

TMS320C54x DSP

CPU 与外设

第一章 综 述

1 总线结构

C54x 包括 8 条 16 比特宽度的总线，其中：

- 一条程序总线 (PB)
- 三条数据总线 (CB、DB、EB)
- 四条地址总线 (PAB、CAB、DAB、EAB)

2 CPU

C54x 的 CPU 结构包括：

- 40 比特的 ALU，其输入来自 16 比特立即数、16 比特来自数据存储器的数据、暂时存储器、T 中的 16 比特数、数据存储器中两个 16 比特字、数据存储器中 32 比特字、累加器中 40 比特字。
- 2 个 40 比特的累加器，分为三个部分，保护位 (39 - 32 比特)、高位字 (31 - 16 比特)、低位字 (15 - 0 比特)。
- 桶型移位器，可产生 0 到 31 比特的左移或 0 到 16 比特的右移。
- 17 × 17 比特的乘法器
- 40 比特的加法器
- 比较选择和存储单元 CSSU
- 数据地址产生器 DAGEN
- 程序地址产生器 PAGEN

3 外设

C54x 包括：

- 通用 I/O 引脚，XF 和 $\overline{BIO}$
- 定时器
- PLL 时钟产生器
- HPI 口，8 比特或 16 比特
- 同步串口
- 带缓存串口，BSP
- 多路带缓存串口，McBSP
- 时分复用串口，TDM
- 可编程等待状态产生器
- 可编程 bank - switching 模块
- 外部总线接口
- IEEE1149.1 标准 JTAG 口

