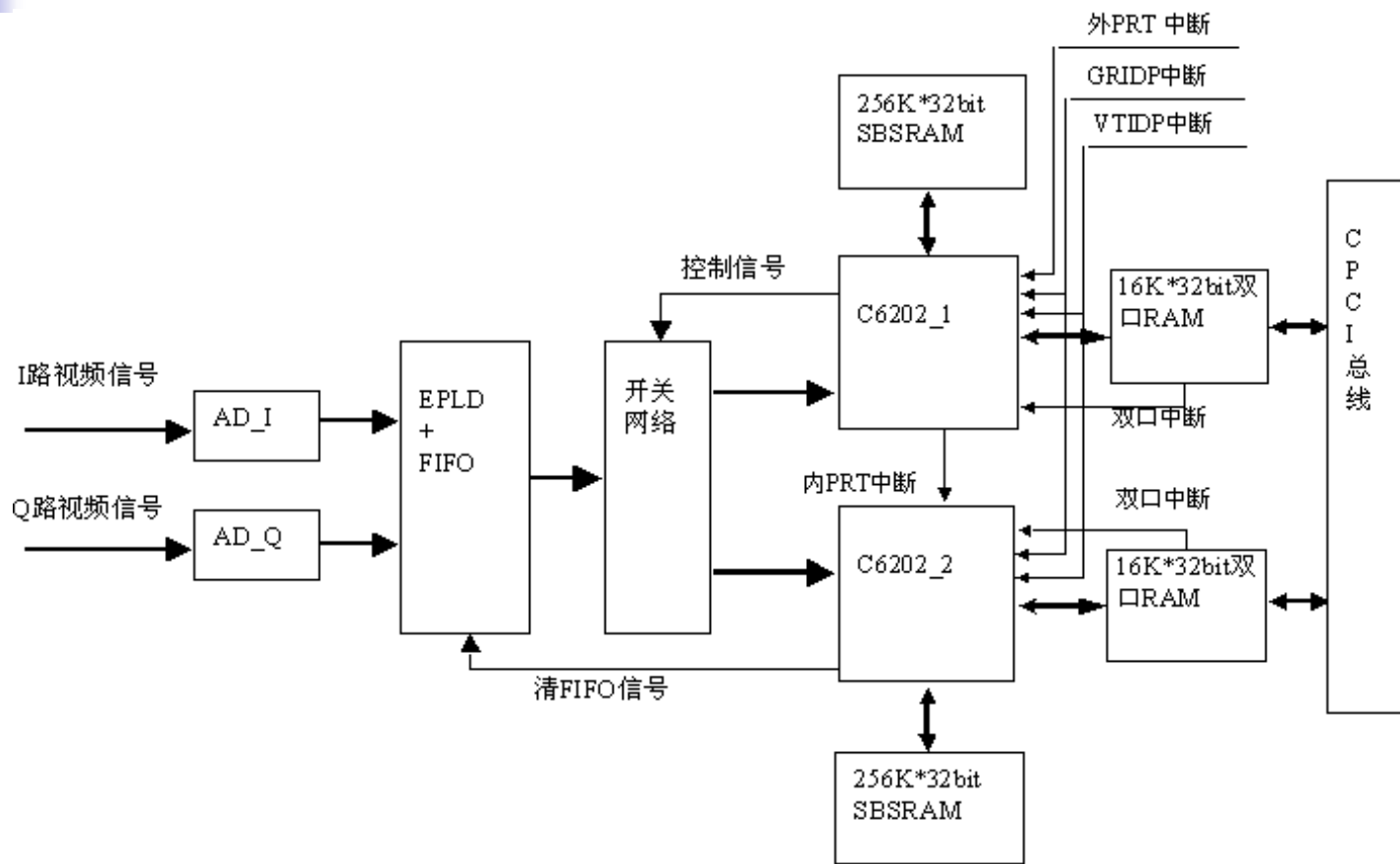


图1 AD板硬件简要框图

数据链路层次 (4) 板间

