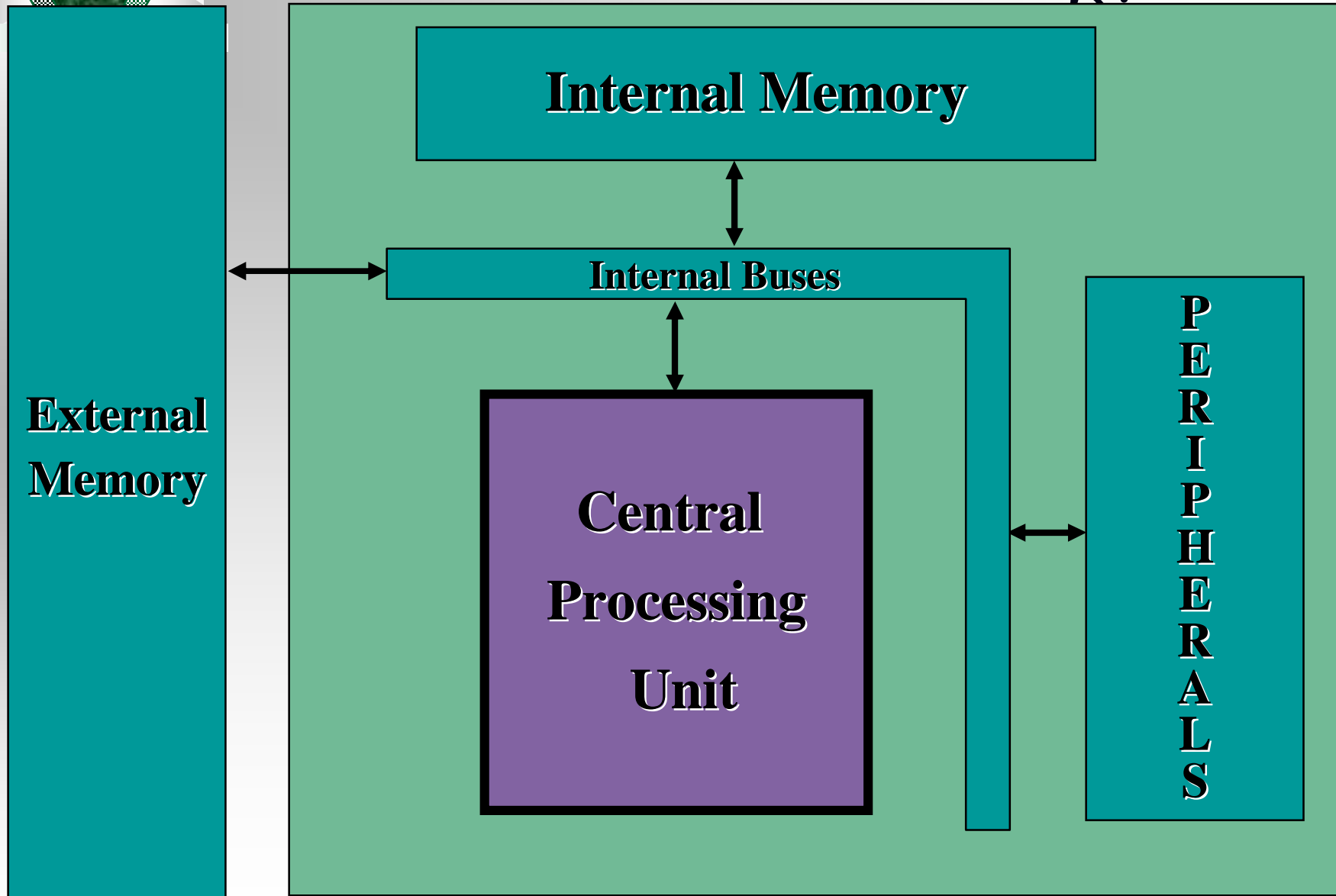


# C6000 体系结构和汇编语言

# General DSP System Block

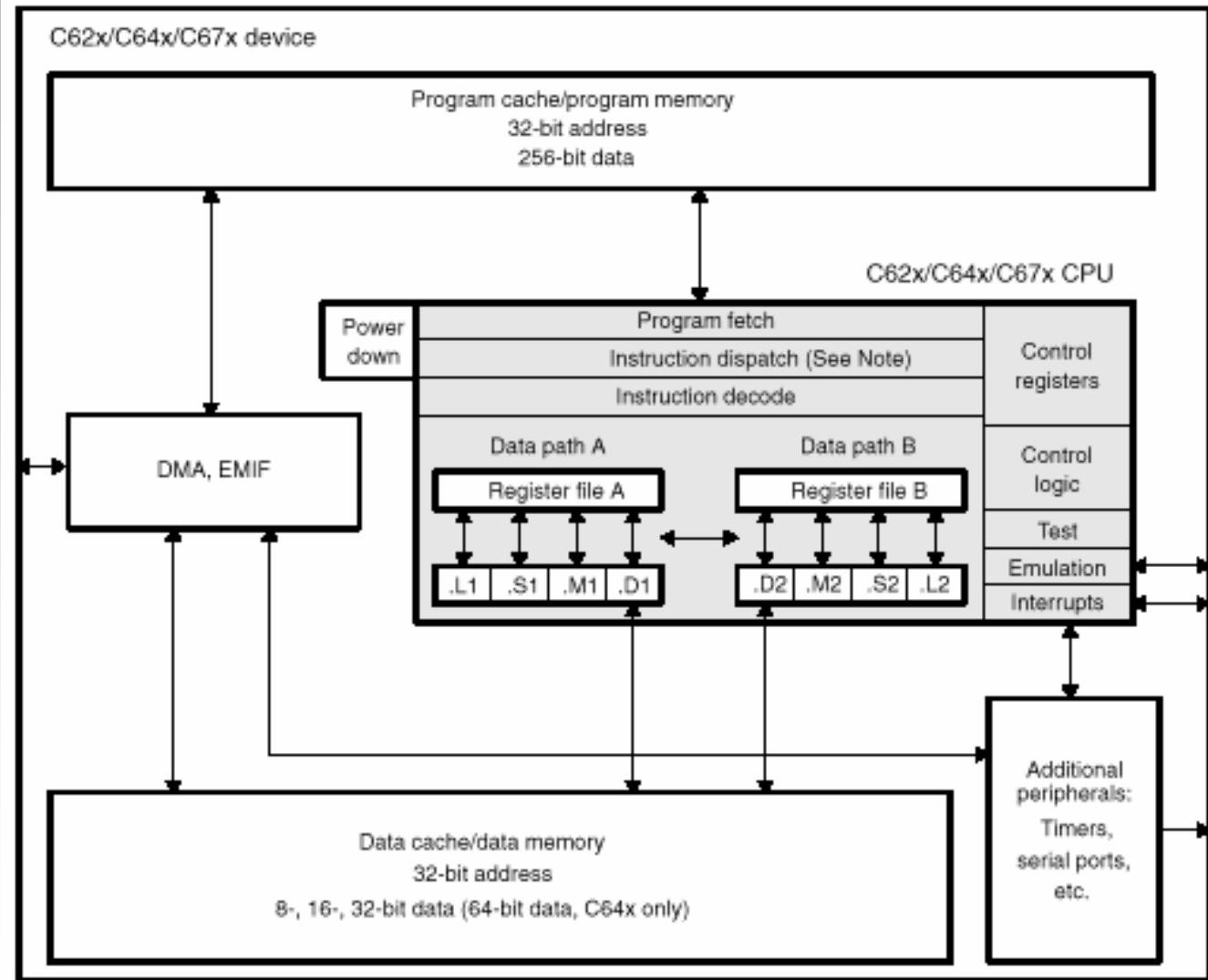