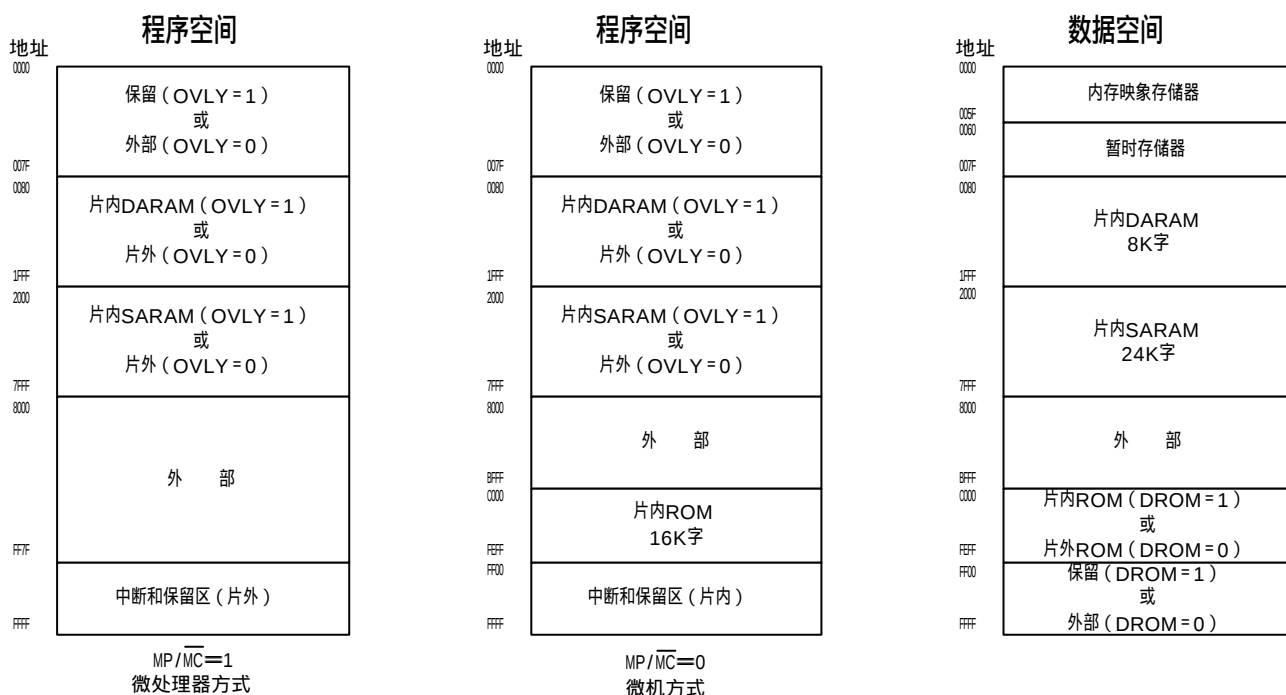
第二章 存储器

一般而言，C54x 的存储空间可达 192K16 比特字，64K 程序空间，64K 数据空间，64K I/O 空间。

依赖其并行的工艺特性和片上 RAM 双向访问的性能，在一个机器周期内，C54x 可以执行 4 条并行并行存储器操作：取指令，两操作数读，一操作数写。

使用片内存储器有三个优点：高速执行（不需要等待），低开销，低功耗。

1 存储空间分配图（以 C549 为例）



复位后，中断矢量表位于程序区 FF80H 位置，可重新定位于程序空间任何一个 128 字的页面（其地址高 9 比特即页号由 PMST 中 IPTR 确定）。

2 程序存储区

C54x 有片内 ROM、DARAM、SARAM，这些区域可以通过软件配置到程序空间。当地址落在这些区域内，自动对这些区域进行访问；当地地址落在这些区域以外，自动产生对外部存储器的访问。

2.1 片内 ROM

片内 ROM（4K，16K，24K，28K 或 48K 字）可能包括的内容有：

- 引导程序，可以从串口、外部存储器、I/O 口或 HPI 口引导
- 256 字的 μ 率扩展表
- 256 字的 A 率扩展表

- 256 字的正弦表
- 中断矢量表

2.2 扩展程序存储器

‘548、‘549、‘5410、‘5420 采用分页扩展的方式使可寻址程序空间达到 8192K 字。这一功能的实现有赖于：

- 23 条地址线
- 扩展程序计数器 XPC
- 6 条访问外部程序空间的指令

当程序空间可以使用片内 RAM 时，程序空间的每一页分为以下两部分：最大 32K 字的通用块和 32K 字的专有块。通用块为所有页共享。

XPC 寄存器指示选定页，复位后，初始化为 0。影响 XPC 的 6 条指令是：

- FB[D] – 长跳转指令
- FBACC[D] – 长跳转指令，跳转地址由 A 或 B 中内容确定
- FCALA[D] – 长调用指令，子程序地址由 A 或 B 中内容确定
- FCALL[D] – 长调用指令
- FRET[D] – 长返回指令
- FRETE[D] – 长中断返回指令

其它的指令不修改 XPC 寄存器，而在当前页中做内部访问。

3 内存映象寄存器（以 ‘549 为例）

0	IMR	中断屏蔽寄存器
1	IFR	中断标志寄存器
2-5	-	测试保留
6	ST0	状态寄存器 0
7	ST1	状态寄存器 1
8	AL	累加器 A 低字（15 – 0 比特）
9	AH	累加器 A 高字（31 – 16 比特）
A	AG	累加器 A 保护位（39 – 32 比特）
B	BL	累加器 B 低字（15 – 0 比特）
C	BH	累加器 B 高字（31 – 16 比特）
D	BG	累加器 B 保护位（39 – 32 比特）
E	T	乘法寄存器
F	TRN	传送寄存器
10-17	AR0-AR7	辅助寄存器 ARn n = 0 ~ 7
18	SP	堆栈指针
19	BK	循环缓存区大小寄存器
1A	BRC	块重复计数器
1B	RSA	块重复开始地址
1C	REA	块重复结束地址
1D	PMST	状态寄存器

