PRU 开发详解

唐超伦 DSP 技术支持工程师

摘要

TI OMAPL13x, C674x, AM180x，OMAP-L137，C6747，AM170x 等芯片上（本文统一以 OMAP-L138 代称）有一个 PRUSS（Programmable Realtime Unit Subsystem），可独立编程实现一些实时性要求高的个性化需求，实现产品的差异化。本文介绍 PRU 处理器架构，开发，调试方法。

目录

图

表

PRUSS 简介

在系统架构上，PRUSS 是连接在 OMAPL138 内部总线 SCR 上的一个模块，与系统中其它主模块如 ARM，DSP 一样，可以访问芯片上的其它外设，工作在 PLL0_SYSCLK2 时钟域，即 ARM/DSP 频率的一半。

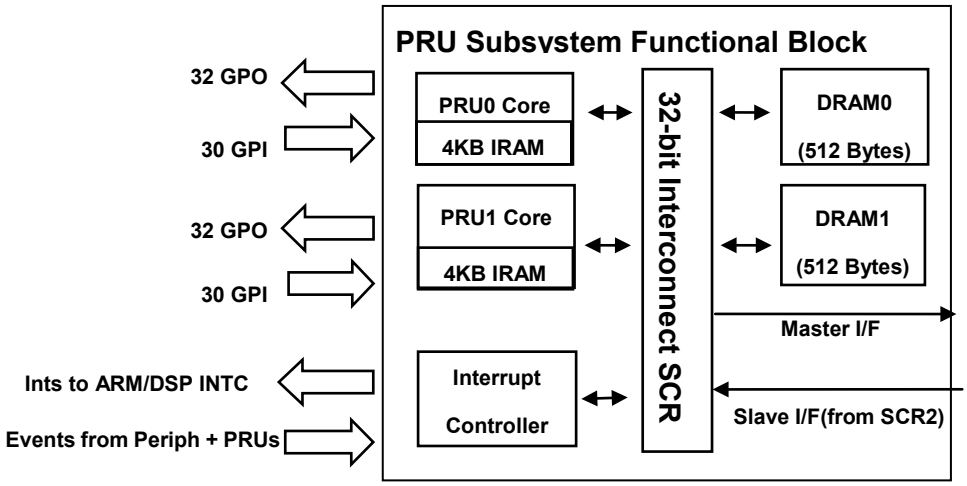


图 1 PRUSS 框图

PRUSS 包括两个 PRU，通过自己的 SCR 与子系统的中断控制器、指令内存、数据内存、以及系统 SCR 相连。PRU 不是一个加速器，它是 32-bit Load/Store RISC 架构小端处理器，每个 PRU 有 32 个通用寄存器 R0～R31，4K byte 指令 RAM，512 byte 数据 RAM，指令 RAM 是独立的，互相之间不能访问，但数据 RAM 可以通过映射地址互相访问；专用的 30 个输入引脚和 32 个输出引脚（注：OMAPL137 系列没有 PRU 外部引脚）。PRU 没有 Cache、指令流水线、及乘法指令。

1 PRU 内存映射

1.1 指令空间

每个 PRU 独立的指令空间为 0x00000000～0x00000FFF。指令空间由外部主处理器 ARM/DSP 初始化，程序指针 PC 是 32bit 字地址，不是字节地址，如 PC=2，代表指令地址 0x08。

表 1 PRU 指令空间映射表

起始地址	结束地址	PRU0	PRU1
0x00000000	0x00000FFF	PRU0 Instruction RAM	PRU1 Instruction RAM

1.2 数据空间

每个 PRU 独立的 512byte 数据 RAM 空间为 0x0000~0x01FF，因为数据 RAM 连接在 PRU 子系统的 SCR 上，所以子系统中的其它主模块也可以访问到这块空间，这段内存空间在另一个 PRU 上的映射地址为 0x2000~0x21FF。

位于数据空间的还有共用的中断控制器寄存器；PRU 控制/状态寄存器，有各自的地址空间。

表 2 PRUSS 本地数据空间内存映射表

起始地址	结束地址	PRU0	PRU1
0x00000000	0x000001FF	Data RAM 0*	Data RAM 1*
0x00000200	0x00001FFF	Reserved	Reserved
0x00002000	0x000021FF	Data RAM 1*	Data RAM 0*
0x00002200	0x00003FFF	Reserved	Reserved
0x00004000	0x00006FFF	INTC Registers	INTC Registers
0x00007000	0x000077FF	PRU0 Registers	PRU0 Registers
0x00007800	0x00007FFF	PRU1 Registers	PRU1 Registers
0x00008000	0x0000FFFF	Reserved	Reserved
0x00010000	0xFFFFFFFF	Reserved	Reserved