远见品质



远见品质

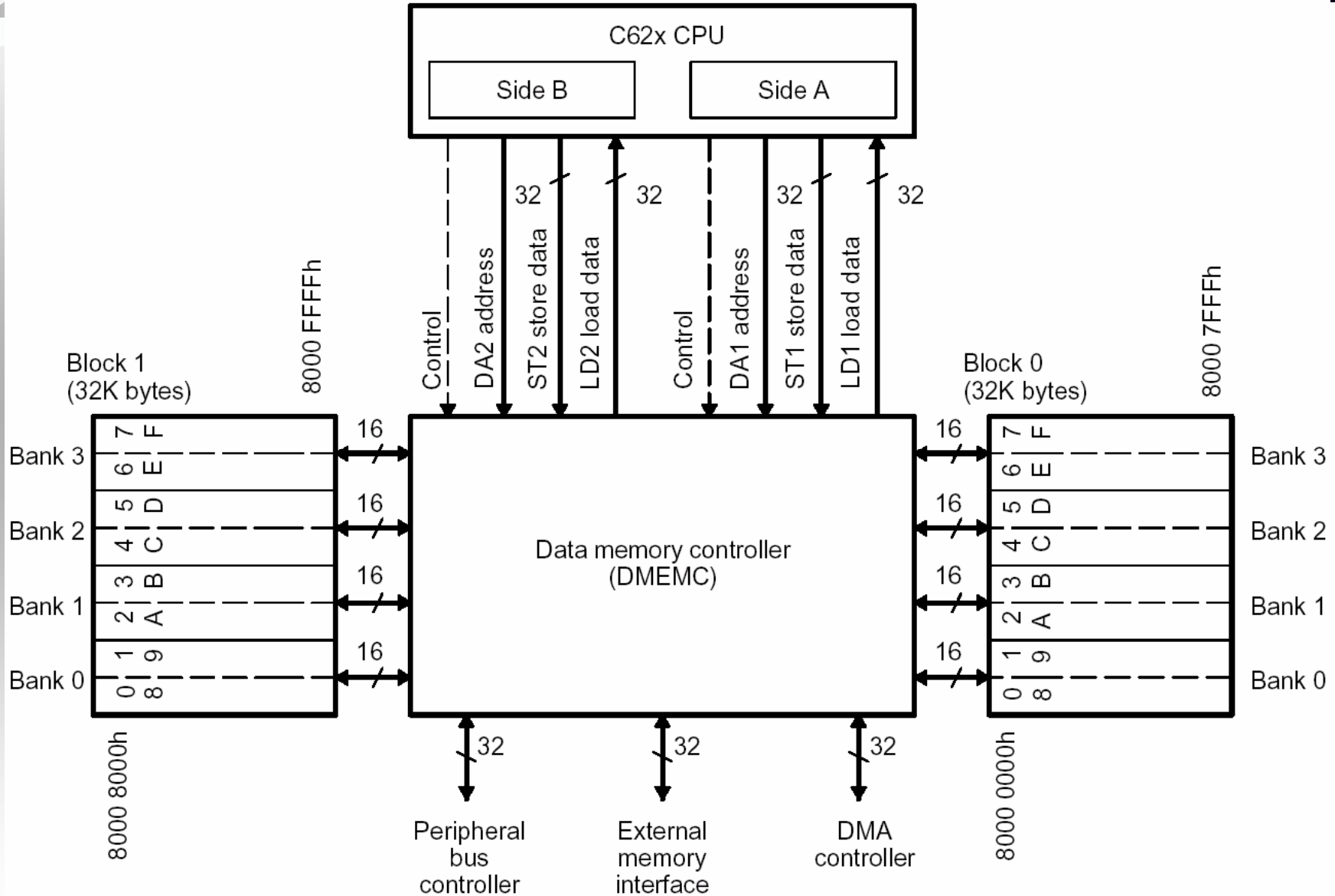
# *C6000 CPU Architecture*

VLIW, Very