

			到地址[ParaBufAddr]的 SDRAM 缓冲器。	
--	--	--	------------------------------	--

### CMD\_SET\_FRAME\_BUF\_STRIDE (0x184)

位	名称	类型	功能	指令
10:0	LineStride	读/写	图片帧存储器的线幅度偏移幅度数以字节计数	SET_FRAME_BUF

### CMD\_ENC\_HEADER\_CODE (0x180)

位	名称	类型	功能	指令
2:0	HeaderCode	读/写	有效标头代码 MPEG4, 3' b000 - VOL 标头 3' b001 - VOS 标头 3' b010 - VIS 标头 H. 264, 3' b000 - SPS rbsp 3' b001 - PPS rbsp H. 263, ENC_HEADER 指令被忽略	ENC_HEADER

### CMD\_DEC\_PARA\_SET\_TYPE (0x180)

位	名称	类型	功能	指令
0	DecParaSetType	读/写	参数设置类型 0 - 次序参数设置 1 - 图片参数设置	DEC_PARA_SET

### CMD\_DEC\_PARA\_SET\_SIZE (0x184)

位	名称	类型	功能	指令
8:0	DecParaSetSize	读/写	顺序/图片参数设置 Rbsp 字节大小 最大允许 Rbsp 大小为 511 字节。	DEC_PARA_SET

### CMD\_ENC\_PARA\_SET\_TYPE (0x180)

位	名称	类型	功能	指令
0	EncParaSetType	读/写	参数设置类型 0 - 次序参数设置 1 - 图片参数设置	ENC_PARA_SET

### RET\_ENC\_PARA\_SET\_SIZE (0x1C0)

位	名称	类型	功能	指令
8:0	EncParaSetSize	读/写	有效顺序/图片参数设置 Rbsp 字节大小	ENC_PARA_SET

## 21.13 BIT 处理器操作

### 21.13.1. BIT 处理寄存器描述

根据主机接口寄存器的用途，可分成三类：

#### (1) BIT处理器控制寄存器

该种类的主机接口寄存器用于将更新或者显示BIT处理器状态送到主机处理器。在导入期间，大部分用于初始化BIT。

#### (2) BIT处理器通用寄存器

该类的主机接口寄存器用来存储所有的通用变量。一般地，所有的缓冲器地址和一些通用选项将安全地存储在这些寄存器中。

#### (3) BIT处理器指令I/O寄存器

只要新的指令从主机处理器发出，该类的主机接口寄存器将覆盖或更新。带有输入变元的所有指令和带有返回值所有相应的应答将被通过使用这些寄存器操纵。

地址0x000 ~ 0x0FC (64位寄存器地址空间) 是H/W寄存器，该寄存器有复位值和确定的功能（不能配置）。地址0x100~0x1FC是通用S/W寄存器。它们没有复位值，可通过BIT固件配置。它们主要用于主机和BIT处理器之间的接口。

高32位寄存器（地址是0x100~0x17C）用于静态参量。对于所有种类的运行指令（EQ\_INIT, SEQ\_END,

PICTURE\_RUN, ...), 它们功能并不改变, 并且应用到整个指令和处理过程。低32位寄存器(地址是0x180~0x1FC)用做暂时指令变元。对于每条指令, 这些寄存器的功能或意义可能发生改变。

例如, BitStreamCtrl寄存器用于整个处理过程。意思是通过BitStreamCtrl寄存器影响到所有的编码/解码操作。但是CMD\_DEC\_SEQ\_STRIDE寄存器只应用到执行DEC\_SEQ\_INIT指令的当前处理过程中。

[LineStride]的值只用于当前处理, 因此所有的处理将有不同的线跳跃偏移, 因为BIT处理器在执行DEC\_SEQ\_INIT 期间读CMD\_DEC\_SEQ\_STRIDE寄存器, 和存储[LineStride]值到内部存储器。