1.3 全局地址空间映射

PRU 局部地址空间在系统全局地址空间映射端口为 0x01C30000，如表 3 所示，PRU 可以通过表 2 的局部地址，也可以通过表 3 的全局地址访问 PRUSS 的数据空间，通过全局地址空间访问要经过系统 SCR2，比通过局部地址空间访问要慢。PRUSS 外部主模块如 ARM，DSP 等可通过全局地址空间访问 PRU 资源。

表 3 PRUSS 全局空间内存映射表

起始地址	结束地址	区域
0x01C30000	0x01C301FF	Data RAM 0
0x01C30200	0x01C31FFF	Reserved

0x01C32000	0x01C321FF	Data RAM 1
0x01C32200	0x01C33FFF	Reserved
0x01C34000	0x01C36FFF	INTC Registers
0x01C37000	0x01C377FF	PRU0 Registers
0x01C37800	0x01C37FFF	PRU1 Registers
0x01C38000	0x01C38FFF	PRU0 Instruction RAM
0x01C39000	0x01C3BFFF	Reserved
0x01C3C000	0x01C3CFFF	PRU1 Instruction RAM
0x01C3D000	0x01C3FFFF	Reserved

2 控制/状态寄存器

PRU0 的控制/状态寄存器地址位于 0x00007000～0x000077FF，PRU1 的控制/状态寄存器地址位于 0x00007800～0x00007FFF，寄存器列表如表 4 所示，各寄存器的详细说明请参阅[1]。

表 4 PRU 控制/状态寄存器表

偏移地址	寄存器	寄存器说明
0x0000	CONTROL	PRU Control Register
0x0004	STATUS	PRU Status Register
0x0008	WAKEUP	PRU Wakeup Enable Register
0x000C	CYCLECNT	PRU Cycle Count
0x0010	STALLCNT	PRU Stall Count
0x0020	CONTABBLKIDX0	PRU Constant Table Block Index Register 0
0x0028	CONTABPROPTR0	PRU Constant Table Programmable Pointer Register 0
0x002C	CONTABPROPTR1	PRU Constant Table Programmable Pointer Register 1
0x0400 –0x047C	INTGPR0-INTGPR31	PRU Internal General Purpose Registers (for Debug)
0x0480 –0x04FC	INTCTER0-INTCTER 31	PRU Internal Constants Table Entry Registers (for Debug)

2.1 CONTROL 控制寄存器

外部主模块 ARM/DSP 通过控制寄存器可以控制 PRU 的运行状态。

表 5 PRU 控制寄存器说明表

位域	名称	R/W	功能描述
31:16	PCRESETVAL	R/W	Program Counter Reset Value:控制 PRU 复位后的起始运行地址
15	RUNSTATE	R	0：PRU 暂停，主机可访问指令内存和调试寄存器 1：PRU 正在运行，主机不能访问指令内存和调试寄存器
14:9	RESERVED	R	保留
8	SINGLESTEP	R/W	控制 ENABLE 使能时，PRU 的运行规则 0：PRU 全速运行 1：PRU 运行一条指令，然后清除 ENABLE 位。
7:4	RESERVED	R	保留
3	CONTERENABLE	R/W	使能 CYCLECNT 计数
2	SLEEPING	R/W	0：PRU 退出睡眠状态 1：PRU 进入睡眠状态
1	ENABLE	R/W	控制 PRU 取指令 0：PRU 停止取指 1：PRU 允许取指 写 0 时，如果多周期指令（如 LBxO,SBxO,SCAN 等）已经完成初始周期，那么当前指令运行完，再暂停 PRU，否者立即暂停 PRU，所以 ENABLE 不能完全指示 PRU 的状态，RUNSTATE 完全指示 PRU 的运行状态。 PRU 在暂停时，内部状态保持一致，重新 ENABLE 后，程序从原来暂停的位置继续运行。
0	SOFTRESET	R	写 0 复位 PRU，一个周期后恢复为 1

2.2 STATUS 状态寄存器

状态寄存即 PRU 的程序指针寄存器，与程序的真正运行状态有一个周期的延时。

2.3 WAKEUP 唤醒使能寄存器

在程序执行 SLP 指令进入睡眠状态之前，使能 WAKEUP 寄存器相应的位，当输入状态寄存器 R31 相应的位置 1 时，即 WAKEUP&R31 ！= 0 时，唤醒 PRU。