Long Instruction word



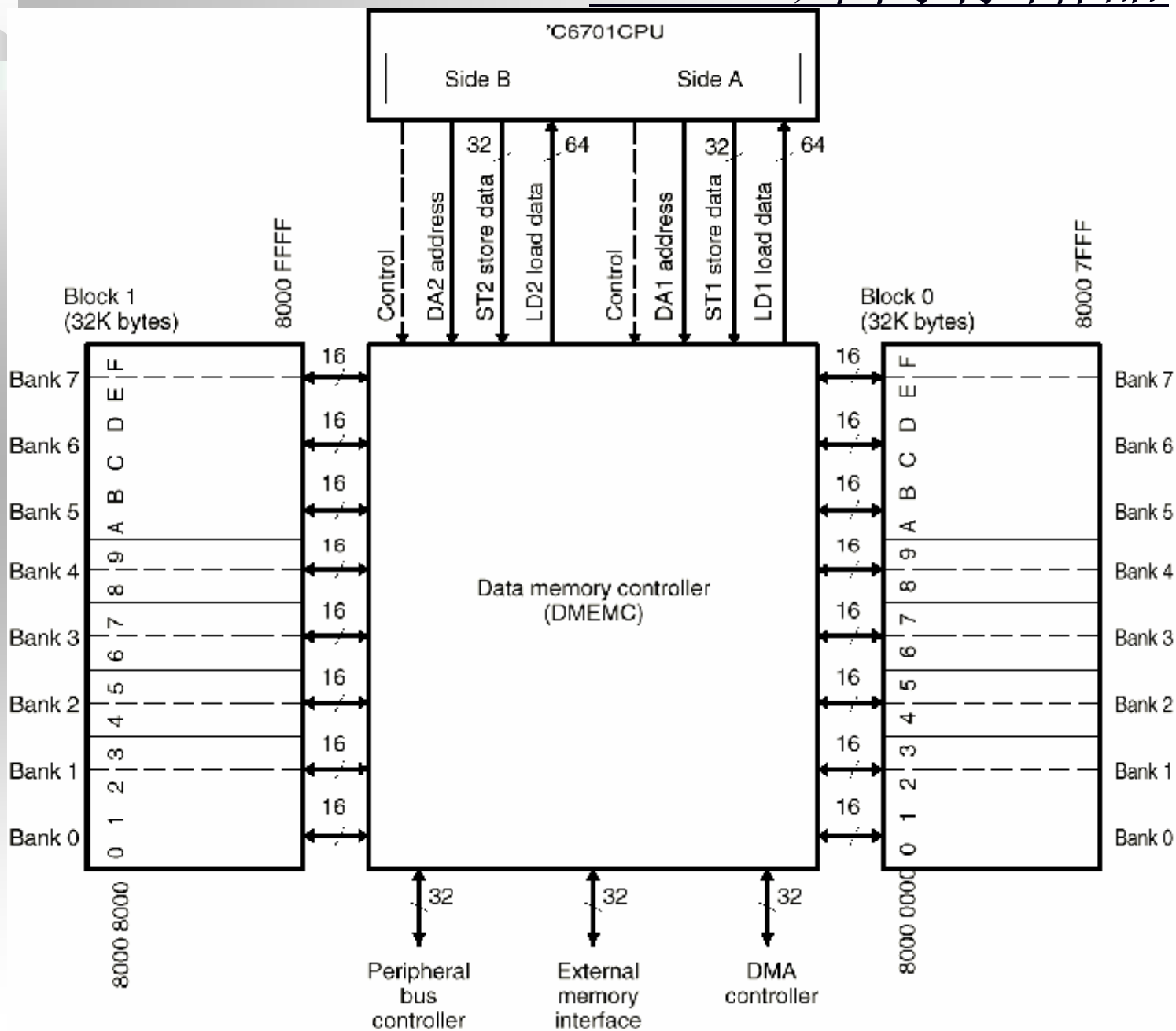


# C6201/04/05 片内存储器