## 21.13.2. BIT 处理器代码下载

BIT 处理器有 6144 个字的内部代码存储器。内部编码存储器用做通过 BIT 固件指示缓存控制。整个代码镜像可能大于 6144 字, 并且必须属于 SDRAM。运行时, BIT 固件从 SDRAM 载入合适的代码镜像。主机必须通过设置[CodeBufAddr]寄存器, 告知 BIT 处理器代码镜像的起始 SDRAM 字节地址。

例如, 两个处理过程(MPEG4 解码器, H.264 编码器)同时运行。如果主机在 DEC\_PIC\_RUN 指令后执行 ENC\_PIC\_RUN 指令, BIT 处理器需要 H.264 编码代码镜像来执行 H.264 编码一个图片, 因此, 它自动从 SDRAM (内容交换) 载入 H.264 编码代码镜像。对于初始 BIT 处理器执行, 主机必须直接下载初始导入代码镜像。通过设置[CodeRun]寄存器执行 BIT 处理器之前, 通过设置[CodeDownload]寄存器, 主机必须直接下载导入代码镜像(一些达到代码镜像的最上部)到最低代码存储器(地址 0)。

BIT 固件的当前版本的编码镜像总的字节数是 80 KB(40 个字), 引导代码镜像是 1024 字节(512 字)。如图 21-42 所示, 显示位处理器的代码下载。