2.4 CYCLECNT 周期计数器

当 CONTROL[ENABLE]=1 和 CONTROL[COUNTENABLE]=1 时，CYCLECNT 以 PRU 时钟周期计数。当 CONTROL[ENABLE]=0 或 CONTROL[COUNTENABLE]=0 时，计数停止。当重新使能时，恢复继续计数。

2.5 STALLCNT 取指停止计数器

当 CONTROL[ENABLE]=1 和 CONTROL[COUNTENABLE]=1，且由于某种原因 PRU 不能取指令时，STALLCNT 开始以 PRU 时钟周期计数。其值总是小于，或等于 CYCLECNT 的值。

2.6 常量表

PRU 提供 32 个常量地址表 C0～C31， INTCTER0-INTCTER31 是常量表的调试接口，当 PRU 停止时，外部主模块读取 INTCTERn 即得到常量表 Cn 的值。通过指令 LBCO 或 SBCO 从常量表指向的地址与寄存器之间以簇发方式传递数据。指令格式举例如下：

```
LBCO      R2, C2, 5, 8   //从 C2+5 的地址读取 8 字节到 R2, R3。

SBCO      R2, C2, 5, 8   //将 R2, R3 的数据写到 C2+5 开始的地址。
```

表 6 PRU 常量表

Entry #	Region Pointed To	Value [31:0]
0	PRU INTC	0x00004000
1	Timer64P0	0x01C20000
2	I2C0	0x01C22000
3	PRU0/1 Local Data	0x00000000
4	PRU1/0 Local Data	0x00002000
5	MMC/SD	0x01C40000
6	SPI0	0x01C41000
7	UART0	0x01C42000
8	McASP0 DMA	0x01D02000
9	*RESERVED*	*0x01D06000*
10	*RESERVED*	*0x01D0A000*
11	UART1	0x01D0C000
12	UART2	0x01D0D000
13	USB0	0x01E00000
14	USB1	0x01E25000
15	UHPI Config	0x01E10000

Entry #	Region Pointed To	Value [31:0]
16	*RESERVED*	*0x01E12000*
17	I2C1	0x01E28000
18	EPWM0	0x01F00000
19	EPWM1	0x01F02000
20	*RESERVED*	*0x01F04000*
21	ECAP0	0x01F06000
22	ECAP1	0x01F07000
23	ECAP2	0x01F08000
24	PRU0/1 Local Data	0x00000n00, n = c24_blk_index[3:0]
25	McASP0 Control	0x01D00n00, n = c25_blk_index[3:0]
26	*RESERVED*	*0x01D04000*
27	*RESERVED*	*0x01D08000*
28	DSP RAM/ROM	0x11nnnn00, nnnn = c28_pointer[15:0]
29	EMIFa SDRAM	0x40nnnn00, nnnn = c29_pointer[15:0]
30	L3 RAM	0x80nnnn00, nnnn = c30_pointer[15:0]
31	EMIFb Data	0xC0nnnn00, nnnn = c31_pointer[15:0]

在性能上与指令 LBBO 和 SBBO 没有区别，利用常量表可以节省通用寄存器的使用。

0～23 号常量表提供的入口地址是固定的，24～31 号常量表的域可通过寄存器编程设置：

- C24[11:8]通过 CONTABBLKIDX0[3:0]设置；

- C25[11:8]通过 CONTABBLKIDX0[19:16]设置；
- C28[23:8]通过 CONTABPROPTR0[15:0]设置；
- C29[23:8]通过 CONTABPROPTR0[31:16]设置；
- C30[23:8]通过 CONTABPROPTR1[15:0]设置；
- C31[23:8]通过 CONTABPROPTR1[31:16]设置。

2.7 INTGPR0～31 调试通用寄存器

INTGPR0～31 与通用寄存器 R0～R31 对应，为外部主模块提供一个调试窗口。当 PRU 停止时，ARM/DSP 读/写 INTGPR0～31 直接读/写寄存器 R0～R31。

3 PRU 模块接口

3.1 PRU 事件/状态寄存器 R31

R31 是一个特殊的寄存器，读与写操作时的功能是不一样的。

写 R31 寄存器时，写事件号 0～31 到 R31_PRU_VEC[4:0]，同时设置 R31_PRU_VEC_VALID，将产生中断输出事件到中断控制器的 32~63 号系统事件。两个 PRU 输出的中断事件相“或”输出到同一个中断号。

