

CMD_ENC_SEQ_RC_BUF_SIZE (0x1B0)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|------------|-----|---|--------------|
| 31:0 | VbvBufSize | 读/写 | 参考解码器缓冲器大小，以位计 如果 RcEnable = 0 或 InitDelay = 0 该值被忽略。 允许的最大值是 0x7FFF_FFFF 0 :不检查参考解码器缓冲器大小限制 | ENC_SEQ_INIT |

CMD_ENC_SEQ_INTRA_MB (0x1B4)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-------------------|-----|---|--------------|
| 15:0 | IntraMbRefreshNum | 读/写 | 内部 MB 刷新量必须少于被编码的 (PictureHeight×PictureWidth) 0 - 不使用内部MB刷新 N - 每张图片至少 N 个 MBs 被编码做为内部模式 | ENC_SEQ_INIT |

CMD_ENC_SEQ_FMO (0x1B8)

| 位 | 名称 | 类型 | 功能 | 指令 |
|-----|------------|-----|-----------------------------|--------------|
| 0 | FmoEnable | 读/写 | 0: FMO 无效 1: FMO 有效 | ENC_SEQ_INIT |
| 4:1 | FmoSliceNr | 读/写 | 段组的数量 (必须是 2 到 8 之间的数) | |
| 5 | FmoType13 | 读/写 | 0: 类型0 (交叉) 1: 类型 1 (分散) | |

CMD_ENC_SEQ_TMP_BUF_1 (0x1D0)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-------------|-------|--|--------------|
| 31:0 | TempBufAddr | 读 / 写 | 临时缓冲器 SDRAM 字节地址 临时缓冲器必须是 256 字节对齐。 在执行 ENC_SEQ_INIT 指令之前，主机必须写入该寄存器。 临时缓冲器 1 被用做 ACDC 预测缓冲器 (对于 Mpeg4) 和内部预测 Y 缓冲器 (对于 AVC)。 | ENC_SEQ_INIT |

| | | | | |
|--|--|--|---|--|
| | | | 临时缓冲器 1 必须大于 $\text{picwidth} \times 8$ (对于 mpeg4) 和 $\text{stride} \times \text{picheight} / 16$ (对于 AVC)。 | |
|--|--|--|---|--|

CMD_ENC_SEQ_TMP_BUF_2 (0x1D4)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-------------|-----|--|--------------|
| 31:0 | TempBufAddr | 读/写 | <p>临时缓冲器 SDRAM 字节地址</p> <p>临时缓冲器必须是 256 字节对齐。</p> <p>执行 ENC_SEQ_INIT 指令前主机必须写入该寄存器。</p> <p>临时缓冲器 2 用作数据分区部分 2 保存缓冲器 (对于 Mpeg4) 和内部预测 Cb 缓冲器 (对于 AVC)。</p> <p>临时缓冲器 2 必须比 $(\text{stride}/2) \times (\text{picheight}/2)/8$ (对于 AVC) 大。</p> <p>对于 mpeg4 情况, 不能估计出临时缓冲器 2 大小。</p> | ENC_SEQ_INIT |

CMD_ENC_SEQ_TMP_BUF_3 (0x1D8)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-------------|-----|---|--------------|
| 31:0 | TempBufAddr | 读/写 | <p>临时缓冲器 SDRAM 字节地址</p> <p>临时缓冲器必须是 256 字节对齐。</p> <p>执行 ENC_SEQ_INIT 指令之前, 主机必须写入该寄存器。</p> <p>临时寄存器 3 用作数据分区部分 3 保存缓冲器 (对于 Mpeg4) 和内部预测 Cr 缓冲器 (对于 AVC)。</p> <p>临时缓冲器 3 必须比 $(\text{stride}/2) \times (\text{picheight}/2)/8$ (对于 AVC) 大。</p> <p>对于 mpeg4, 不能估计出临时缓冲器 3 的大小。</p> | ENC_SEQ_INIT |

CMD_ENC_SEQ_TMP_BUF_4 (0x1DC)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-------------|-------|--|--------------|
| 31:0 | TempBufAddr | 读 / 写 | <p>临时缓冲器 SDRAM 字节地址</p> <p>临时缓冲器必须是 4 字节对齐。</p> <p>执行 ENC_SEQ_INIT 指令前, 主机必须写入该寄存器。</p> <p>在 H.264 编码器中, 当 FMO 选项有效时, 临时缓冲器 4 被使用。该缓冲器</p> | ENC_SEQ_INIT |

| | | | | |
|--|--|--|---|--|
| | | | 的大小通过 $(32 \text{ KB} \times \text{FmoSliceNr})$ 得出。例如，如果段组数是 8，该缓冲器的大小应该是 $32 \times 8 \text{ KB}$ 。 | |
|--|--|--|---|--|