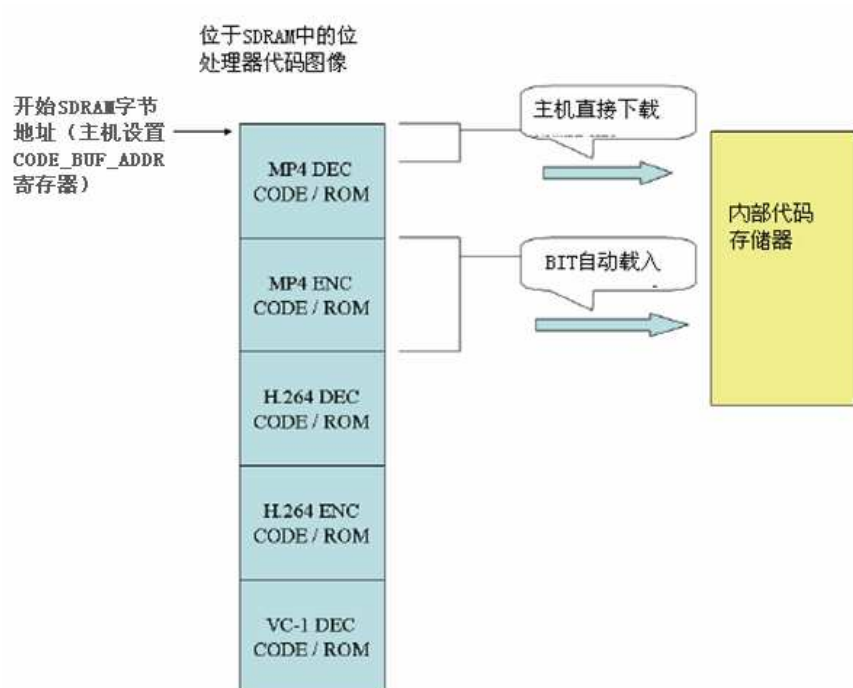


图 21-42 位处理器代码下载

### 21.13.3. 位流缓冲区管理

位于 SDRAM 的外部位流缓冲器由环形缓冲器组成。主机必须将环形缓冲器的起始地址和大小写入 BIT 处理器。

在解码的时候，主机写入要被解码的位流，接下来 BIT 处理器读读取该位流。在这种情况下，位流覆盖或者向下溢出可能发生，解码将失败。为了防止覆盖或者向下溢出，当前位流读/写指针必须在主机和 BIT 处理器之间执行。BIT 处理器写环形缓冲器的当前读指针到 BitStreamRdPtr 寄存器，并且主机必须写环形缓冲器的当前写指针到 BitStreamWrPtr 寄存器。BIT 处理器通过比较当前的 BitStreamRdPtr 和 BitStreamWrPtr 来检测位缓冲区空（向下溢出）状态。如果没有得到更多要被解码的位流数据（缓冲区空状态），BIT 处理器停止位流解码来防止误读该位流，并且一直等到主机写入更多的位流数据，更新 BitStreamWrPtr。在写更多位流数据到环形缓冲区前，主机必须核对当前 BitStreamRdPtr 和 BitStreamWrPtr 来防止覆盖位流数据。

在编码时，BIT 处理器和主机改变卷轴，例如，BIT 处理器将位流写入，主机读取该编码的位流。因此 BIT 处理器将环形缓冲器的当前写指针写入 BitStreamWrPtr 寄存器，主机必须将环形缓冲器的当前读指针写入 BitStreamRdPtr 寄存器。如果环形缓冲器为空，BIT 处理器将等待编码位流数据。因此下一次写入将覆盖预存储的数据，主机还没有读取这些数据。如果环形缓冲器为空，主机将一直等到读取被编码的位流数据，因此下一次读取信息的时候，又读取了一次预读数据。

如果 BitStreamCtrl 寄存器中的 [BufStsCheckDis] 标志是“1”，BIT 处理器禁止位流的向上溢出/向下溢出检测选项。因此编码/解码因位流缓冲区状态的原因，操作不会停止。在这种情况下，主机处理器不需要将当前读/写指针写入 BitStreamRdPtr/BitStreamWrPtr 寄存器里，但是主机保证了不发生位流向上溢出/向下溢出。

BitStreamCtrl 寄存器的 [BufPicReset] 标志是“1”，在编码条件下，外部位流缓冲器不会作为一个环形缓冲器。被编码的位流将一直写到外部位流缓冲器的起始位。因此，ENC\_PIC\_RUN 指令完成，主机必须读一张图片的整个编码的位流。

如果 BitStreamCtrl 寄存器的 [BufPicFlush] 标志是“1”，在每次对一张图片编码结束时，一张图片所有被编码的位流被写入到外部位缓冲器。精确的字节地址被写入到 BitStreamWrPtr 寄存器中。

BIT 处理器中也存在内部位流缓冲器，它是 BIT 处理器和 SDRAM 的外部位流缓冲器之间的高速缓冲存储器。在解码的情况下，BIT 处理器从 SDRAM 的外部位流缓冲器读取位流数据，并且存储到内部位流缓冲器。为了提高效率，真正的分解操作是在内部缓冲器中执行的。在编码的时候，被编码的位流数据首先存储在内部缓冲器，然后成块地写入外部缓冲器。内部位流缓冲器有 512 个字节。因此，位流数据以 512 个字节从外部缓冲器读取数据（在编码的情况下，写入数据）。BitStreamRdPtr 或 BitStreamWrPtr 寄存器以 512 个字节一组增加，由于内部位流数据被读/写只以 512 个这些剩余字节做为一块。因此，编码一张图片结束时，并不是所有被编码的位流写入到外部位缓冲器。该图片最后编码的位流数据（少于 512 字节），有一定数量存在于内部位缓冲器，并且当聚集的被编码的数据超过 512 个字节，这些存在于内部缓冲区的数据被冲掉。为了得到这些剩余的被编码的数据，主机执行 ENC\_SEQ\_END 指令，BIT 处理器将剩余的这些数据存储到外部位缓冲器。在 ENC\_SEQ\_END 指令之后，BitStreamWrPtr 寄存器增长到确定的字节地址，主机将计算被编码的位流的精确的字节大小。