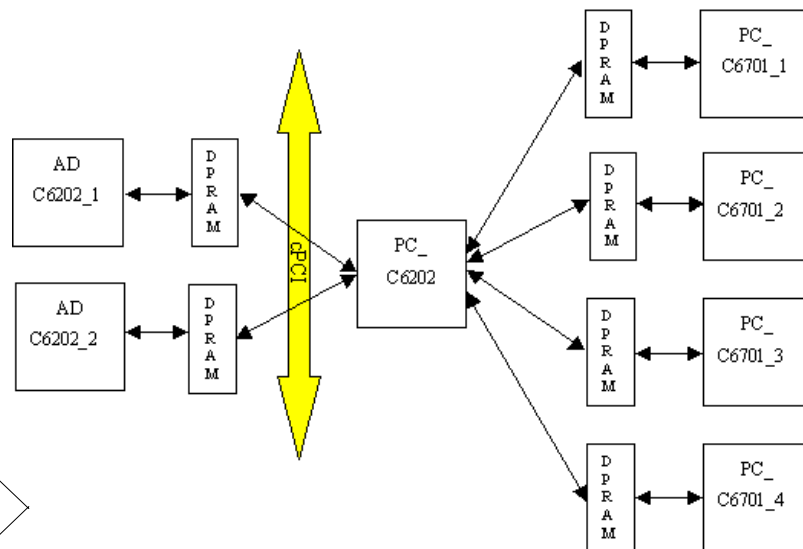
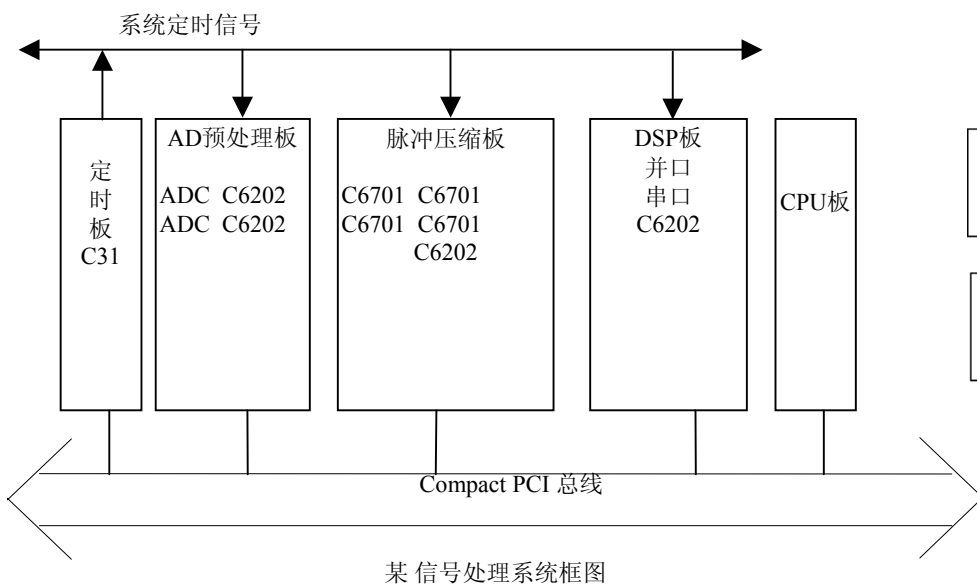