表 7 写 R31 寄存器

位	名称	描述
31:6	RSV	Reserved
5	PRU_VEC_VALID	Valid strobe for vector output
4:0	PRU_VEC[4:0]	Vector output

读 R31 寄存器时，R31[29:0]反映 PRU 的输入管脚 PRU_R30[29:0]的状态。R31[31:30]是映射到中断控制器的 INTR_IN[0]和 INTR_IN[1]的状态。

表 8 读 R31 寄存器

位	名称	描述
31	PRU_INTR_IN[1]	PRU Host 1 interrupt from INTC
30	PRU_INTR_IN[0]	PRU Host 0 interrupt from INTC
29:0	PRU_R31_STATUS[29:0]	Status inputs from PRU*n*_R31[29:0]

3.2 通用输出寄存器 R30

每个 PRU 有 32 个独立的输出管脚 PRU0_R31[31:0]和 PRU1_R31[31:0]，写到寄存器 R30[31:0]的值直接输出到 PRUn_R31[31:0]管脚。

4 PRU 中断控制器

PRU 中断控制器支持 64 个系统事件，10 个中断通道，10 个主机中断。

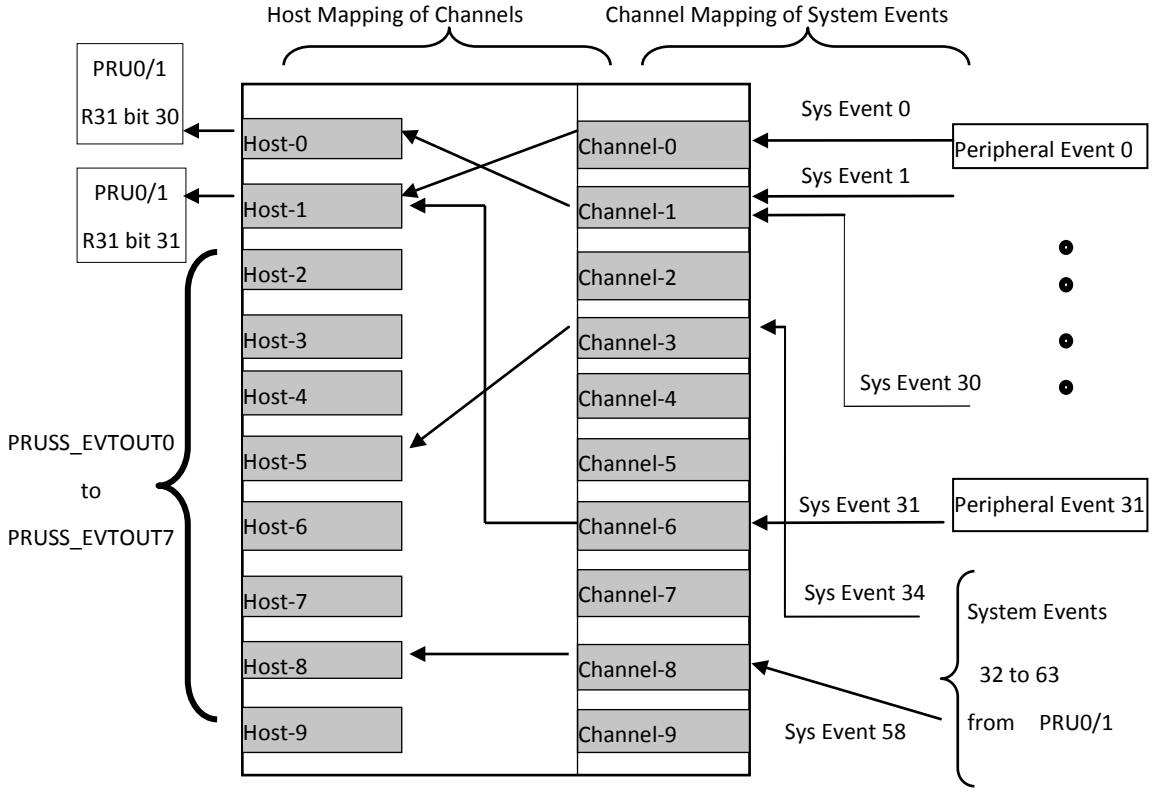


图 2 PRU 中断控制器框图

系统事件 0～31 为 32 个外部事件如表 9 所示，可以通过系统配置寄存器 CFGCHIP3 的第 3 位 PRUEVTSEL 在两组事件中进行选择，PRUSSEVTSEL=0 时选择第一列的 32 个外部系统事件，PRUSSEVTSEL=1 时选择第二列的 32 个外部系统事件；系统事件 32～63 由写 R31 产生。