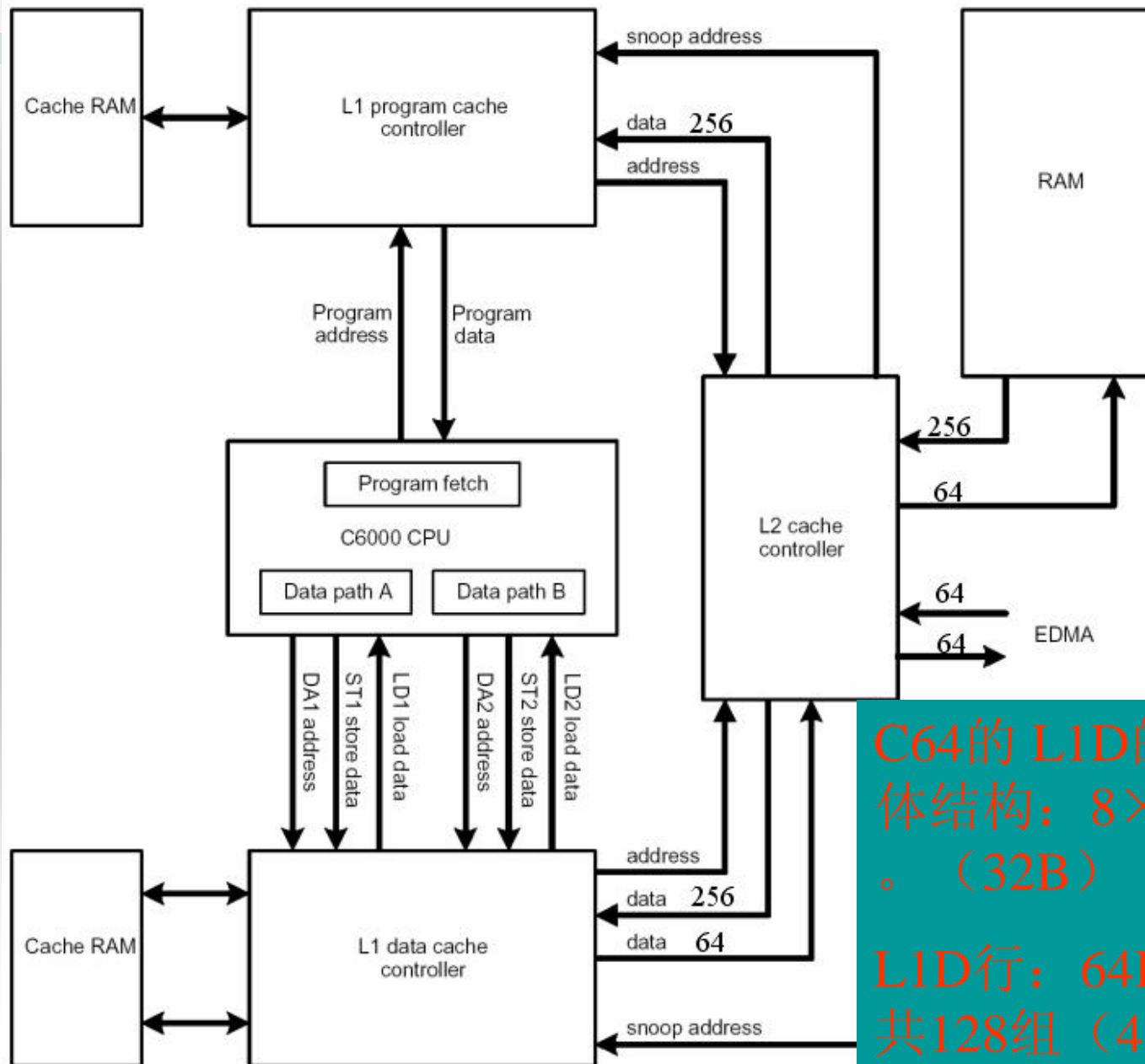
远见品质

# C6701片内存储器



远见品质

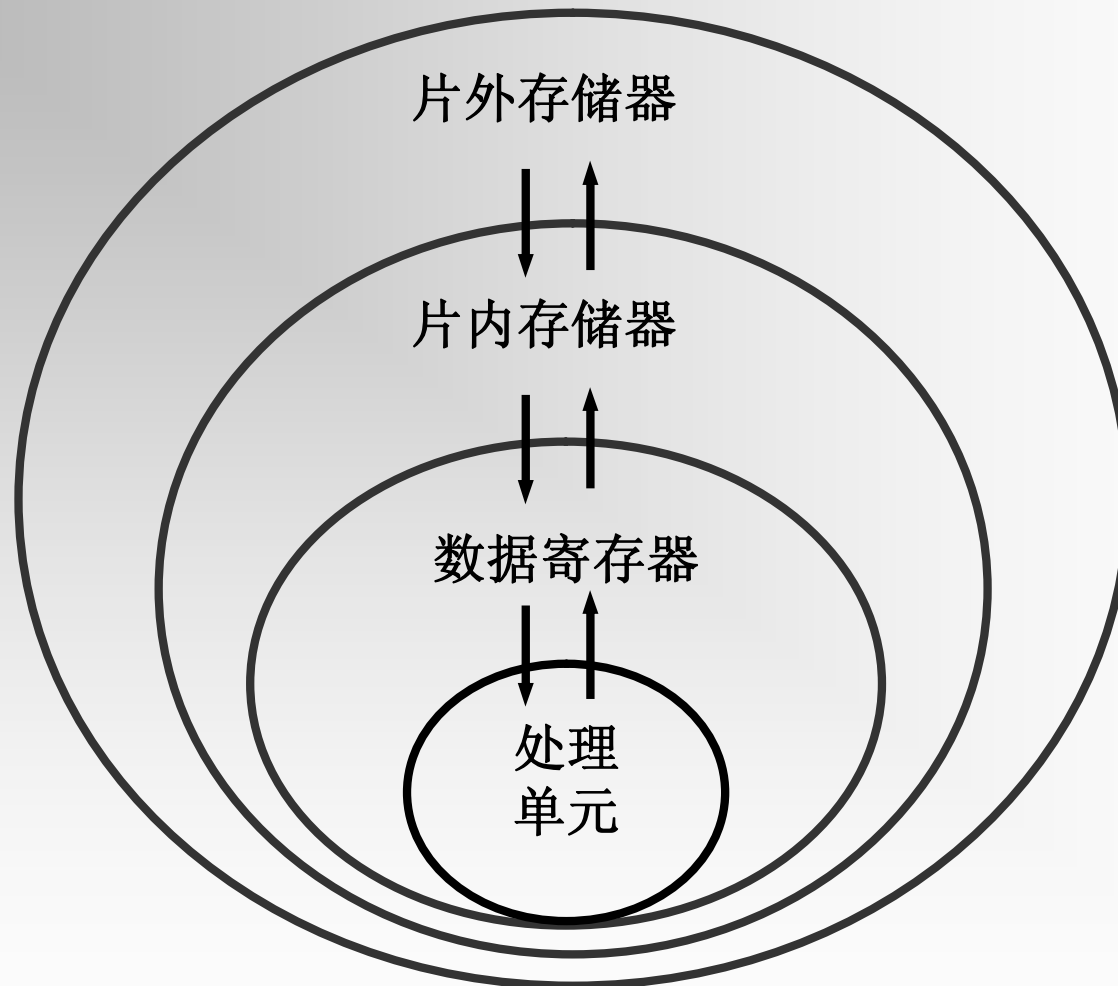
# C64x 片内2级存储器



C64的 