图 5.19 用作数据调度的 PC_C6202 和其他 C6000 DSPs 的拓扑结构

数据链路层次 (5) 系统间

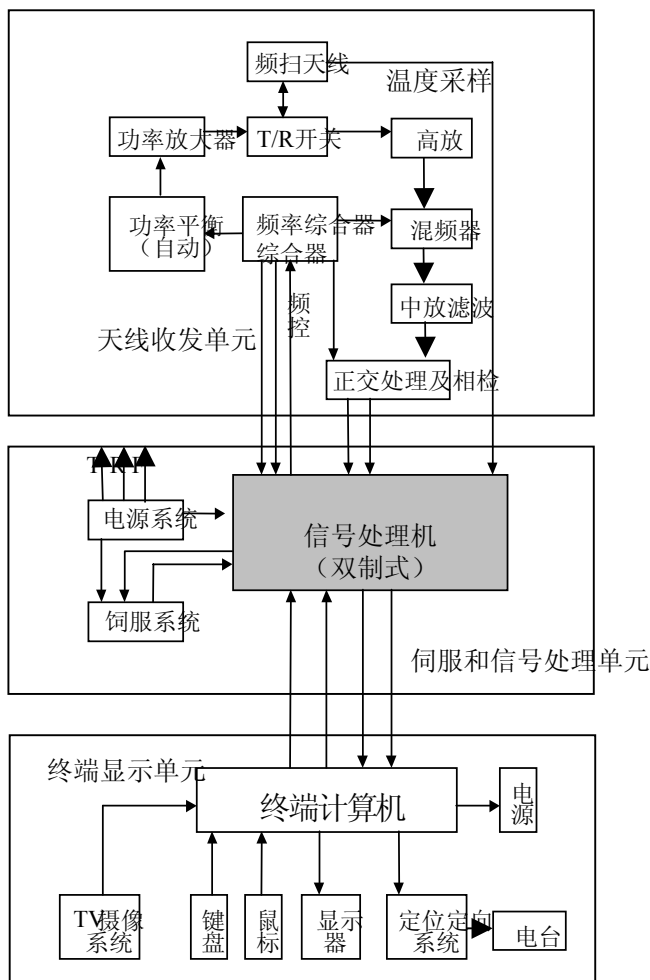


图5.1 某雷达系统原理框图

- 其实是一个雷达系统的子系统之间的数据链路。
- 再进一步：这个雷达系统可以组网，利用通信系统给决策单位提供信息，进行后数据处理，做出决策
- 总之，要对数据链路有一个清晰的各级层次的传输图



数据IO技术的历史与趋势

- 专用总线:

优点: 针对特别设计优化的, 性能比较高;

缺点: 专用设计的费用高; 功能固定, 难于升级; 由于ASIC设计的复杂性使产品推向市场的时间比较长

- 标准总线

优点: 工业标准, 可以得到很多厂商的支持

缺点: 采用共享总线, 每个设备共用总线带宽; 有总线长度的物理限制, 大约1英尺

- 基于开关的、点对点的互联

基于LVDS技术;

RapidIO: $100\text{M} \times 8\text{对线 (bit)} = 800\text{Mbyte/s}$



DSP系统程序的分类

- 简单地，可以分为
 - (1)数据处理的程序：
各种算法的实现；与硬件相对无关
 - (2)数据传输的程序：与硬件紧密相关；
中断、**DMA**：提高数据传输效率的
两种古老而有效的手段
- 两种程序的开发时间：



C6000的中断（1）

- 三个概念：
 - (1)中断源（Interrupt source）：即中断事件，32个
 - (2)中断选择号（Interrupt Selection Number）：5bit
 - (3)中断(Interrupt)：12个可屏蔽+NMI（No.1）
+RESET（No.0）+ RTDX（No.3）
- 中断寄存器的设置：

中断源通过中断选择子（Interrupt Selector）“映射”到CPU的中断上，并可通过外中断极性设置寄存器设置EXT_INT4→EXT_INT7的中断极性（即中断地触发时刻）



C6000的中断（2）

- 中断服务程序：
map1: IDRAM 0x000~0x200
0x200 = 16 * 32 ; 16个中断，每个中断的8条指令；
- 二次跳转：
中断服务程序的任务实现超过8条指令时，跳转到真正处理中断的子程序

```
_INT4: .asg 4, vec
      •      stw .d2 b0, *--b15
      •      || mvk      (_isr_jump_table + vec * 4), b0
      •      mvkh     (_isr_jump_table + vec * 4), b0
      •      ldw .d2 *b0, b0
      •      nop      4
      •      b .s2 b0
      •      || ldw .d2 *b15++, b0
      •      nop      5
```