1E	XPC	PC 扩展寄存器
1E—1F	—	保留
20	BDRR0	带缓存串口 0 数据接收寄存器
21	BDXR0	带缓存串口 0 数据发送寄存器
22	BSPC0	带缓存串口 0 控制寄存器
23	BSPCE0	带缓存串口 0 控制扩展寄存器
24	TIM	定时器计数器
25	PRD	定时器周期寄存器
26	TCR	定时器控制寄存器
27	—	保留
28	SWWSR	等待状态产生寄存器
29	BSCR	Bank-switching 控制寄存器
2A	—	保留
2B	XSWR	扩展等待状态寄存器
2C	HPIC	主机接口控制寄存器
2D—2F	—	保留
30	TRCN	TDM 串口数据接收寄存器
31	TDXR	TDM 串口数据发送寄存器
32	TSPC	TDM 串口控制寄存器
33	TCSR	TDM 串口通道选择寄存器
34	TRTA	TDM 串口接收发送寄存器
35	TRAD	TDM 串口接收地址寄存器
36—37	—	保留
38	AXR0	ABU0 发送地址寄存器
39	BKX0	ABU0 发送缓冲大小寄存器
3A	ARR0	ABU0 接收地址寄存器
3B	BKR0	ABU0 接收缓冲大小寄存器
3C	AXR1	ABU1 发送地址寄存器
3D	BKX1	ABU1 发送缓冲大小寄存器
3E	ARR1	ABU1 接收地址寄存器
3F	BKR1	ABU1 接收缓冲大小寄存器
40	BDRR1	带缓存串口 1 数据接收寄存器
41	BDXR1	带缓存串口 1 数据发送寄存器
42	BSPC1	带缓存串口 1 控制寄存器
43	BSPCE1	带缓存串口 1 控制扩展寄存器
44—57	—	保留
58	CLKMD	时钟方式寄存器
59—5F	—	保留

下面着重对三个状态寄存器作一简要说明。

3.1 ST0 寄存器

ST0 结构为:

15 - 13	12	11	10	9	8 - 0
ARP	TC	C	OVA	OVB	DP

- ARP - 辅助寄存器指针
- TC - 测试/控制位, 存储了 ALU 测试位操作的结果, 受 BIT, BITF, CMPM, CMPR, CMPS, SFTC 指令的影响
- C - 如果加法产生进位 C 为 1, 减法产生借位 C 为 0。ADD 指令只能置位 C, 而 SUB 指令只能清 C
- OVA - 累加器 A 溢出标志
- OVB - 累加器 B 溢出标志
- DP - 数据存储空间页标志, 由 DP 指定页, 在此页中采用直接寻址指令。其绝对地址为 DP 指示的高 9 位加上直接寻址指令中的地址为其低 7 位构成。

3.2 ST1 寄存器

ST1 结构为:

15	14	13	12	11	10	9	8	7	6	5	4-0
BRAF	CPL	XF	HM	INTM	0	OVM	SXM	C16	FRCT	CMPT	ASM

- BRAF - 块重复指示。BRAF = 1, 块重复操作。
- CPL - 编译器模式, 指示相关直接寻址选用指针。CPL = 0, 使用 DP 指针; CPL = 1, 使用 SP 指针。
- XF - 指示外部引脚 XF 状态。
- HM - 挂起方式。指示 CPU 响应 $\overline{HOLD}$ 信号的方式。HM = 0, 外部接口呈高阻; HM = 1, CPU 停机。
- INTM - 全局中断控制位。INTM = 0, 开中断; INTM = 1, 屏蔽可屏蔽中断。
- 0 - 读出值总为 0。
- OVM - 溢出处理方式。指示发生溢出时, 对累加器中数值的处理。OVM = 0, 溢出值不变; OVM = 1, 载入正向最大值 007FFFFFFFH 或负向最大值 0080000000H。
- SXM - 符号扩展方式。SXM = 0, 符号不扩展; SXM = 1, 数据被 ALU 使用前先行符号扩展。
- C16 - C16 = 0, ALU 操作采用双精度方式 (全 32 位方式), C16 = 1, ALU 操作采用双 16 位方式。
- FRCT - 小数模式。FRCT = 1, 乘法器输出左移一位补偿多余的符号位。
- CMPT - 兼容模式。CMPT = 0, ARP 在只有单数据存储器操作数的间接寻址方式中不更新, 在这种方式下, ARP 必须总置 0; CMPT = 1, ARP 在上述条件下更新, 除非使用 AR0。
- ASM - 累加器移位数。规定了移位范围 -16 - 15, 用于并行存储