L1D的存储体结构： $8 \times 32\text{bit}$ 。  
(32B)

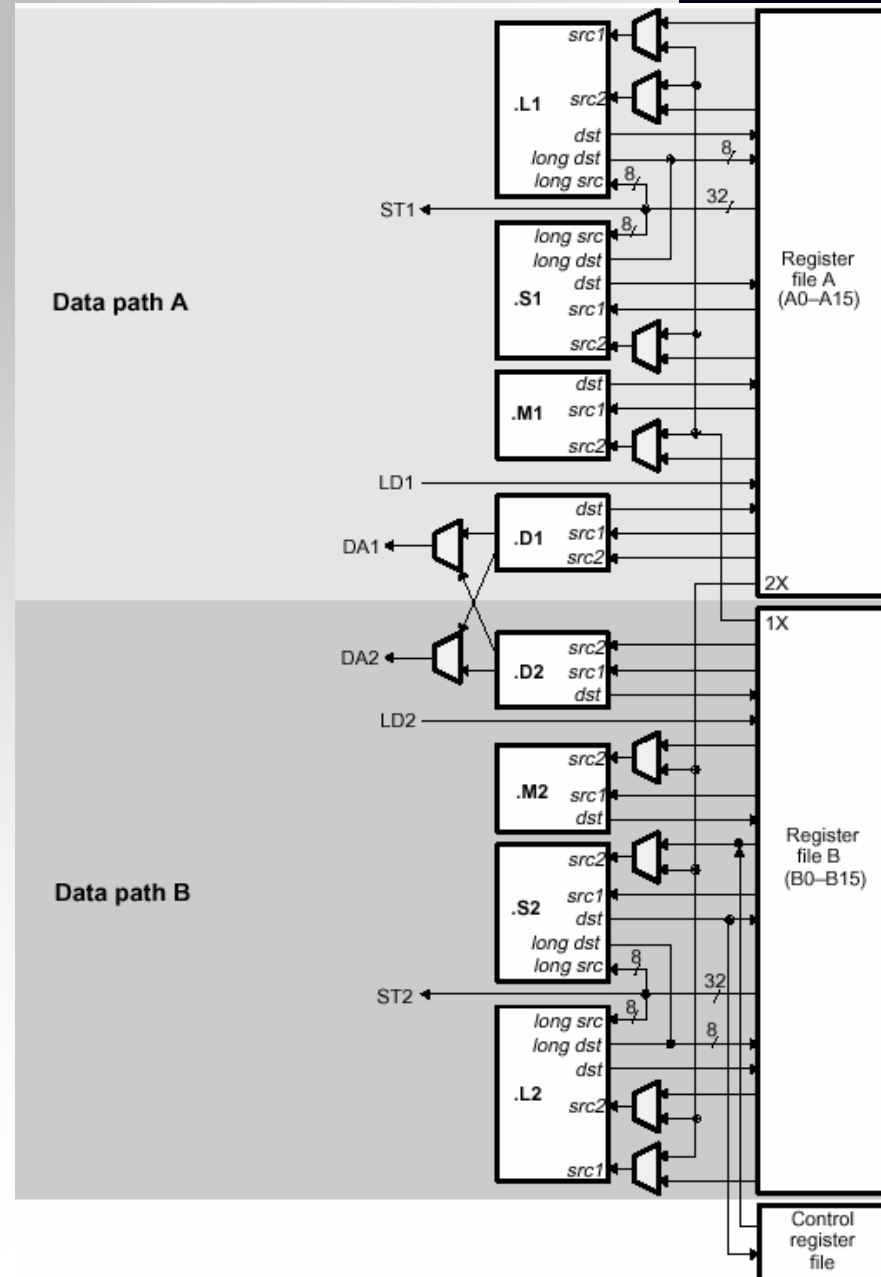
L1D行：64B。  
共128组(4K字)