RET_ENC_SEQ_SUCCESS (0x1C0)

| | 名称 | 类型 | 功能 | 指令 |
|---|-----------|----|--|--------------|
| 0 | RetStatus | 读 | 0 - ENC_SEQ_INIT 指令执行错误 1 - ENC_SEQ_INIT 指令执行成功 | ENC_SEQ_INIT |

CMD_DEC_PIC_ROT_MODE (0x180)

| 位 | 名称 | 类型 | 功能 | 指令 |
|-----|-------------|-----|---|-------------|
| 3:0 | PostRotMode | 读/写 | 点旋转模式 PostRotMode[3:0] = {HorMir, VerMir, RotAng[1:0]} HorMir : 水平镜像 VerMir : 垂直镜像 RotAng[1:0] 0 : 逆时针旋转0度 1 : 逆时针旋转90度 2 : 逆时针旋转180度 3 : 逆时针旋转 270 度 | DEC_PIC_RUN |
| 4 | PostRotEn | 读/写 | 点旋转有效 如果该值是“1”，则旋转图像存储到DecPicRotAddrY, DecPicRotAddrCb, DecPicRotAddrCr 地址，用于将来参考。 如果该位是“0”，点旋转无效，并且 PostRotMode 被忽略。 | |

CMD_DEC_PIC_ROT_ADDR_Y (0x184)

| | 名称 | 类型 | 功能 | 指令 |
|------|-------------|-----|--|-------------|
| 31:0 | DecRotAddrY | 读/写 | 亮度旋转显示帧地址 如果 PostRotEn 域是“1”，旋转的图像存储到该地址。 在 VC—1 模式，不使用该地址。旋转输出将是 RET_DEC_SEQ_FRAME_NEED(0x1CC) 先前分派的一帧。 | DEC_PIC_RUN |

CMD_DEC_PIC_ROT_ADDR_CB (0x188)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|--------------|-------|--|-------------|
| 31:0 | DecRotAddrCb | 读 / 写 | Cb 的旋转显示帧地址 在 VC—1 模式，不使用该寄存器。旋转输出先前 RET_DEC_SEQ_FRAME_NEED(0x1CC) 分派的一帧。 | DEC_PIC_RUN |

CMD_DEC_PIC_ROT_ADDR_CR (0x18C)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|--------------|-------|--|-------------|
| 31:0 | DecRotAddrCr | 读 / 写 | Cr 的旋转显示帧地址 在 VC—1 模式，不使用该寄存器。旋转输出将是先前 RET_DEC_SEQ_FRAME_NEED(0x1CC) 分派的一帧。 | DEC_PIC_RUN |

CMD_DEC_PIC_DBK_ADDR_Y (0x190)

| | 名称 | 类型 | 功能 | 指令 |
|------|-------------|-----|--|-------------|
| 31:0 | DecDbkAddrY | 读/写 | 亮度解锁的显示帧地址 如果 Mp4DbkOn 域是“1”，解锁的图像存储在该地址。 | DEC_PIC_RUN |

CMD_DEC_PIC_DBK_ADDR_CB (0x194)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|--------------|-----|------------|-------------|
| 31:0 | DecDbkAddrCb | 读/写 | Cb 解锁显示帧地址 | DEC_PIC_RUN |

CMD_DEC_PIC_DBK_ADDR_CR (0x198)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|--------------|-----|------------|-------------|
| 31:0 | DecDbkAddrCr | 读/写 | Cr 解锁显示帧地址 | DEC_PIC_RUN |

CMD_DEC_PIC_ROT_STRIDE (0x19C)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|--------------|-----|---------|-------------|
| 10:0 | DecRotStride | 读/写 | 旋转显示帧幅度 | DEC_PIC_RUN |

CMD_DEC_PIC_CHUNK_SIZE (0x1A8)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-----------------|-----|--|-------------|
| 31:0 | DecPicChunkSize | 读/写 | 图片流数据字节大小 该值只在文件演示模式有效，并且当 BIT 处理器更新图片流缓冲器写指针时用作数据流的大小。 | DEC_PIC_RUN |

CMD_DEC_PIC_BB_START (0x1AC)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-------------------|-----|--|-------------|
| 31:0 | DecPicBitBufStart | 读/写 | 解码器输入如片流缓冲器的 4 字节对齐字节的地址 在文件演示模式下，如果解码器动态缓冲器分派选项有效，该值有效。在这种情况下，当 BIT 处理器更新图片流缓冲器的写指针，该地址将用作位流数据的其始地址。 | DEC_PIC_RUN |