## 21.13.4. 运行指令的描述

指令变元寄存器必须通过主机执行该指令来设置。指令完成后，主机必须设置 BusyFlag 寄存器为“0”。

BIT 处理器接受新的指令，BusyFlag 寄存器是“1”，当指令完成，BusyFlag 设置为“0”。主机通过轮询 BusyFlag 寄存器或者通过 BIT 处理器中断来确认指令完成。

### **21.13.5. DEC\_SEQ\_INIT**

该指令开始一个解码过程。在BIT处理器执行DEC\_SEQ\_INIT指令期间，所有的DEC\_SEQ\_INIT指令变元寄存器的值存储到内部存储器，并且应用到当前处理过程中。因此，DEC\_SEQ\_INIT指令寄存器的值将在后面有指令跟随期间（例如，DEC\_PIC\_RUN指令）不会改变。

在 DEC\_SEQ\_INIT指令中，BIT处理器将找到顺序标头，分解标头来获得位流信息（如图片大小）。接下来将这些信息报告给DEC\_SEQ\_INIT返回寄存器。

### **21.13.6. ENC\_SEQ\_INIT**

该指令开始一个编码过程。在 ENC\_SEQ\_INIT 指令中，BIT 处理器从指令变元寄存器读取编码参数和编码顺序标头。存在于 ENC\_SEQ\_INIT 指令变元寄存器的编码参数被应用到后面的 ENC\_PIC\_RUN 指令，并且不改变。例如，CMD\_ENC\_SLICE\_MODE 寄存器的[SliceMode]应用于所有图片，但是CMD\_ENC\_PIC\_QS 寄存器的[PictureQs]在每一条 ENC\_PIC\_RUN 指令该变。如果主机设置的编码参数无效，BIT 处理器将错误标志发送到 RET\_ENC\_SEQ\_SUCCESS 寄存器。

### **21.13.7. DEC\_SEQ\_END**

该指令结束解码过程。执行 DEC\_SEQ\_END 后，不再接收 DEC\_PIC\_RUN 指令。

### **21.13.8. ENC\_SEQ\_END**

该指令结束编码过程。如果[BufPicFlush]标志是“0”，并且[BufPicReset]标志是“0”，BIT 处理器将剩余的内部被编码的位流数据传输到外部位流数据缓冲器。再执行 ENC\_SEQ\_END 指令后，不再接收 ENC\_PIC\_RUN 指令。

### 21.13.9. DEC\_PIC\_RUN

该指令解码一张图片。解码图片后，BIT处理器向RET\_DEC\_PIC\_ERR\_MB\_NUM报告解码成功，并且在RET\_DEC\_PIC\_IDX寄存器显示帧缓冲器索引。

### 21.13.10. ENC\_PIC\_RUN

该指令编码一张图片。编码源帧缓冲器的SDRAM地址必须设置到指令变元寄存器。

### 21.13.11. SET\_FRAME\_BUF

该指令告知BIT处理器，帧缓冲器地址要被用来解码/重组图像。一共63个帧缓冲器可能要用到。通过RET\_DEC\_SEQ\_FRAME\_NEED r寄存器可获知用到的帧缓冲器的最小数量。在编码的情况下，两个帧缓冲器就够了。主机通过设置CMD\_SET\_FRAME\_BUF\_NUM寄存器来设置要用到的帧缓冲器的数量。

解码（解码条件下）/重组（编码条件下）图像应当保留以用作运动补偿参考，直到对于参考无用。因此该解码的/重组的帧缓冲器被再次用到。在编码/解码图片前，BIT处理器通过该指令保存整个帧缓冲器，然后为下一个存储区的编码/解码管理帧缓冲器的分配。

