

(2) 扫描类型：对于编码和解码，位处理器并没有考虑它应该写或读系数存储器的哪个位置的系数。重新排序它们基于指定扫描类型后，系数存储器接口写或读系数。

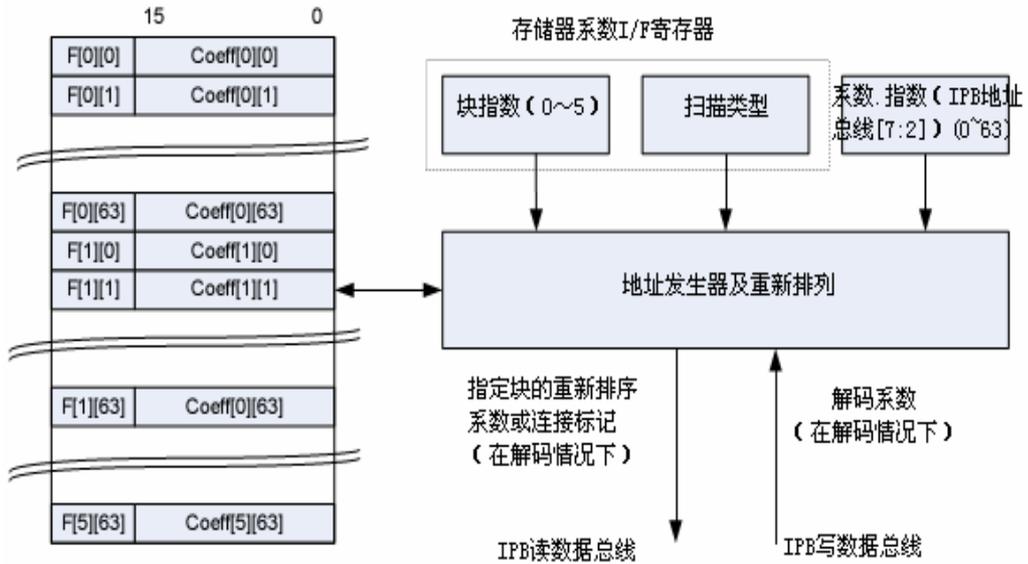


图 21-36 存储器系数访问

4. 编码器操作

在编码情况下，量化系数不需要任何调整即可写入系数缓冲器接口的内部存储器中。当处理器读取系数时，它们被重新排序并标记显示相应的系数有一个非零值被发送到位处理器来阻止是否有一个系数被编码。通过简单的计算非零位的数量，位处理器可以知道非零的系数的数量。因此，位处理器只读非零的系数。如图 21-37 和图 21-38 所示，显示 MPEG-4 编码案例和 H.264 编码案例。

