



图 22-8 运动 JPEG 解码的流程图

采取下列步骤，为运动的 JPEG 解码：

- (1) 在JPGMOD寄存器中，设置流程模式为解码流程。
- (2) 设置第一帧解码图像数据IMG_ADDR0和第二帧解码图像数据地址IMG_ADDR1的目的地址。
- (3) 设置第一帧JPEG文件HUFADDR0和下一JPEG文件地址HUFADDR1的源地址。
- (4) 在JPGCON寄存器中，设置运动JPEG解码模式。

MJ_ENC, HW_DEC, CLK_SEL, MJ_DEC

0	0	0	1	←基于HCLK的帧速率计数
0	0	1	1	←基于PCLK的帧速率计数

这里，PCLK是像素时钟，HCLK是系统时钟。

- (5) 在FRAM_INTV中，设置帧解码速率。该值必须大于一帧解码的时间。
- (6) 设置混合寄存器MISC（设置MODE_SEL 为0x1或 0x2，EMS位为0）。
- (7) 设置SW_JSTART为高电平。这是初始化开始命令。初始化开始命令以后，当FRAME_INTV寄存器的

显示的时间过去时，JPEG将产生下一个开始命令。

代码实现：

具体在 ARM11 中 Motion JPEG 的解码，代码编写如下：

```
void JPEG_StartDecodingMotion(u32 uJpgAddr, u32 uRawAddr, u32 uMjpegMaxSize)
{
    printf(" Dec: jpeg=%08x, raw=%08x\n", uJpgAddr, uRawAddr);
    JPEG_Reset();
    // Delay(1000);
    //=====//
    // For compling below func. is changed to InitRegsForDecoding(uJpgAddr, uRawAddr,
    HEADER_N_BODY)
    // so, below code shoule be modified when motion decoding func. is written
    #if 0
        InitRegsForDecoding(uJpgAddr, uRawAddr);
    #else
        JPEG_InitRegsForDecoding(uJpgAddr, uRawAddr, HEADER_N_BODY, false, false);
    #endif
    //=====//
    JPEG_SetNextFrameStartAddr(uJpgAddr + uMjpegMaxSize);
    Outp32(JPG_CON, (1<<0)); // Enable auto decoding
    // Outp32(JFRAME_INTV, 0x800000);
    Outp32(JFRAME_INTV, 0x300000);
    Outp32(JSTART, 1);
}
```

22.3 JPEG 编码寄存器

寄存器	偏移量	读/写	描述	复位值
JPGMOD	0x00	读/写	处理模式寄存器。	0x00000000
JPGSTS	0x04	读	操作状态寄存器。	0x00000000

JPGQHNO	0x08	读/写	量化表数目寄存器和哈夫曼表数目寄存器。	0x00000000
JPGDRI	0x0c	读/写	微控制器嵌入 RST 标记。	0x00000000
JPGY	0x10	读/写	垂直分辨率。	0x0000
JPGX	0x14	读/写	水平分辨率。	0x0000
JPGCNT	0x18	读	压缩数据的字节数目。	-
JPGIRQS	0x1c	读/写	中断设置寄存器。	0x0
JPGIRQ	0x20	读	中断状态寄存器。	-
QTBL0	0x400 0x404 0x4FC	写	8 位量化表数 0（在地址上具有 4 个远程交换的 64 数据）。	-
QTBL1	0x500 0x504 0x5FC	写	8 位量化表数 1（在地址上具有 4 个远程交换的 64 数据）。	-
QTBL2	0x600 0x604 0x6FC	写	8 位量化表数 2（在地址上具有 4 个远程交换的 64 数据）。	-
QTBL3	0x700 0x704 0x7FC	写	8 位量化表数 3（在地址上具有 4 个远程交换的 64 数据）。	-
HDCTBL0	0x800 0x804 0x83C	写	每个代码长度的数量（在地址上具有 4 个远程交换的 16 数据）。	-
HDCTBLG0	0x840	写	用于发生率的群阶数（在地址上具有 4 个	-