# 程序员角度的DSP结构：存储器的层次



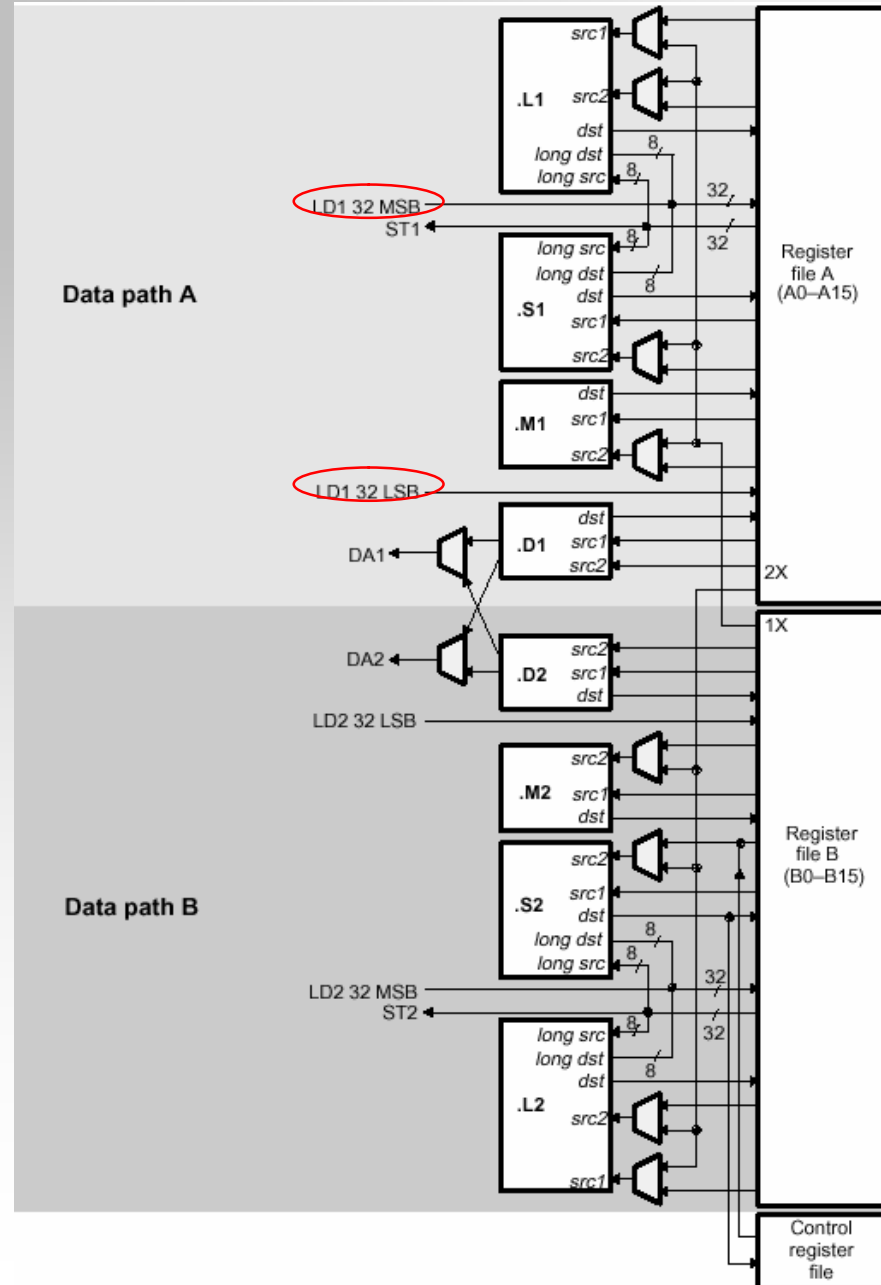
远见品质

# C62xx CPU Core



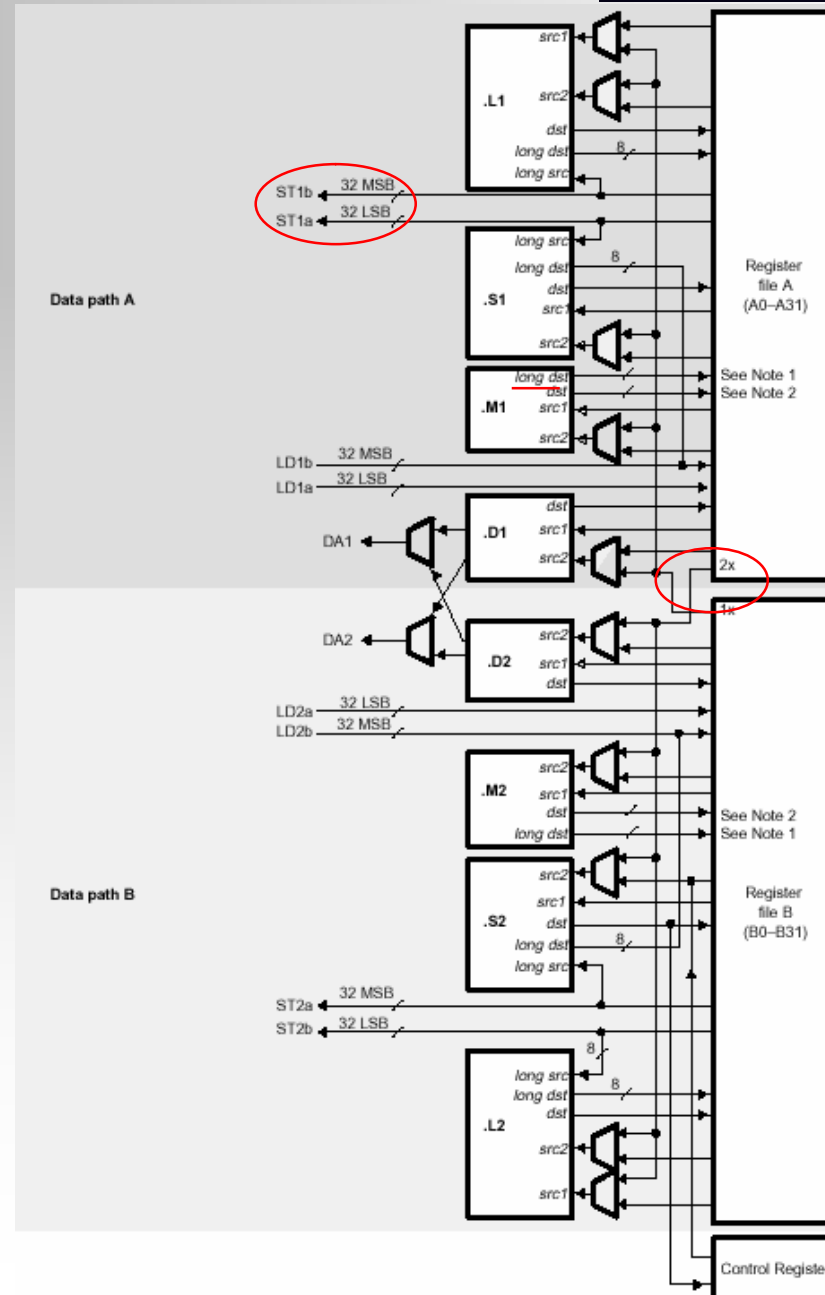
远见品质

# C67xx CPU Core



远见品质

# C64xx CPU Core



## 一条C6000的指令和其机器码

- u C6000: ADD .D2 B5,B4,B4  
ADD (.D2 or.D1) src2,src1,dst1
- u 00000010000101001000100001000010
- u 000 0 00100 00101 00100 010000 10000 1 0
- u (1) (2) (3) (4) (5) (6) (7) (8) (9)
- u (1) 条件寄存器: A1,A2,B0~2; C64添加A0
- u (2) z, 指定条件寄存器的判断条件
- u (3) dst, 目的
- u (4) src2, 源2
- u (5) src1, 源1
- u (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 ;
- u (7) 固定值
- u (8) s, 选择A边寄存器还是B边寄存器
- u (9) p, 是否并行

## 对比：8086的ADD指令

**8086/8088 一条指令长1~6字节**

**8086 ADD CX,DX CX和DX相加放到CX**

**000000 0 1 11 001 010; 寻址方式和立即数, 这里没有**

**ADD (1) (2) (3) (4) (5)**

**(1) 0 reg为目的,**

**(2) 1 字处理 (8086的字为16bit), 表示操作的是CX,DX不是CL,DL。**

**(3) 11, 寄存器模式, 和 001, 010,一起表示是CX和DX**

**(4) CX, (2) (3) 和001一起决定是CX**

**(5) DX, (2) (3) 和010一起决定是DX**



远见品质

## Implementation of Sum of Products (SOP)

SOP is the key element for most DSP algorithms.