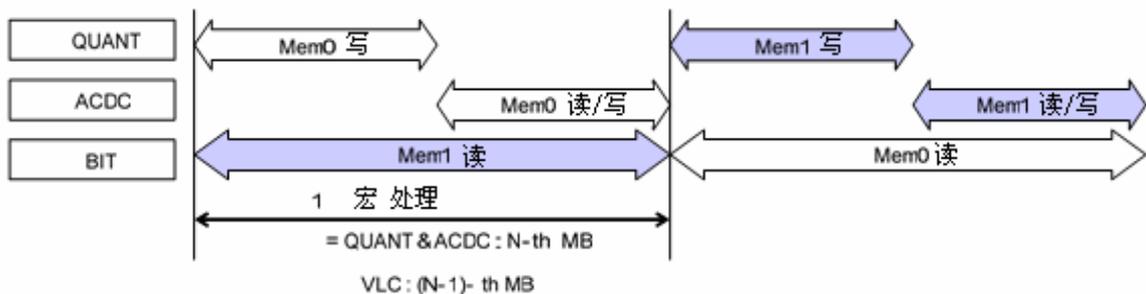


图 21-37 MPEG-4 编码案例

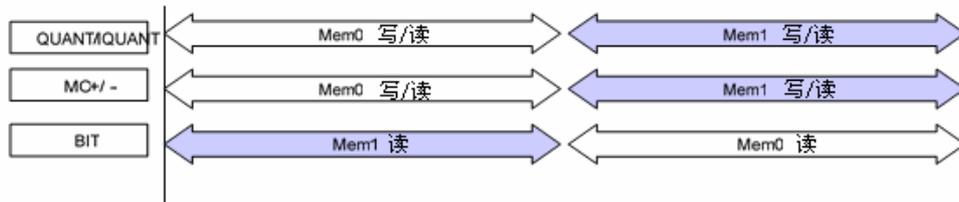


图 21-38 H.264 编码案例

5. 解码器操作

位处理器没有重新排序反向 zig-zag 扫描来写入解码系数。当逆量化模块从系数缓冲器接口读取系数时，执行重新排序的过程。如果系数相应标记为零，子模块如量化和 AC/DC 预测处理为零。如图 21-39 和图 21-40 所示，显示 MPEG-4 编码案例和 H.264 编码案例。

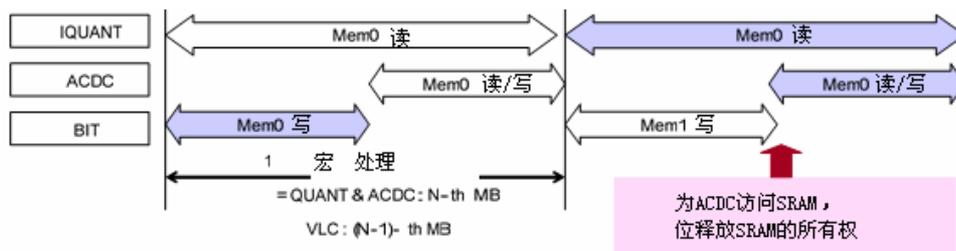


图 21-39 MPEG-4 解码案例

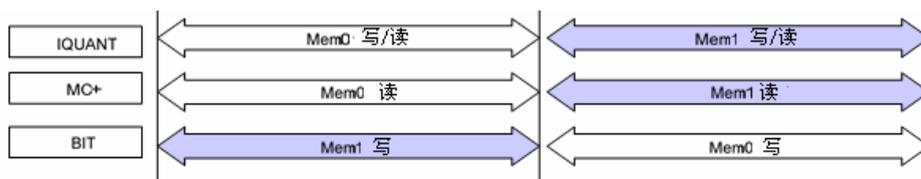


图 21-40 H.264 解码案例

注：在图 12-86，12-87，12-88 和 12-89 中，MC+意味重建，MC-意味处理余量。

6. 宏控制器

FIMV-MFC V1.0 有一个复杂而大量的管线，性能高。位处理器完全控制它是不合适的。因此，通过位处理器，FIMV-MFC V1.0 嵌入了宏控制器基于管线的配置上所有的视频编解码器的子模块。这个计划降低了位处理器的加载和保证 IP 的可编程。