	0x844 0x86C		远程交换的 12 数据)。	
HACTBL0	0x880 0x884 0x8BC	写	每个代码长度的数量 (在地址上具有 4 个远程交换的 16 数据)。	-
HACTBLG0	0x8C0 0x8C4 0xB44	写	用于发生率/组数量的群阶数 (在地址上具有 4 个远程交换的 162 数据)。	-
HDCTBL1	0xC00 0xC04 0xC3C	写	每个代码长度的数量 (在地址上具有 4 个远程交换的 16 数据)。 8 位寄存器	-
HDCTBLG1	0xC40 0xC44 0xC6C	写	用于发生率的群阶数 (在地址上具有 4 个远程交换的 12 数据)。 8 位寄存器	-
HACTBL1	0xC80 0xC84 0xCBC	写	每个代码长度的数量 (在地址上具有 4 个远程交换的 16 数据)。 8 位寄存器	-
HACTBLG1	0xCC0 0xCC4 0xF44	写	用于发生率/组数量的群阶数 (在地址上具有 4 个远程交换的 162 数据)。 8 位寄存器	-
IMG_ADDR0	0x1000	读/写	源或目的图像地址 1。	0x00000000
IMG_ADDR1	0x1004	读/写	源或目的图像地址 2。	0x00000000

HUFADDR0	0x1008	读/写	源或目的 JPEG 文件地址 1。	0x00000000
HUFADDR1	0x100C	读/写	源或目的 JPEG 文件地址 2。	0x00000000
SW_JSTART	0x1010	读/写	开始 JPEG 过程： 0：没有开始过程 1：开始过程	0x00000000
SW_JRSTART	0x1014	读/写	重新开始 JPEG 过程： 0：没有重新开始过程 1：重新开始过程	0x00000000
S_RESET_CON	0x1018	读/写	0：软复位使能 1：软复位禁用	0x00000001
JPG_CON	0x101C	读/写	JPEG 控制寄存器。	0x00000000
COEF1	0x1020	读/写	系数值 RGB ↔ YcbC 的转换器。	0x00000000
COEF2	0x1024	读/写	系数值 RGB ↔ YcbC 的转换器。	0x00000000
COEF3	0x1028	读/写	系数值 RGB ↔ YcbC 的转换器。	0x00000000
MISC	0x102C	读/写	杂项。	0x00000000
FRAM_INTV	0x1030	读/写	帧间的间隔时间值。 (基础时钟周期)	0x00000000

22.3.1. JPEG 过程模式寄存器(JPGMOD)

Off=0x00, R/W, Reset=0x0000_0000

JPGMOD	位	描述	初始状态
Reserved	[31:4]	保留。	0x00000000
Process_Mode	[3]	过程模式： 0：编码过程。 1：解码过程。	
Sub_sampling_Mode	[2:0]	亚抽样模式： 0x0：色度4:4:4格式 0x1：色度4:2:2格式	

		0x2: 色度4:2:0格式 0x6: 色度4:1:1格式 0x3: 灰色格式(单一组成) 其他被禁止。在解码过程中, 它们是只读的。 在编码过程中, 只有 0x1 或 0x2 被允许。	
--	--	---	--

22.3.2. JPEG 状态寄存器 (JPGSTS)

Off=0x04, R, Reset=0x0000_0000

JPGSTS	位	描述	初始状态
Reserved	[31:1]	保留。	0x00000000
Status	[0]	过程模式: 1: 没有进行正常的操作 0: 进行正常的操作	