CMD_DEC_PIC_START_BYTE (0x1B0)

| 位 | 名称 | 类型 | 功能 | 指令 |
|-----|----------------------|-----|--------------------|-------------|
| 1:0 | DecPicValidByteStart | 读/写 | 数据图片流缓冲器的有效位流的字节地址 | DEC_PIC_RUN |

RET_DEC_PIC_FRAME_NUM (0x1C0)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-------------|-----|---------------------------------------|-------------|
| 15:0 | DecFrameNum | 读/写 | 解码帧数 BIT 解码一帧，BIT 增加帧数和存储到该寄存器的帧数。 | DEC_PIC_RUN |

RET_DEC_PIC_IDX (0x1C4)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-----------|-------|---|-------------|
| 15:0 | DecPicIdx | 读 / 写 | 显示帧索引 BIT 解码一帧之后，BIT 返回显示帧索引到该寄存器。 该帧索引是帧缓冲器地址数组的索引，是主机通过 SET_FRAME_BUF 指令发出的。 如果 BIT 返回-1 (0xFFFF)，它表示所有图片已经都被解码，如果主机给的一个位流包含 20 张图片，主机将从第 21 个 PICTURE_RUN 得到-1 值。 在文件演示模式下，当 BIT 没有图片要显示时，将返回-3 (0xFFFD)。例如，当 BIT 解码 reorderEn 位有效的 H.264 流，在解码最多 16 张图片前，BIT 没有图片来显示。并且当 BIT 解码 VC-1 流，解码 2 张图片前，BIT 没有图片来显示。 | DEC_PIC_RUN |

RET_DEC_PIC_ERR_MB_NUM (0x1C8)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|----------|-----|---|-------------|
| 15:0 | ErrMbNum | 读/写 | 当前解码图片中错误的 MB 数 如果 BIT 识别出该流错误，BIT 在 MB 基础上执行错误隐蔽，在整个图片返回正确的 MB 数。 如果该值是“0”，解码的帧没有错误。 | DEC_PIC_RUN |

RET_DEC_PIC_TYPE (0x1CC)

| 位 | 名称 | 类型 | 功能 | 指令 |
|---|------------|-----|--|-------------|
| 0 | DecPicType | 读/写 | 当前解码的图片的图片类型 0 - I (内部) 图片 1 - P (内部) 图片 | DEC_PIC_RUN |

RET_DEC_PIC_SUCCESS (0x1D8)

| 位 | 名称 | 类型 | 功能 | 指令 |
|---|-------------|----|--|-------------|
| 0 | RetStatus | 读 | 0 - DEC_PIC_RUN 指令被执行，带有错误 1 - DEC_PIC_RUN 指令被成功执行 | DEC_PIC_RUN |
| 1 | Invalid PPS | 读 | 无效 PPS 当给出的流中没有有效的 PPS 该位被设置。 该位只在文件演示模式下被设置，并且当 RetStatus 位为 0 时有效。 | |

CMD_ENC_PIC_SRC_ADDR_Y (0x180)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|----------|-----|--|-------------|
| 31:0 | SrcAddrY | 读/写 | 亮度编码源帧地址 执行 ENC_PIC_RUN 指令前主机必须写入该寄存器。 在每一张编码图片前，主机必须设置 SDRAM 帧缓冲器起始地址到该寄存器。 | ENC_PIC_RUN |

CMD_ENC_PIC_SRC_ADDR_CB (0x184)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-----------|-----|------------|-------------|
| 31:0 | SrcAddrCb | 读/写 | Cb 的编码源帧地址 | ENC_PIC_RUN |

CMD_ENC_PIC_SRC_ADDR_CR (0x188)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-----------|-----|------------|-------------|
| 31:0 | SrcAddrCr | 读/写 | Cr 的编码源帧地址 | ENC_PIC_RUN |

CMD_ENC_PIC_QS (0x18C)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-----------|-----|--|-------------|
| 31:0 | PictureQs | 读/写 | 编码过程图片量化幅度参数 MPEG4/H. 263 模式，允许的范围是 1 到 31。 H. 264 模式，允许的范围是 0 到 51。 | ENC_PIC_RUN |

| | | | | |
|--|--|--|--|--|
| | | | 如果速率控制有效，该寄存器被忽略。如果速率控制无效，BIT 在该值范围内，对当前图片编码整个 MB。 | |
|--|--|--|--|--|