表 9 PRUSS 系统中断事件

	PRUSSEVTSEL = 0	PRUSSEVTSEL = 1
事件号	**描述**	**描述**
0	Emulation Suspend Signal (Software Use Only)	Emulation Suspend Signal (Software Use Only)

1	ECAP0 Interrupt	Timer64P2_T12CMPEVT0
2	ECAP1 Interrupt	Timer64P2_T12CMPEVT1
3	Timer64P0 Event Out 12	Timer64P2_T12CMPEVT2
4	ECAP2 Interrupt	Timer64P2_T12CMPEVT3
5	McASP0 TX DMA Request	Timer64P2_T12CMPEVT4
6	McASP0 RX DMA Request	Timer64P2_T12CMPEVT5
7	McBSP0 TX DMA Request	Timer64P2_T12CMPEVT6
8	McBSP0 RX DMA Request	Timer64P2_T12CMPEVT7
9	McBSP1 TX DMA Request	Timer64P3_T12CMPEVT0
10	McBSP1 RX DMA Request	Timer64P3_T12CMPEVT1
11	SPI0 Interrupt 0	Timer64P3_T12CMPEVT2
12	SPI1 Interrupt 0	Timer64P3_T12CMPEVT3
13	UART0 Interrupt	Timer64P3_T12CMPEVT4
14	UART1 Interrupt	Timer64P3_T12CMPEVT5
15	I2C0 Interrupt	Timer64P3_T12CMPEVT6
16	I2C1 Interrupt	Timer64P3_T12CMPEVT7
17	UART2 Interrupt	Timer64P0_T12CMPEVT0 or Timer64P0_T12CMPEVT1 or Timer64P0_T12CMPEVT2 or Timer64P0_T12CMPEVT3 or Timer64P0_T12CMPEVT4 or Timer64P0_T12CMPEVT5 or Timer64P0_T12CMPEVT6 or Timer64P0_T12CMPEVT7
18	MMCSD0 Interrupt 0	Timer64P2 Event Out 12
19	MMCSD0 Interrupt 1	Timer64P3 Event Out 12
20	USB0 (USB2.0 HS OTG) Subsystem Interrupt Request (aggregated from subsystem's INTD)	Timer64P1 Event Out 12
21	USB1 (USB1.1 FS OHCI) Subsystem IRQ Interrupt	UART1 Interrupt
22	Timer64P0 Event Out 34	UART2 Interrupt
23	ECAP0 input (output from mux)	SPI0 Interrupt 0

24	EPWM0 Interrupt	EPWM0 Interrupt
25	EPWM1 Interrupt	EPWM1 Interrupt
26	SATA Interrupt	SPI1 Interrupt 0
27	EDMA3_0_CC0_INT2 (region 2)	GPIO Bank 0 Interrupt
28	EDMA3_0_CC0_INT3 (region 3)	GPIO Bank 1 Interrupt
29	UHPI CPU_INT	McBSP0 TX DMA Request
30	EPWM0TZ Interrupt or EPWM1TZ Interrupt	McBSP0 RX DMA Request
31	McASP0 TX Interrupt or McASP0 RX Interrupt	McASP0 TX Interrupt or McASP0 RX Interrupt

10 个通道可以由任意 64 个系统事件映射，可以多个系统事件映射到一个通道，但不要将一个系统事件映射到多个通道。

10 个主机中断与 10 个通道之间可以任意映射，可以多个通道映射到一个主机中断，但不要将一个通道映射到多个主机中断，推荐按 x 号通道映射到 x 号主机中断方式映射。

主机中断 0 输出到 R31.b30，主机中断 1 输出到 R31.b31。主机中断 2～9 接输出 PRUSS 到 ARM 和 DSP 的中断控制器的系统事件 PRUSS_EVTOUT0～7。PRU 不支持中断向量表，产生的 0，1 主机中断可用来唤醒 PRU，或为 PRU 软件提供状态查询。

PRU 中断控制器寄存器说明请参考文献[2]。

5 PRU 编程

5.1 汇编工具 PASM

PRU 只支持汇编编程，可用任意的文本编辑器编写源代码。PASM 是 PRU 的命令行汇编器，其语法为：

pasm [-bcmldz] [-Dname=value] [-Cname] InFile [OutFileBaseName]