视频编解码器编码或解码一个宏，位处理器配置编解码器的管线是如何构成的。如果为编码/解码一

个宏，所有进程被完成，宏控制器显示它的完成。

总之，位处理器配置，为当前宏处理，其中子模块启动，并宏控制器控制相应的子模块基于配置之上。如图 21-41 所示，显示宏控制器的连接。

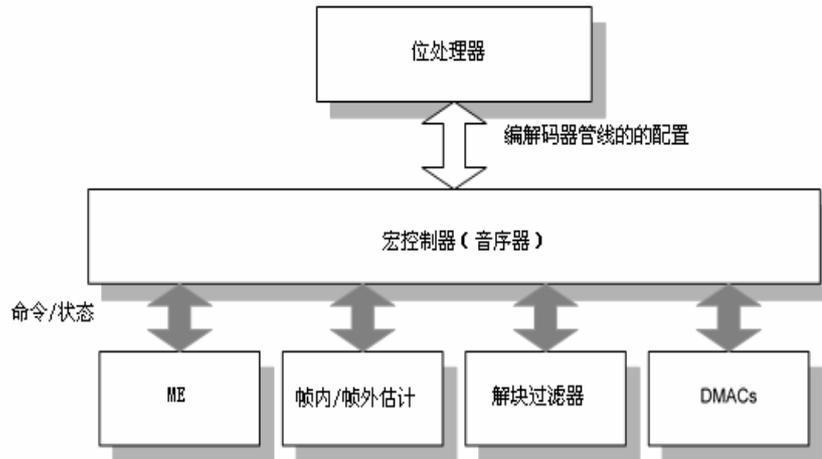


图 21-41 宏控制器的连接

21.12 FIMV-MFC V1.0 编程模式（特殊功能寄存器）

FIMV-MFC V1.0 通过 APB 总线接口与一个主机处理器相通。如表 21-8 所示，说明区域的地址图可以通过 APB 被存取。

表 21-8 内部寄存器地址图

PADDR[11:9]	模块	描述
3' b000	位处理器的主机接口	在正常运作中，主机控制器无法访问这些寄存器
3' b001	宏控制器（音序器）	
3' b010	系数存储器接口	
3' b011	分割滤波器	
3' b100	运动估值	
3' b101	帧间估计	
3' b110	VC-1 定标器	
3' b111	S/W RESET	

1. 主机接口寄存器

位处理器寄存器分为两类。

地址 0x000~0x0FC (64 位寄存器地址空间) 是 H/W 寄存器。这些寄存器有重置值, 并且它们的功能是固定的 (无可配置)。地址 0x100~0x1FC (64 位寄存器) 是通用 S/W 寄存器。它们没有重置值并通过位固件可配置的。它们被用来作为接口在主机和位处理器之间。高位 32 位寄存器 (地址 0x100~0x17C) 用来作为静态参数。这些寄存器的含义或功能没有被改变, 用于所有种类运动命令 (SEQ_INIT, SEQ_END, PICTURE_RUN), 并且适用于所有命令。低位 32 位寄存器 (地址 0x180~0x1FC) 用来作为暂时命令自变量。这些寄存器的含义或功能被改变, 用于每个运动命令。

2. 主机接口寄存器的概要

如表 21-9 所示, 显示了位处理器共同寄存器概要说明。

表 21-9 位处理器共同寄存器概要 (基址=0x7E002000)

地址	类型	宽度	复位值	名称	描述
BASE+0x000	写		0	CodeRun	位运动启动
BASE+0x004	写		0	CodeDownLoad	代码下载数据寄存器
BASE+0x008	写		0	HostIntReq	主机中断请求至位
BASE+0x00C	写		0	BitIntClear	位中断清除
BASE+0x010	读		0	BitIntSts	位中断状态
BASE+0x014	写		0	BitCodeReset	位代码重置
BASE+0x018	读		0	BitCurPc	位通用 PC
BASE+0x100	读/写		N/A	CodeBufAddr	代码表 SDRAM 地址
BASE+0x104	读/写		N/A	WorkBufAddr	工作缓冲 SDRAM 地址
BASE+0x108	读/写		N/A	ParaBufAddr	自变量/返回参数缓冲 SDRAM 地址
BASE+0x10C	读/写		N/A	BitStreamCtrl	位流缓冲控制
BASE+0x110	读/写		N/A	FrameMemCtrl	帧存储器控制
BASE+0x114	读/写		N/A	DecFuncCtrl	解码器功能控制 1
BASE+0x11C	读/写		N/A	BitWorkBufCtrl	工作 Buf 控制 2
BASE+0x120	读/写		N/A	BitStreamRdPtr0	位流缓冲器读运动指数 0 的地址

BASE+0x124	读/写		N/A	BitStreamWrPtr0	位流缓冲器写运动指数 0 的地址
BASE+0x128	读/写		N/A	BitStreamRdPtr1	位流缓冲器读运动指数 1 的地址
BASE+0x12C	读/写		N/A	BitStreamWrPtr1	位流缓冲器写运动指数 1 的地址
BASE+0x130	读/写		N/A	BitStreamRdPtr2	位流缓冲器读运动指数 2 的地址
BASE+0x134	读/写		N/A	BitStreamWrPtr2	位流缓冲器写运动指数 2 的地址
BASE+0x138	读/写		N/A	BitStreamRdPtr3	位流缓冲器读运动指数 3 的地址
BASE+0x13C	读/写		N/A	BitStreamWrPtr3	位流缓冲器写运动指数 3 的地址
BASE+0x140	读/写		N/A	BitStreamRdPtr4	位流缓冲器读运动指数 4 的地址
BASE+0x144	读/写		N/A	BitStreamWrPtr4	位流缓冲器写运动指数 4 的地址
BASE+0x148	读/写		N/A	BitStreamRdPtr5	位流缓冲器读运动指数 5 的地址
BASE+0x14C	读/写		N/A	BitStreamWrPtr5	位流缓冲器写运动指数 5 的地址
BASE+0x150	读/写		N/A	BitStreamRdPtr6	位流缓冲器读运动指数 6 的地址
BASE+0x154	读/写		N/A	BitStreamWrPtr6	位流缓冲器写运动指数 6 的地址
BASE+0x158	读/写		N/A	BitStreamRdPtr7	位流缓冲器读运动指数 7 的地址
BASE+0x15C	读/写		N/A	BitStreamWrPtr7	位流缓冲器写运动指数 7 ³ 的地址
BASE+0x160	读		N/A	BusyFlag	处理器忙碌标记
BASE+0x164	读/写		N/A	RunCommand 4	运动命令
BASE+0x168	读/写		N/A	RunIndex	运动进程指数
BASE+0x16C	读/写		N/A	RunCodStd	运动编解码器标准
BASE+0x170	读/写		N/A	IntEnable	中断启动
BASE+0x174	读/写		N/A	IntReason	中断原因
为内部使用的保护					
BASE+0x180 ~ BASE + 1D8	读/写		N/A	Command I/O Reg	命令 I/O 寄存器