CMD_ENC_PIC_ROT_MODE (0x190)

| 位 | 名称 | 类型 | 功能 | 指令 |
|-----|------------|-----|---|-------------|
| 3:0 | PreRotMode | 读/写 | 预旋转模式 $PreRotMode[3:0] = \{HorMir, VerMir, RotAng[1:0]\}$ HorMir : 水平镜像 VerMir : 垂直镜像 RotAng[1:0] 0 : 逆时针旋转0度 1 : 逆时针旋转90度 2 : 逆时针旋转180度 3 : 逆时针旋转 270 度 如果该域是 4' b0000，预旋转无效。 | ENC_PIC_RUN |
| 4 | PreRotEn | 读/写 | 预旋转有效 如果该域是“1”，则源图像旋转优先于编码； 如果该域是“0”，则预旋转无效，并且 PreRotMode 被忽略。 | |

CMD_ENC_PIC_OPTION (0x194)

| 位 | 名称 | 类型 | 功能 | 指令 |
|---|-----------|-----|--|-------------|
| 0 | PicSkipEn | 读/写 | 图片跳跃标志 如果该域是“1”，EncSrcAddrY、EncSrcAddrCb、EncSrcAddrCr 被忽略，并且跳跃图片被编码。在这种情况下，重新组成的图片在解码中是先前图片的备份。不论[EncGopNum]是什么，该被跳过的图片被编码成 P 类型（内部）。 | ENC_PIC_RUN |

| | | | |
|---|--------|-----|--|
| | | | <p>当没有得到下一个要被编码的源帧时，主机可能设置该域为“1”。</p> <p>例如，在编码帧速为 5 Hz 的情况下，如果在第四张图片照相机输出没有获得，主机必须在 1 秒内设置 ENC_PIC_RUN 指令 5 次，并且设置 PicSkipEn 标志为“0、0、0、1、0”</p> |
| 1 | IdrPic | 读/写 | <p>如果该域是“1”，则不管 [EncGopNum] 的值，源图片被编码成 IDR IDR (Instantaneous Decoding Refresh) 图片 (H. 264) 或 I (内部) 图片 (MPEG-4/H. 263)。</p> <p>IDR 图是 0 frame_num 值的 I (内部) 图，所有的解码状态 (除了涉及到的图片列表) 被设置。</p> <p>编码 DIR 图片后，I 图周期计算被设置为初始状态。例如，如果主机设置 [IdrPic] 标志设置为第 18 帧，[EncGopNum] 是 15，编码图片类型是：</p> <p>第一帧：I (IDR - 自动)</p> <p>第二帧：P</p> <p>...</p> <p>第十四帧：P</p> <p>第十五帧：I</p> <p>第十六帧：P</p> <p>第十七帧：P</p> <p>第十八帧：I (IDR - 主机设置)</p> <p>第十九帧：P</p> <p>...</p> <p>第三十二帧：I</p> <p>第三十三帧：P</p> <p>...</p> <p>在 MPEG-4/H. 263 情况下，I (内部) 图片对于解码器恢复是足够的。对于插入编码器刷新点，主机必须周期性地设置该区域为“1”。</p> |

RET_ENC_PIC_FRAME_NUM (0x1C0)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-------------|-----|--|-------------|
| 15:0 | EncFrameNum | 读/写 | 有效帧数 BIT 编码一帧后, BIT 增加帧数, 并且将帧数存储到该寄存器。 | ENC_PIC_RUN |

RET_ENC_PIC_TYPE (0x1C4)

| 位 | 名称 | 类型 | 功能 | 指令 |
|---|------------|-----|--|-------------|
| 0 | EncPicType | 读/写 | 当前有效的图片的图片类型 0 - I (内部) 图片 1 - P (内部) 图片 | ENC_PIC_RUN |

RET_ENC_PIC_IDX (0x1C8)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-----------|-----|---------------------------------------|-------------|
| 15:0 | RecPicIdx | 读/写 | 重建的帧索引 BIT编码一帧后, BIT返回重建的帧索引到该寄存器。 | ENC_PIC_RUN |

RET_ENC_PIC_SLICE_NUM (0x1CC)

| 位 | 名称 | 类型 | 功能 | 指令 |
|------|-------------|-----|---|-------------|
| 14:0 | EncSliceNum | 读/写 | 如果 CMD_ENC_SEQ_OPTION 寄存器的[SliceInfoReport]标记, 则 BIT 返回编码段数到该寄存器。 每一段的编码段结束位置被存储到 SDRAM 中。 | ENC_PIC_RUN |

CMD_SET_FRAME_BUF_NUM (0x180)

| 位 | 名称 | 类型 | 功能 | 指令 |
|-----|-------------|-----|---|---------------|
| 4:0 | FrameBufNum | 读/写 | 用于参考或者数据再排序的帧数。 在解码条件下或者编码条件下是“2”, 该值必须大于或等于 RET_DEC_SEQ_FRAME_NEED 寄存器的[FrameBufNeed]。 主机必须设置相关的帧缓冲器 SDRAM 的地址 (Y, Cb, Cr) | SET_FRAME_BUF |