C6000的中断（3）

- 二次跳转的实现：
 - (1) 中断跳转表：`isr_jump_table` 全局变量
 - (2) “挂”中断：

```
void intr_hook(void (*fp)(void),int  
cpu_intr);
```
- 来源：

ti\c6000\evm6x\dsp\... 库函数，头文件，源文件

v1.2 有； v2.0 只含头文件



C6000的中断（4）

■ 中断嵌套

- 进入中断后，**GIE=0**
- 允许中断嵌套的设置

■ 中断重入

- 打乱时序关系，应该尽量避免

- ```
IRPBak = GET_REG(IRP);
CSRBak = GET_REG(CSR);
IERBak = GET_REG(IER);
INTR_DISABLE(CPU_INT6);
INTR_GLOBAL_ENABLE();
.....
INTR_CLR_FLAG(CPU_INT6);
INTR_GLOBAL_DISABLE();
SET_REG(IRP, IRPBak);
SET_REG(CSR, CSRBak);
SET_REG(IER, IERBak);
```



# 写C6000中断程序

- 嵌入式系统是中断驱动的系统  
典型的结构：**main**函数中初始化→**While**循环等待中断到来，在中断处理程序中进行真正的数据处理
- 中断之间的关系：
  - 同步关系：中断的触发次数成比例
  - 异步关系：。。。
  - 因果关系：一个中断触发其他中断，可看作是同步关系中的一种特例
- 基于以上关系的中断间的数据共享模式：
  - 乒乓模式：采数中断+处理中断，同步中断关系
  - 圆周**buffer**：串口中断+处理中断，异步中断关系
  - 共享**memory**：传数中断+**DMA**结束中断，因果关系



# C6000的DMA（1）

- C6000的DMA控制器在CPU结构中的位置，确定DMA的数据链路：编程时考虑效率  
XB, PBC, EMIF, DMC, PMC  
\*其中和PMC的连接是单向的，只能向程序存储空间写入
- 4个DMA通道+1个辅助DMA（HPI/XB）
- 复杂的传输模式：满足不同的数据传输应用  
单通道分割，多帧传输，自动初始化，传输完毕后触发中断.....



## C6000的DMA（2）

- 由于DMAC的位置，某些存储区之间的数据搬移要通过中间的buffer（一般为IDRAM）实现，否则时间消耗大。

如双口和SBSRAM的数据传输，2次DMA的时间小于一次直接DMA的时间；2次DMA传输中的第二次传输的启动一般是靠第一次DMA完成后产生的软件中断触发的
- 用DMA实现MemoryCopy和MemorySet；

片内的MemorySet可以有更快的方法，优化
- DMA的优先级：可设，但默认DMA0最高

如果高优先级DMA启动时，低优先级DMA没有完成时，要将低优先级DMA的操作挂起，等完成高优先级DMA后，继续完成。所以系统中有几个DMA同时发生时，要根据数据的重要性和数据传输处理的顺序选择合适的DMA通道。
- DMAC通用管脚在调试中的使用：发光二极管



# C6000的Timer

---

- 利用时钟中断计算系统内的各种时间, 起到计时器的作用
  - 中断发生的间隔时间可设: **1ms**
  - 计算时间间隔精度: **1us**
- 利用计时器防止等待超时