¹这个控制寄存器的功能是由主机处理器新增提供一个扩展控制配置。

²这个缓冲控制寄存器的工作是有主机处理器新增的支持缓冲配置的工作。

³新增的从 0x140 到 0x15C 寄存器，可支持最多 8 种实例。

⁴为检查 F/W 版本的一个新命令，用来新增正好的安排。

如表 21-10 所示，显示了 DEC_SEQ_INIT 参数寄存器的概要说明。

表 21-10 DEC_SEQ_INIT 参数寄存器概要

DEC_SEQ_INIT				
	地址	类型	名称	描述
输入 自变量	BASE+0x180	读/写	CMD_DEC_SEQ_BIT_BUF_START	位流缓冲器地址
	BASE+0x184	读/写	CMD_DEC_SEQ_BIT_BUF_SIZE	位流缓冲器大小
	BASE+0x188	读/写	CMD_DEC_SEQ_OPTION5	解码序列的选择
	BASE+0x18C	读/写	CMD_DEC_SEQ_PRO_BUF	进程缓冲器地址
	BASE+0x190	读/写	CMD_DEC_SEQ_TMP_BUF_1	暂时缓冲器 1 地址
	BASE+0x194	读/写	CMD_DEC_SEQ_TMP_BUF_2	暂时缓冲器 2 地址
	BASE+0x198	读/写	CMD_DEC_SEQ_TMP_BUF_3	暂时缓冲器 3 地址
	BASE+0x19C	读/写	CMD_DEC_SEQ_TMP_BUF_4	暂时缓冲器 4 地址
	BASE+0x1A0	读/写	CMD_DEC_SEQ_TMP_BUF_5	暂时缓冲器 5 地址
	BASE+0x1A4	读/写	CMD_DEC_SEQ_START_BYTE	有效流数据的起始字节
输出返 回	BASE+0x1C0	读	RET_DEC_SEQ_SUCCESS	命令执行结果的状态
	BASE+0x1C4	读	RET_DEC_SEQ_SRC_SIZE	解码初始图大小
	BASE+0x1C8	读	RET_DEC_SEQ_SRC_F_RATE	解码初始帧速率
	BASE+0x1CC	读	RET_DEC_SEQ_FRAME_NEED	必须的最低解码帧缓冲器
	BASE+0x1D0	读	RET_DEC_SEQ_FRAME_DELAY	最大显示帧缓冲器延迟
	BASE+0x1D4	读	RET_DEC_SEQ_INFO ⁷	解码序列信息