帧缓冲器地址存储到地址[ParaBufAddr]的SDRAM。必须被存储每一帧亮度和两个色度缓冲器的地址索引。地址格式介绍如图 21-43 所示。

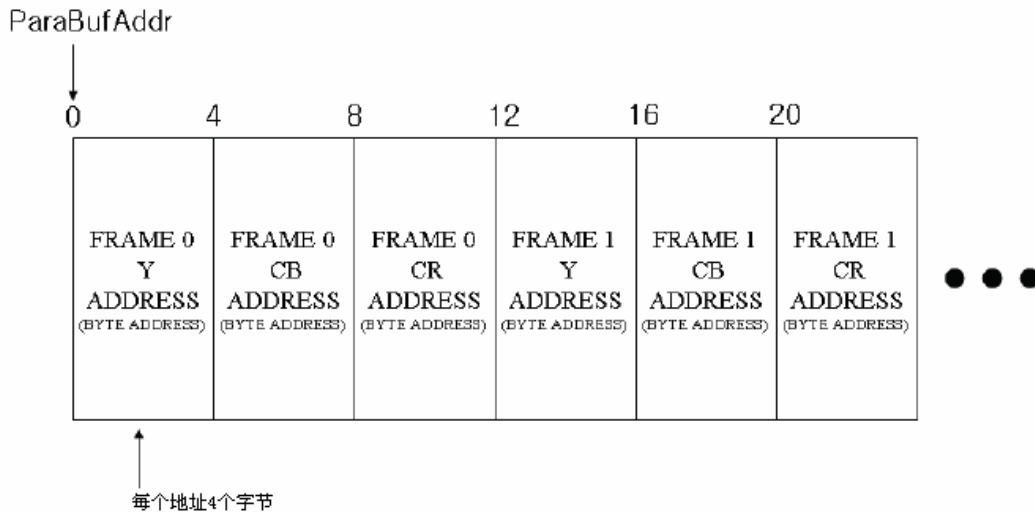


图 21-43 帧缓冲器地址格式

该指令编码特殊的表头。顺序标头必须在位流的起始位编码，在 ENC\_SEQ\_INIT 指令中，BIT 处理器编码顺序标头。在一些应用，如视频电话中，对于错误恢复或解码恢复，顺序标头必须传输。在该情况下，在 ENC\_PIC\_RUN 指令之间设置 ENC\_HEADER 指令来插入精确的标头。输入标头的标头编码必须设置到 CMD\_ENC\_HEADER\_CODE 寄存器中。

### 21.13.12. DEC\_PARA\_SET

在 H.264 情况下，该指令用于将顺序参数设置或者图片参数设置传送到解码器上。

顺序参数设置或者图片参数设置可能通过“out-of-band”传递。在该情况下，主机必须通过该指令传输顺序参数设置或者图片参数设置到解码器。顺序参数设置或图片参数设置必须写入 BIT 处理器的参数缓冲器。BIT 处理器解码传输顺序/图片参数和存储解码内容。解码的顺序/图片参数设置在带有匹配的顺序/图片参数设置 id 的解码部分标头是有效的。

多顺序/图片参数设置可能传递到解码器。它们通过不同的顺序/图片设置 id 来区分。在顺序设置 RBSP 中，顺序参数设置 id 被编码以 5 位（0~31）。图片参数设置 id 以 8 位（0~255）编码。BIT 处理器能处理 32 个顺序参数设置和 256 个图片参数设置。被传递的顺序/图片参数设置的类型（顺序或者图片）和大小（以字节计数）必须通过指令变元寄存器传递到 BIT 处理器。



### 21. 13. 13. ENC\_PARA\_SET

在 H. 264 情况下，该指令编码顺序参数设置或图片参数设置，并且通过参数缓冲器传递到主机。对于通过“out-of-band”传递的顺序/图片参数设置的应用，主机能通过该指令获得顺序/图片参数设置 RBSP。