指令 STH、STL、ADD、SUB 和 LD。

3.3 PMST 寄存器

ST1 结构为：

15—7	6	5	4	3	2	1	0
IPTR	MP/MC	OVLY	AVIS	DROM	CLKOFF	SMUL	SST

- IPTR – 中断矢量指针。此 9 位指向内存空间中以 128 字为单位的页首址。即中断矢量必须位于页的起始部分。复位后 IPTR 为 1FFH，指向 FF80H 的位置，可重新定义到任何页。
- $MP/\overline{MC}$ – 微机/微处理器模式。 $MP/\overline{MC} = 0$ ，微机模式，片内 ROM 可访问； $MP/\overline{MC} = 1$ ，微处理器模式，片内 ROM 不可访问。
- OVLY – 片内 RAM 是否配置入程序空间。OVLY = 0，片内 RAM 不配置入程序空间；OVLY = 1，除 00 – 7FH 外，片内 RAM 配置入程序空间。
- AVIS – 地址显示模式。控制对内部程序访问时，地址数据是否显示在地址线上。AVIS = 0，不显示；AVIS = 1，显示。
- DROM – 数据 ROM 配置。DROM = 0，片内 ROM 不配置在数据空间；DROM = 1，部分片内 ROM 配置在数据空间。
- CLKOFF – CLKOUT 关闭。CLKOFF = 1，CLKOUT 输出禁止，保持高电平。
- SMUL – 乘法溢出处理。当 SMUL = 1，且 OVM = 1，FRCT = 1 时，对 MAC（乘累加）和 MAS（乘累减）指令的操作基于 ETSI GSM 规范。体现在在小数模式下，在进行后续加/减之前，8000H × 8000H 的结果被调整为 7FFFFFFFH。这等同于在 OVM = 1 下 MPY + ADD 指令。如果只有 OVM = 1，而 SMUL 不为 1，只在加/减结果后作溢出调整。
- SST – 存储溢出处理。当 SST = 1，累加器中数据在存储到数据空间之前进行溢出调整。影响指令有：STH，STL，STLM，DST，ST||ADD，ST||LD，ST||MACR[R]，ST||MAS[R]，ST||MPY 和 ST||SUB。步骤是：
 - 1) 依据指令，累加器中数据完成左移或右移。
 - 2) 40 比特值根据 SXM 位调整为 32 位。SXM = 0（不符号扩展），若值大于 7FFFFFFFH，取 7FFFFFFFH；若 SXM = 1（符号扩展），若值大于 7FFFFFFFH，取 7FFFFFFFH，若值小于 80000000H，取 80000000H。
 - 3) 调整后数据存入数据空间。
 - 4) 在调整过程中，累加器中数据不改变。

3.4 累加器的存储操作

累加器中数据存储，AH 和 AL 部分比较好办，采用 STH、STL 和 STLM 即可完成。要存储 AG 中的内容，则需要采用间接方式，如累加器 A

中内容为 0FF 4321 1234H，要存储 AG 部分内容至 TEMP 存储器中，可用“STH A, -8, TEMP”指令来完成，TEMP = FF43H。

第三章 数据寻址

C54x 提供七类寻址方式：

- 立即数寻址
- 绝对地址寻址
- 累加器寻址
- 直接寻址
- 间接寻址
- 内存映象寄存器寻址
- 堆栈寻址

下面着重介绍以下几种方式。

1 绝对地址寻址

术语解释：dmad - 数据存储地址；pmad - 程序存储地址；PA - 口地址；lk - 长整数。

绝对地址寻址包括四类：

- dmad 寻址
- pmad 寻址
- PA 寻址
- $*(lk)$ 寻址

1.1 dmad 寻址

dmad 寻址用一标号标记数据空间地址，这类指令有：

- MVDK Smem, dmad
- MVDM dmad, MMR
- MVKD dmad, Smem
- MVMD MMR, dmad

如：MVKD SAMPLE, *AR5。此处 SAMPLE 即是 dmad。

1.2 pmad 寻址

pmad 寻址用一标号标记程序空间地址，这类指令有：

- FIRS Xmem, Ymem, pmad
- MACD Smem, pmad, src
- MACP Smem, pmad, src
- MVDP Smem, pmad
- MVPD pmad, Smem