So let's write the code for this algorithm and at the same time learn the C6000 architecture.

$$Y = \sum_{n=1}^N a_n * x_n$$
$$= a_1 * x_1 + a_2 * x_2 + \dots + a_N * x_N$$

Two basic operations are required for this algorithm.

(1) **Multiplication**

(2) **Addition**

Therefore two basic instructions are required

远见品质

## CODE

```
MVKL .S2 pt1, A5
MVKH .S2 pt1, A5

MVKL .S2 pt2, A6
MVKH .S2 pt2, A6

MVKL .S2 pt3, A7
MVKH .S2 pt3, A7
MVKL .S2 count, B0
ZERO .L A4
loop LDH .D *A5++, A0
LDH .D *A6++, A1
MPY .M A0, A1, A3
ADD .L A4, A3, A4
SUB .S B0, 1, B0
[B0] B .S loop
STH .D A4, *A7
```



远见品质

# *TMS320C62/C64/67 Instruction Set*

# 'C62/C64 Basic Instruction Set (by category)

## Arithmetic

ABS  
 ADD  
 ADDA  
 ADDK  
 ADD2  
 MPY  
 MPYH  
 NEG  
 SMPY  
 SMPYH  
 SADD  
 SAT  
 SSUB  
 SUB  
 SUBA  
 SUBC  
 SUB2  
 ZERO

## Logical

AND  
 CMPEQ  
 CMPGT  
 CMPLT  
 NOT  
 OR  
 SHL  
 SHR  
 SSSL  
 XOR

## Data Mgmt

LDB/H/W  
 MV  
 MVC  
 MVK  
 MVKL  
 MVKH  
 MVKLH  
 STB/H/W

## Program Ctrl

B  
 IDLE  
 NOP

## Bit Mgmt

CLR  
 EXT  
 LMBD  
 NORM  
 SET

Note: Refer to the 'C6000 CPU Reference Guide for more details.

远见品质

# 'C62/C64 Basic Instruction Set(by unit)

<b>.S Unit</b>	
<b>ADD</b>	<b>MVKLH</b>
<b>ADDK</b>	<b>NEG</b>
<b>ADD2</b>	<b>NOT</b>
<b>AND</b>	<b>OR</b>
<b>B</b>	<b>SET</b>
<b>CLR</b>	<b>SHL</b>
<b>EXT</b>	<b>SHR</b>
<b>MV</b>	<b>SSHL</b>
<b>MVC</b>	<b>SUB</b>
<b>MVK</b>	<b>SUB2</b>
<b>MVKL</b>	<b>XOR</b>
<b>MVKH</b>	<b>ZERO</b>

<b>.M Unit</b>	
<b>MPY</b>	<b>SMPY</b>
<b>MPYH</b>	<b>SMPYH</b>

<b>Other</b>	
<b>NOP</b>	<b>IDLE</b>

<b>.L Unit</b>	
<b>ABS</b>	<b>NOT</b>
<b>ADD</b>	<b>OR</b>
<b>AND</b>	<b>SADD</b>
<b>CMPEQ</b>	<b>SAT</b>
<b>CMPGT</b>	<b>SSUB</b>
<b>CMPLT</b>	<b>SUB</b>
<b>LMBD</b>	<b>SUBC</b>
<b>MV</b>	<b>XOR</b>
<b>NEG</b>	<b>ZERO</b>
<b>NORM</b>	

<b>.D Unit</b>	
<b>ADD</b>	<b>STB/H/W</b>
<b>ADDA</b>	<b>SUB</b>
<b>LDB/H/W</b>	<b>SUBA</b>
<b>MV</b>	<b>ZERO</b>
<b>NEG</b>	

Note: Refer to the 'C6000 CPU Reference Guide for more details.

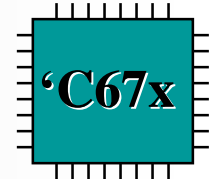
远见品质

## 'C67xx Additional Instructions (by unit)

<b>.S Unit</b>	
ABSSP	CMPLTDP
ABS DP	RCPS P
CMPGTSP	RCPD P
CMPEQSP	RSQRSP
CMPLTSP	RSQRDP
CMPGTDP	SPDP
CMPEQDP	

<b>.M Unit</b>	
MPYSP	MPYI
MPYDP	MPYID

<b>.L Unit</b>	
ADDDP	INTSP
ADDSP	INTSPU
DPINT	SPINT
DPSP	SPTRUNC
INTDP	SUBSP
INTDPU	SUBDP



<b>.D Unit</b>	
ADDAD	LDDW

Note: Refer to the 'C6000 CPU Reference Guide for more details.

1. C6000汇编语言和体系结构  
流水线
2. C6000 C运行时环境  
寄存器使用约定、函数调用约  
定、系统启动流程
3. C6000的RTOS – DSP/BIOS

本课程中其他内容:

(1) 基于CCS的嵌入式实时软件开发

与调试: 以上述1,2,3为基础

(2) C6000代码优化: 以上述1为基础



[www.farsight.com.cn](http://www.farsight.com.cn)

谢谢!