参数含义如表 10 所示：

表 10 PASM 命令表

命令选项	输出文件格式	输出文件名
-b	小端二进制文件	myprog.bin

-c	C 数组，默认名为 PDSPcode[]* ，可用-Cname 指定数组名	myprog_bin.h
-m	镜像文件，每行为一个 32bit 的指令码	myprog.img
-l	包括源代码与产生的指令码的交叉列表文件	myprog.lst
-d	pView debugger 输出文件(opcodes with source and label info)	myprog.dbg
-z	编译时输出调试信息	
-C	指定输出 C 数组名	
-D	常量定义，命令行的常量定义不会覆盖代码中的常量定义	

PASM 将汇编代码转换成一整块可执行的二进制数据用于加载，代码从指令内存的首地址开始排放，编译时没有链接过程，没有内存映射，所以没有段，从而也没有段指示符。输出格式可以为纯二进制文件，C 语言形式的数组，Hex 格式文件。

5.2 PRU 指令集

指令按功能分为四类：数据搬移指令，算术运算指令，逻辑操作指令，程序流控指令。指令的语法规则参考[3]。

表 11 PRU 指令集

数据搬移指令	算术运算指令	逻辑操作指令	程序流控指令
MOV	ADD	LSL	JMP
LDI	ADC	LSR	JAL
MVIx	SUB	AND	CALL
LBBO	SUC	OR	RET
SBBO	RSB	XOR	QBGT
LBCO	RSC	NOT	QBLT
SBCO		MIN	QBLE
LFC		MAX	QBEQ
STC		CLR	QBNE
ZERO		SET	QBA
		SCAN	QBBS
		LMBD	QBBC
			WBS
			WBC

			HALT
			SLP

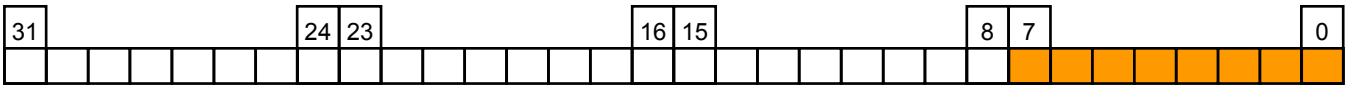
除了访问外部内存的指令，其它所有 PRU 指令都是单周期指令。指令中寄存器名加上后缀表示寄存器中的位、字节、半字，这种灵活的寄存器访问方式可以将数据包，结构体分解成小的数据单位处理。

表 12　寄存器位域表示方式

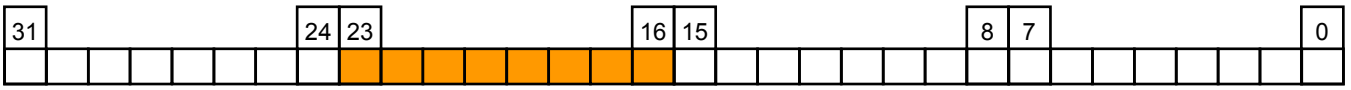
后缀	N 的范围	描述
.wn	0 to 2	选择以字节偏移的第 n 个 16 bit 域
.bn	0 to 3	选择父域的以字节偏移的第 n 个 8 bit 域
.tn	0 to 31	选择父域的第 n 个 bit