BIT 处理器编码的顺序/图片参数设置和编码的 RBSP 流被存储到参数缓冲器，代替了正常的位流缓冲器。顺序/图片参数设置的编码的字节数量被指令返回寄存器返回。

#### (1) 参数缓冲器管理

参数缓冲器用于输入指令变元或者输出返回数据。参数缓冲器位于 SDRAM，开始字节地址必须设置到 ParaBufAddr 寄存器。

#### (2) 帧缓冲器地址

这个是输入指令变元，用于 SET\_FRAME\_BUF 指令。

#### (3) 编码的宏块位数

这是输出返回数据，用于 ENC\_PIC\_RUN 指令。如果 CMD\_ENC\_SEQ\_OPTION 寄存器的 [MbBitReport] 标志设置为“1”，ENC\_PIC\_RUN 指令完成后，BIT 处理器存储每一个宏块的起始位置到参数缓冲器。对于每一个宏块，开始位置以无符号的 16 位被存储。宏块的行幅度偏移是 128 个字节，因此最后  $128 - 720 / 16 \times 2 = 38$  字节不会用到。关于宏块位置的字节地址  $\langle MbX, MbY \rangle$  是  $\{ParaBufAddr + MbY \times 128 + MbX \times 2\}$ 。最大编码源图片高度是 576，所以宏块位数缓冲器的最大大小是  $128 \times 576 / 16 = 4608$  字节（4.5 KB）。详细格式如图 21-44 所示。

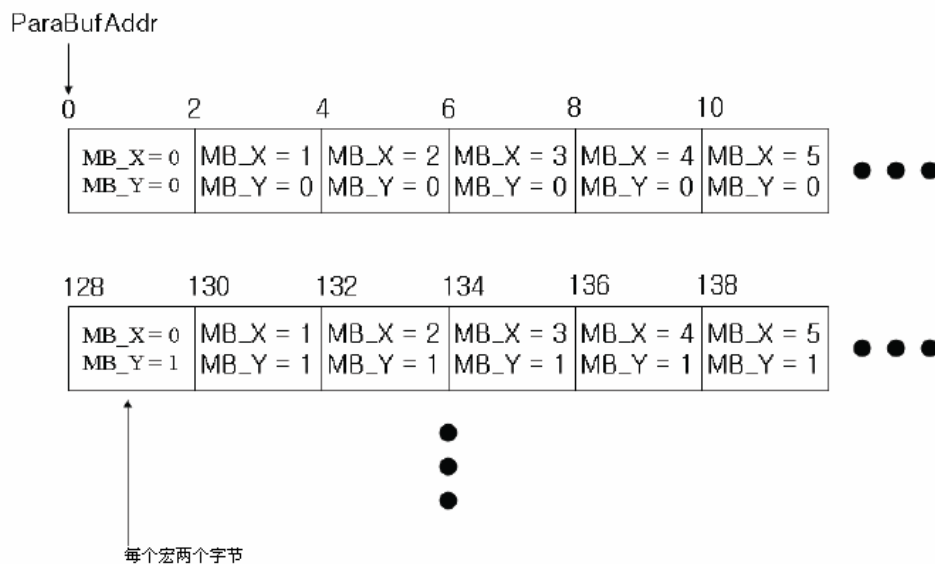


图 21-44 码的宏块位数

### 1. 编码的段的信息

这是输出返回数据，用于 ENC\_PIC\_RUN 指令。ENC\_PIC\_RUN 完成后，如果 CMD\_ENC\_SEQ\_OPTION 寄存器中的 [SliceInfoReport] 标志设置位“1”，BIT 处理器将每一个被编码的段的外部位流缓冲器的结束 SDRAM 地址存储到参数缓冲器。产生的段的总数存储到 RET\_ENC\_PIC\_SLICE\_NUM 寄存器，每一个段的结束位置的 SDRAM 地址存储到参数缓冲器。从每段的最后的位置，主机将计算每一段的精确的字节数。段位置缓冲器的起始地址是 {ParaBufAddr + 4608 (0x1200)}，仅在宏块位数缓冲器的下面。段的 SDRAM 地址结束位置是以无符号 32 位数表示。详细格式介绍如图 21-45 所示。