# 一个AD板的例子

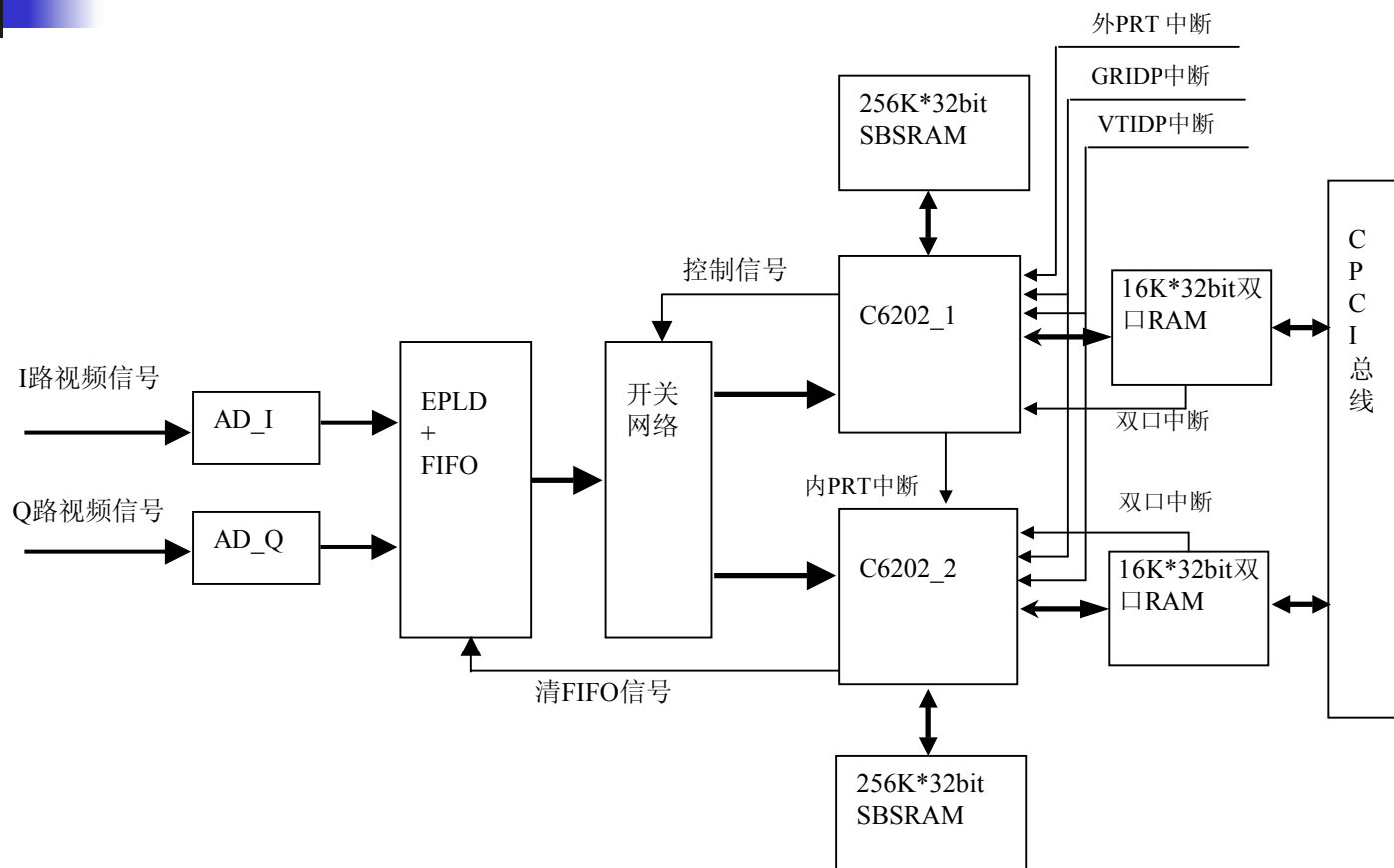


图1 AD板硬件简要框图



# AD板的中断,DMA,Timer小结

表 1. AD 板 C6202 中断

⊕

| 中断号/性质    | 中断名称       | 触发源               |         | 中断处理程序的功能              |             |
|-----------|------------|-------------------|---------|------------------------|-------------|
|           |            | C6202_1           | C6202_2 | C6202_1                | C6202_2     |
| INT4 / 硬  | PRT 中断     | 定时板               | C6202_1 | 开关网络拨向 C6202_1, 读 FIFO | 读 FIFO      |
| INT5 / 硬  | VTIDP 中断   | 定时板               |         | 命令字解析 ( ? )            |             |
| INT6 / 硬  | GRIDP 中断   | 定时板               |         | 主处理                    |             |
| INT7 / 硬  | 双口中断       | 脉压板的 C6202        |         | 把处理结果传到片内 buffer       |             |
| INT8 / 软  | PRTDMAOver | PRT 中断启动的 DMA0 结束 |         | 开关网络拨向 C6202_2, 预处理    | 清 FIFO, 预处理 |
| INT12 / 软 | DMA3Over   | 双口中断启动的 DMA3 结束   |         | 把处理结果传到双口 RAM          |             |
| INT14 / 软 | 时钟中断       | 时钟计数              |         | 测试程序片运行时间              |             |

□