如：MVPD TABLE, *AR7-。此处 TABLE 即为 pmad。

1.3 PA 寻址

PA 寻址用一标号标记外部 I/O 口地址，这类指令有：

- PORTR PA, Smem
- PORTW Smem, PA

如：PORTR FIFO, *AR5。此处 FIFO 即为 PA。

1.4 $*(lk)$ 寻址

$*(lk)$ 寻址也是用标号标记数据空间地址。

如：LD *(BUFFER),A。使用这类指令的好处在于不用修改 DP 和 AR 值。但有一点要注意的是，这类指令不能用于重复执行单指令中 (RPT, RPTZ)。

2 直接寻址

在这类指令中，指令中的地址标号构成 dmad 的低 7 位 (DP 方式) 或作为正向偏移量 (SP 方式)。所以指令中地址标号不得超过 7 位，采用 DP 或 SP 方式，由 ST1 中 CPL 位决定。CPL = 0，采用 DP 方式；CPL = 1，采用 SP 方式。

2.1 DP 方式

DP 方式，DP 中内容作为 dmad 地址高 9 位，指令中地址标号作为低 7 位构成 dmad 地址。

2.2 SP 方式

SP 方式，以 SP 中内容作为 dmad 基地址，指令中地址标号作为正向偏移量，二者相加构成 dmad 地址。

3 间接寻址

间接寻址是指 dmad 由辅助寄存器 AR0 - AR7 内容构成。使用间接寻址的灵活性不仅在于一条指令完成读或写数据存储区 (单操作数)，还在于一条指令可以完成两个不同存储区的读操作，或两个不同存储区的写操作，或对两个不同位置的一读一写操作。

3.1 对 SARAM 的访问

对 SARAM 访问使用寄存器 AR0 - AR7，其中 AR0 为变址寄存器。用法见下表：

语 法	功 能	描 述
*ARx	dmad=ARx	ARx 包含 dmad
*ARx -	dmad=ARx ARx=ARx - 1	访问后，ARx 中地址内容减 1
*ARx+	dmad=ARx ARx=ARx+1	访问后，ARx 中地址内容加 1
*+ARx	dmad=ARx+1 ARx=ARx+1	访问前，ARx 中地址内容加 1
*ARx - 0B	dmad=ARx ARx=B (ARx - AR0)	访问后，ARx 中内容减去 AR0 中内容，并产生反向借位翻转 (用于 FFT 算法)
*ARx - 0	dmad=ARx ARx=ARx - AR0	访问后，ARx 中内容减去 AR0 中内容
*ARx+0	dmad=ARx ARx=ARx+AR0	访问后，ARx 中内容加上 AR0 中内容
*ARx+0B	dmad=ARx ARx=B (ARx+AR0)	访问后，ARx 中内容加上 AR0 中内容，并产生反向进位翻转 (用于 FFT 算法)

*ARx-%	dmad=ARx ARx=circ(ARx-1)	访问后, ARx 中内容按循环寻址方式减 1
*ARx-0%	dmad=ARx ARx=circ(ARx-AR0)	访问后, ARx 中内容按循环寻址方式减去 AR0 中内容
*ARx+%	dmad=ARx ARx=circ(ARx+1)	访问后, ARx 中内容按循环寻址方式加 1
*ARx+0%	dmad=ARx ARx=circ(ARx+AR0)	访问后, ARx 中内容按循环寻址方式加上 AR0 中内容
*ARx(1k)	dmad=ARx+1k ARx=ARx	ARx 中内容加上 16 位长偏移 (1k) 为 dmad, ARx 不更新
*+ARx(1k)	dmad=ARx+1k ARx=ARx+1k	ARx 中内容加上 16 位长偏移 (1k) 为 dmad, ARx 更新
*+ARx(1k)%	dmad=circ(ARx+1k) ARx=circ(ARx+1k)	ARx 中内容按循环寻址方式加上 16 位长偏移 (1k) 为 dmad, ARx 更新
*(1k)	dmad=1k	16 位绝对地址寻址

循环寻址操作如下:

寄存器 BK 作为循环缓存区大小 (R)。循环缓存区基地址的低 N 位必须为 0。R 和 N 满足以下关系: $2^N \leq R$ 。末地址为基地址 + R。步进值 $step \geq 1$ 。变址 index 由以下算法确定:

若 $0 \leq index + step \leq BK$, $index = index + step$;

若 $index + step \geq BK$, $index = index + step - BK$;

若 $index + step \leq 0$, $index = index + step + BK$ 。

3.2 对 DARAM 的访问

对 DARAM 访问使用辅助寄存器 AR2 - AR5, 用法见下表:

语 法	功 能	描 述
*ARx	dmad=ARx	ARx 包含 dmad
*ARx-	dmad=ARx ARx=ARx-1	访问后, ARx 中地址内容减 1
*ARx+	dmad=ARx ARx=ARx+1	访问后, ARx 中地址内容加 1
*ARx+0%	dmad=ARx ARx=circ(ARx+AR0)	访问后, ARx 中内容按循环寻址方式加上 AR0 中内容

第四章 中断系统

硬件和软件驱动都可以使 C54x 产生中断。软件中断是指由程序指令引起的中断，这类指令有：INTR，TRAP，RESET。硬件中断可由外部硬件引发，或由片内外设内部引发。无论软件中断还是硬件中断都可分为可屏蔽中断和不可屏蔽中断。C54x 处理中断按以下三个步骤：

- 接收中断请求
- 响应中断
- 执行中断服务程序

3.2 中断标志寄存器 IFR（以 C549 为例）

15- 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	BMI NT1	BMI NT0	BXI NT1	BRI NT1	HPI NT	INT 3	TXI NT	TRI NT	BXI NT0	BRI NT0	TIN T	INT 2	INT 1	INT 0

IFR 中某位为 1 表明有相应中断，再次写 1 清除中断。

3.2 中断屏蔽寄存器 IMR（以 C549 为例）

15- 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	BMI NT1	BMI NT0	BXI NT1	BRI NT1	HPI NT	INT 3	TXI NT	TRI NT	BXI NT0	BRI NT0	TIN T	INT 2	INT 1	INT 0

要打开某个中断，只需向 IMR 中相应某位写 1。而 ST1 中 INTM 位为中断总开关。

3.3 中断位置与优先级（以 C549 为例）

中断号	优先级	名称	位置	功能
0	1	RS/SINTR	0	复位（软硬件）
1	2	NMI/SINT16	4	非屏蔽中断
2	—	SINT17	8	软件中断 17
3	—	SINT18	C	软件中断 18
4	—	SINT19	10	软件中断 19
5	—	SINT20	14	软件中断 20
6	—	SINT21	18	软件中断 21
7	—	SINT22	1C	软件中断 22
8	—	SINT23	20	软件中断 23
9	—	SINT24	24	软件中断 24
10	—	SINT25	28	软件中断 25

11	-	SINT26	2C	软件中断 26
12	-	SINT27	30	软件中断 27
13	-	SINT28	34	软件中断 28
14	-	SINT29	38	软件中断 29
15	-	SINT30	3C	软件中断 30
16	3	INT0/SINT0	40	外部中断 0
17	4	INT1/SINT1	44	外部中断 1
18	5	INT2/SINT2	48	外部中断 2
19	6	TINT/SINT3	4C	定时器中断
20	7	BRINT0/SINT4	50	带缓存串口 0 接收中断
21	8	BXINT0/SINT5	54	带缓存串口 0 发送中断
22	9	TRINT/SINT6	58	TDM 串口接收中断
23	10	TXINT/SINT7	5C	TDM 串口发送中断
24	11	INT3/SINT8	60	外部中断 3
25	12	HINT/SINT9	64	HPI 口中断
26	13	BRINT1/SINT10	68	带缓存串口 1 接收中断
27	14	BXINT1/SINT11	6C	带缓存串口 1 发送中断
28	15	BMINT0/SINT14	70	BSP0 失步检测中断
29	16	BMINT1/SINT13	74	BSP1 失步检测中断
30-31	-		78-7F	保留