⁵ 表现文件播放模式的位，动态缓冲器分配启动被添加这个寄存器。

⁶ 从 18C 到 1A0，这个寄存器被添加提供一种用来工作缓冲器配置的方式。

⁷ 表现附件 J 指示的位，被添加这个寄存器。

如表 21-11 所示，显示了 ENC_SEQ_INIT 参数寄存器的概要说明。

表 21-11 ENC_SEQ_INIT 参数寄存器概要

ENC_SEQ_INIT				
	地址	类型	名称	描述
	BASE+0x180	读/写	CMD_ENC_SEQ_BIT_BUF_START	流缓冲器地址

输入 自变量	BASE+0x184	读/写	CMD_ENC_SEQ_BIT_BUF_SIZE	流缓冲器大小
	BASE+0x188	读/写	CMD_ENC_SEQ_OPTION	编码序列选择
	BASE+0x18C	读/写	CMD_ENC_SEQ_COD_STD	编码译码标准
	BASE+0x190	读/写	CMD_ENC_SEQ_SRC_SIZE	编码初始帧大小
	BASE+0x194	读/写	CMD_ENC_SEQ_SRC_F_RATE	编码初始帧速率
	BASE+0x198	读/写	CMD_ENC_SEQ_MP4_PARA	编码的 MPEG4 参数
	BASE+0x19C	读/写	CMD_ENC_SEQ_263_PARA	编码的 H. 263 参数
	BASE+0x1A0	读/写	CMD_ENC_SEQ_264_PARA	编码的 H. 264 参数
	BASE+0x1A4	读/写	CMD_ENC_SEQ_SLICE_MODE	编码片模式
	BASE+0x1A8	读/写	CMD_ENC_SEQ_GOP_NUM	编码 GOP 数量
	BASE+0x1AC	读/写	CMD_ENC_SEQ_RC_PARA	编码率控制参数
	BASE+0x1B0	读/写	CMD_ENC_SEQ_RC_BUF_SIZE	编码率控制缓冲器大小
	BASE+0x1B4	读/写	CMD_ENC_SEQ_INTRA_MB	编码内部 MB 刷新数量
BASE+0x1B8	读/写	CMD_ENC_SEQ_FMO	在 H. 264 Enc 中 FMO 配置	
	BASE+0x1D0	读/写	CMD_ENC_SEQ_TMP_BUF_1	临时缓冲器 1 地址
	BASE+0x1D4	读/写	CMD_ENC_SEQ_TMP_BUF_2	临时缓冲器 2 地址
	BASE+0x1D8	读/写	CMD_ENC_SEQ_TMP_BUF_3	临时缓冲器 2 地址
	BASE+0x1DC	读/写	CMD_ENC_SEQ_TMP_BUF_4	临时缓冲器 4 地址
输出返回	BASE+0x1C0	读	RET_ENC_SEQ_SUCCESS	命令执行结果状态

如表 21-12 所示，显示了 DEC_PIC_RUN 参数寄存器的概要说明。

表 21-12 DEC_PIC_RUN 参数寄存器概要

DEC_PIC_RUN				
	地址	类型	名称	描述
输入	BASE+0x180	读/写	CMD_DEC_PIC_ROT_MODE	显示帧后旋转模式
自变量	BASE+0x184	读/写	CMD_DEC_PIC_ROT_ADDR_Y	后旋转帧存储在 Y 地址

	BASE+0x188	读/写	CMD_DEC_PIC_ROT_ADDR_CB	后旋转存储在 CB 地址
	BASE+0x18C	读/写	CMD_DEC_PIC_ROT_ADDR_CR	后旋转帧存储在 CR 地址
	BASE+0x190	读/写	CMD_DEC_PIC_DBK_ADDR_Y	解块帧存储在 Y 地址
	BASE+0x194	读/写	CMD_DEC_PIC_DBK_ADDR_CB	解块帧存储在 CB 地址
	BASE+0x198	读/写	CMD_DEC_PIC_DBK_ADDR_CR	解块帧存储在 CR 地址
	BASE+0x19C	读/写	CMD_DEC_PIC_ROT_STRIDE	后旋转帧的步幅
	BASE+0x1A8	读/写	CMD_DEC_PIC_CHUNK_SIZE	帧块大小
	BASE+0x1AC	读/写	CMD_DEC_PIC_BB_START	图片流缓冲器的 4 字节排队初始地址
	BASE+0x1B0	读/写	CMD_DEC_PIC_START_BYTE	有效流数据的初始字节
输出返回	BASE+0x1C0	读	RET_DEC_PIC_FRAME_NUM	解码帧的数量
	BASE+0x1C4	读	RET_DEC_PIC_IDX	显示帧的指数
	BASE+0x1C8	读	RET_DEC_PIC_ERR_MB_NUM	在解码图片中错误 MB 数量
	BASE+0x1CC	读	RET_DEC_PIC_TYPE	解码图片类型
	BASE+0x1D8	读	RET_DEC_PIC_SUCCESS	命令执行结果的状态