举例说明这种寄存器访问方式：

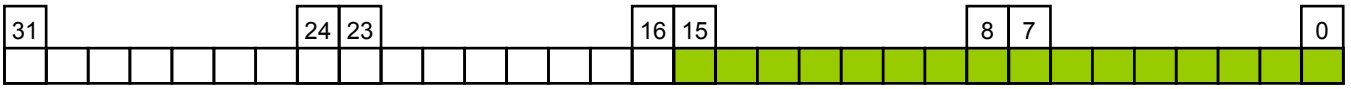
R0.b0



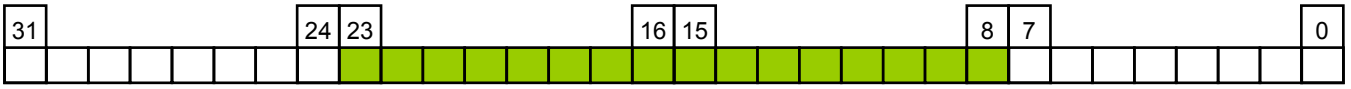
R0.b2



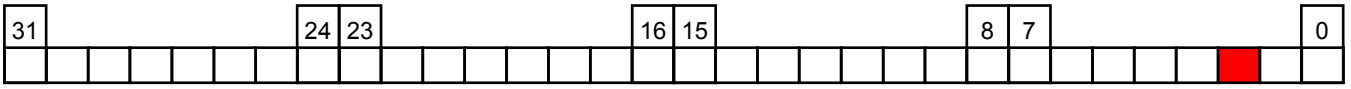
R0.w0



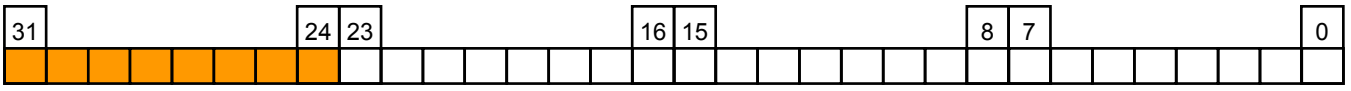
R0.w1



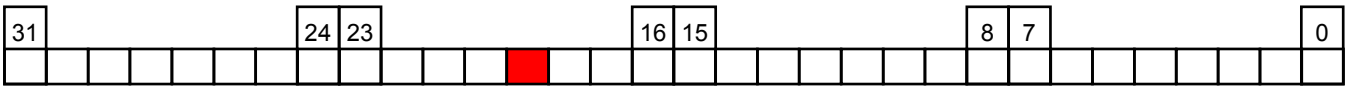
R0.t2



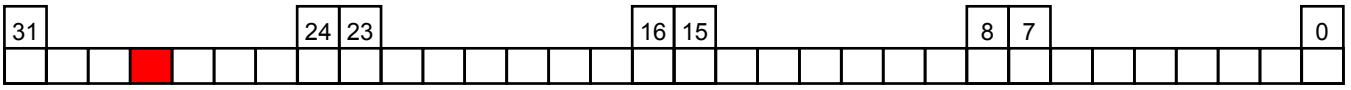
R0.w2.b1=r0.b3



r0.w1.b1.t3 = r0.b2.t3 = r0.t19



r0.w2.t12 = r0.t28



5.3 汇编操作符

PASM 支持 4 类汇编操作符：Hash 命令，指示语句，标签，注释。

5.3.1 Hash 命令

控制汇编器的预处理，包括#include，#define，#undef，#ifdef，#ifndef，#else， #endif，#error，与 C 语言的预处理命令用法类似。

5.3.2 指示符

指示符以“.”开头，单独占一行，不一定要放在一行的开头，后面可以跟注释语句。

表 13 PASM 汇编指示符

命令	描述
.origin	Set start of next assembly statement
.entrypoint	Only used for debugger, specifies starting address
.setcallreg	Specified 16-bit register field for storing return pointer
.macro, .mparam, .endm	Define assembler macros
.struct, .ends, .u32, .u16, .u8	Define structure types for easier register allocation
.assign	Map defined structure into PRU register file
.enter	Create and enter new variable scope
.leave	Leave a specific variable scope
.using	Use a previously created and left scope

5.3.3 标签

在源代码中加入标签名+“：”，用来标示源代码的地址，当被指令引用时，直接替换成标签所在的源代码地址。标签命名不能以数字开头，可以单独为一行，也可以与指令，注释在同一行。

5.3.4 注释

以“//”开始，以行结束符结束，可以出现在代码的任何位置。

6 PRU 开发包

TI 提供的 PRU 开发包里[4]，提供了大量的实例源码，目录结构如图 3 所示。

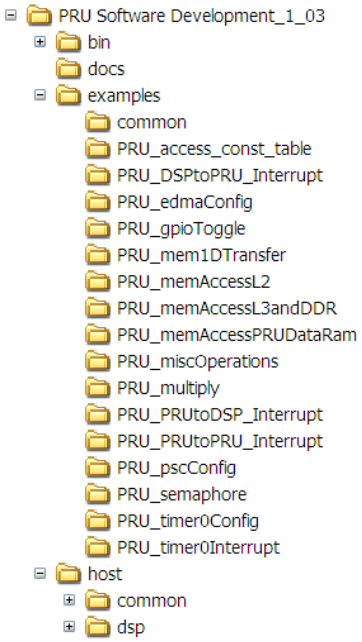


图 3 PRU 开发包目录结构

bin 目录下是 Linux 与 Window 版本的 PASM 工具；docs 目录下的 readme 文件列出了 TI wiki 上的 PRU 帮助说明链接；example 目录下为各种例程源码，每个例程包括 CCSV3.3 的 DSP 工程，和以“.p 和.hp”结尾的 PRU 代码；host 目录下是两个单独的 PRU 代码下载例程，以文件形式的和以数组形式的。