如表 21-13 所示，显示了 ENC_PIC_RUN 参数寄存器的概要说明。

表 21-13 ENC_PIC_RUN 参数寄存器概要

ENC_PIC_RUN				
	地址	类型	名称	描述
输入自变量	BASE+0x180	读/写	CMD_ENC_PIC_SRC_ADDR_Y	输入初始帧缓冲器 Y 的 SDRAM 地址
	BASE+0x184	读/写	CMD_ENC_PIC_SRC_ADDR_CB	输入初始帧缓冲器 CB 的 SDRAM 地址
	BASE+0x188	读/写	CMD_ENC_PIC_SRC_ADDR_CR	输入初始帧缓冲器 CR 的 SDRAM 地址
	BASE+0x18C	读/写	CMD_ENC_PIC_QS	编码图片量化的步幅
	BASE+0x190	读/写	CMD_ENC_PIC_ROT_MODE	输入帧前旋转模式
	BASE+0x194	读/写	CMD_ENC_PIC_OPTION	编码图片选项
输出返回	BASE+0x1C0	读	RET_ENC_PIC_FRAME_NUM	编码帧数量
	BASE+0x1C4	读	RET_ENC_PIC_TYPE	编码图片类型

	BASE+0x1C8	读	RET_ENC_PIC_IDX	重建帧指数
	BASE+0x1CC	读	RET_ENC_PIC_SLICE_NUM	编码片的图片数量

如表 21-14 所示，显示了 SET FRAME BUFFER 参数寄存器的概要说明。

表 21-14 SET FRAME BUFFER 参数寄存器概要

SET_FRAME_BUFFER				
	地址	类型	名称	描述
输入自变量	BASE+0x180	读/写	CMD_SET_FRAME_BUF_NUM	编解码器使用帧缓冲器的数量
	BASE+0x184	读/写	CMD_SET_FRAME_BUF_STRIDE	帧缓冲线的步幅
输出返回				

如表 21-15 所示，显示了 ENC HEADER 参数寄存器的概要说明。

表 21-15 ENC HEADER 参数寄存器概要

ENC_HEADER				
	地址	类型	名称	描述
输入自变量	BASE+0x180	读/写	CMD_ENC_HEADER_CODE	编码标题代码
输出返回				

如表 21-16 所示，显示了 DEC PARA SET 参数寄存器的概要说明。

表 21-16 DEC PARA SET 参数寄存器概要

DEC_PARA_SET				
	地址	类型	名称	描述
输入自变量	BASE+0x180	读/写	CMD_DEC_PARA_SET_TYPE	序列/图片参数设置类型
	BASE+0x184	读/写	CMD_DEC_PARA_SET_SIZE	序列/图片参数设置 Rbsp 字节大小
输出返回				

如表 21-17 所示，显示了 ENC PARA SET 参数寄存器的概要说明。

表 21-17 ENC PARA SET 参数寄存器概要

ENC_PARA_SET				
	地址	类型	名称	描述
输入自变量	BASE+0x180	读/写	CMD_ENC_PARA_SET_TYPE	序列/图片参数设置类型
输出返回	BASE+0x1C0	读	RET_ENC_PARA_SET_SIZE	编码序列/图片参数设置 Rbsp 字节大小

如表 21-18 所示，显示了 GET F/W VER 参数寄存器的概要说明。

表 21-18 GET F/W VER 参数寄存器概要

GET_F/W_VER				
	地址	类型	名称	描述
输入自变量				
输出返回	BASE+0x1C0	读	RET_GET_FW_VER	返回版本代码有以下格式： [31: 16]:产品号 (0xF202) [15:0]M. m. rr 的显示号 (0xMmrr)

下面我们主要对位处理器寄存器的描述。

CodeRun (0x000)

位	名称	类型	功能	复位值
0	CodeRun	写	0- 位处理器停止执行 1- 位处理器开始执行	

CodeDownload (0x004)

位	名称	类型	功能	复位值
15: 0	CodeData	写	16 位代码下载数据	0
28: 16	CodeAddr	写	13 位代码下载地址 位代码字地址 (16 位地址) *当前的设计有 4K 代码字空间 (8KB)。 因此 CodeAddr[12: 0]必须少于 4095	0