7 PRU 代码下载与运行

先通过 PSC 使能 PRU 模块时钟，PRU 空间才可以被访问。

PASM 编译后 PRU 代码通过 ARM/DSP 从全局地址空间加载到 PRU 的指令空间后，ARM/DSP 通过全局空间地址将 PRUCONTROL 寄存器的 CONTERENABLE 和 ENABLE 置 1，PRU 即开始从 0 地址开始运行。在运行过程中 ARM/DSP 可以通过 CONTROL 控制 PRU 的暂停，运行，复位，及复位后的运行地址等；暂停时可以通过 PRU 的调试寄存器观察 PRU 的寄存器状态，也可以通过调试寄存器修改 PRU 寄存器的值。

8 PRU 调试

目前在 CCSV5 里可以用仿真器连接 PRU 查看寄存器，内存。但是没有图形界面的 PRU 调试工具，不能在代码中加入断点这样的方式调试。可以通过以下几种方法进行间接调试。

8.1 加入“打印”信息

在 PRU 代码中增加调试代码，根据程序运行状态在内存里写入相应的值，暂停 PRU 后，通过查看内存值来获取程序的运行状态，也可以在 DSP/ARM 的代码里加入调试代码检测内存值的变化。

8.2 触发中断

在 PRU 代码中通过写 R31 寄存器给 DSP/ARM 发送中断，这种方法减轻 DSP/ARM 的负荷，尤其在 DSP/ARM 软件与 PRU 软件一起调试时更为有效。

8.3 通过输出寄存器 R30 输出状态

在代码中写 R30 设置相应 PRU 输出管脚的状态，用示波器或万用表观察管脚状态以获知程序运行状态。

8.4 加入空循环反复调试

在代码中加入 while(1)，通过查看 PRU 的程序指针是否停在空循环处，可以知道程序运行流程是否正确，查看 PRU 的寄存器和内存判断运行结果是否正确。在 PASM 编译产生的列表文件里有代码的对应的程序指针值。

参考文献：

1. http://processors.wiki.ti.com/index.php/Programmable_Realtime_Unit#PRU_Control_Register_.280x0000.29
2. http://processors.wiki.ti.com/index.php/PRU_Interrupt_Controller
3. http://processors.wiki.ti.com/index.php/PRU_Assembly_Instructions
4. http://www.ti.com/tool/sprc940
5. http://processors.wiki.ti.com/index.php/PASM_Syntax_Highlighting

重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改， 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内，且 TI 认为 有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定，否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应 用相关的风险，客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予 的直接或隐含权限作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息，不能构成从 TI 获得使用这些产品或服 务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可，或是 TI 的专利权或其它 知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分，仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况 下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时，如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分，则会失去相关 TI 组件 或服务的所有明示或暗示授权，且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意，尽管任何应用相关信息或支持仍可能由 TI 提供，但他们将独力负责满足与其产品及在其应用中使用 TI 产品 相关的所有法律、法规和安全相关要求。客户声明并同意，他们具备制定与实施安全措施所需的全部专业技术和知识，可预见 故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因 在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中，为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特 有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此，此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III（或类似的生命攸关医疗设备）的授权许可，除非各方授权官员已经达成了专门管控此类使 用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同 意，对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用，其风险由客户单独承担，并且由客户独 力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品，这些产品主要用于汽车。在任何情况下，因使用非指定产品而无法达到 ISO/TS16949 要求，TI不承担任何责任。

	产品		应用
数字音频	www.ti.com.cn/audio	通信与电信	www.ti.com.cn/telecom
放大器和线性器件	www.ti.com.cn/amplifiers	计算机及周边	www.ti.com.cn/computer
数据转换器	www.ti.com.cn/dataconverters	消费电子	www.ti.com/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com/energy
DSP - 数字信号处理器	www.ti.com.cn/dsp	工业应用	www.ti.com.cn/industrial
时钟和计时器	www.ti.com.cn/clockandtimers	医疗电子	www.ti.com.cn/medical
接口	www.ti.com.cn/interface	安防应用	www.ti.com.cn/security
逻辑	www.ti.com.cn/logic	汽车电子	www.ti.com.cn/automotive
电源管理	www.ti.com.cn/power	视频和影像	www.ti.com.cn/video
微控制器 (MCU)	www.ti.com.cn/microcontrollers		
RFID 系统	www.ti.com.cn/rfidsys		
OMAP应用处理器	www.ti.com/omap		
无线连通性	www.ti.com.cn/wirelessconnectivity	德州仪器在线技术支持社区	www.deyisupport.com

邮寄地址： 上海市浦东新区世纪大道 1568 号，中建大厦 32 楼 邮政编码： 200122