22.3.3. JPEG 量化表和哈夫曼表数目寄存器 (JPGQHNO)

Off=0x08, R/W, Reset=0x0000_0000

JPGQHNO	位	描述	初始状态
Reserved	[31:14]	保留。	0x00000000
Q_table_num3	[13:12]	量化表数目用于第三种色彩分量。	
Q_table_num2	[11:10]	量化表数目用于第二种色彩分量。	
Q_table_num1	[9:8]	量化表数目用于第一种色彩分量。	
Reserved	[7:6]	保留。	
H_table_num3_ac	[5]	哈夫曼表数目用于第三种交流色彩分量。	
H_table_num3_dc	[4]	哈夫曼表数目用于第三种直流色彩分量。	
H_table_num2_ac	[3]	哈夫曼表数目用于第二种交流色彩分量。	
H_table_num2_dc	[2]	哈夫曼表数目用于第二种直流色彩分量。	
H_table_num1_ac	[1]	哈夫曼表数目用于第一种交流色彩分量。	
H_table_num1_dc	[0]	哈夫曼表数目用于第一种直流色彩分量。	

注：灰色模式时，让第一和第三色彩分量有相同的值。（量化表数目和哈夫曼表数目）

22.3.4. JPEG 重置间隔寄存器(JPGDRI)

Off=0x0C, R/W, Reset=0x0000_0000

JPGDRI	位	描述	初始状态
Reserved	[31:16]	保留。	0x0000_0000
Reset_Interval	[15:0]	重置间隔，确定两个邻近的MCU期间的RST标记的距离。	

注：只有在压缩时才有效。当JPGDRI设置为零时，DRI和RST标记将不被声明。

22.3.5. JPEG 垂直大小寄存器(JPGY)

Off=0x10, R/W, Reset=0x0000_0000

JPGY	位	描述	初始状态
Reserved	[31:16]	保留。	0x0000_0000
VSIZE	[15:0]	在垂直方向上定义图像大小的值。	

22.3.6. JPEG 水平大小寄存器(JPGX)

Off=0x14, R/W, Reset=0x0000_0000

JPGX	位	描述	初始状态
Reserved	[31:16]	保留。	0x0000_0000
HSIZE	[15:0]	在水平方向上定义图像大小的值。	

22.3.7. JPEG 字节计数寄存器(JPGCNT)

Off=0x18, R, Reset=0x0000_0000

JPGCNT	位	描述	初始状态
Reserved	[31:24]	保留。	0x0000_0000

B_COUNT	[23:0]	在24位的宽度上，定义压缩数据计数的字节。	
---------	--------	-----------------------	--

22.3.8. JPEG; 中断设置寄存器(JPGIRQS)

Off=0x1C, R/W, Reset=0x0000_0000

JPGIRQS	位	描述	初始状态
Reserved	[31:4]	保留。	0x0000_0000
Intr_enb	[3]	中断使能控制。 0: 解压缩期间。在被压缩数据分析的结果中，禁用读图像的大小和取样因素的值。 1: 解压缩期间。在被压缩数据分析的结果中，使能读图像的大小和取样因素的值。	
Intr_ctrl	[2:0]	写0x0到设置中断。	

22.3.9. JPEG 中断暂存寄存器(JPGIRQ)

Off=0x20, R, Reset=0x0000_0000

JPGIRQ	位	描述	初始状态
Reserved	[31:7]	保留。	0x0000_0000
Result_status	[6]	结果状态： 0: 不正常的处理结束 1: 正常的处理完成	
Reserved	[5]	保留。	
Bitstre_err_status	[4]	位流的错误状态。只有在解压缩期间有效。 0: 在被压缩的文件上，没有语法错误。 1: 在被压缩的文件上，有语法错误。	
Header_status	[3]	标题状态。只有在解压缩期间有效。 0: 图像大小和取样因素值不可读。 1: 图像大小和取样因素值可读。	
Reserved	[2:0]	保留。	

注：该就寄存器的值在读出后将被重置。

22.3.10. JPEG 量化表 0 寄存器(QTBL0)

Off=0x400~0x4FC, R/W, Reset= -

QTBL0	位	描述	初始状态
Reserved	[31:8]	保留。	-
Q_val0	[7:0]	定义量化表0。在这里，用户必须写一些值。	

注：0x04作为地址偏移的增加（文字处理）。

22.3.11. JPEG 量化表 1 寄存器(QTBL1)

Off=0x500~0x5FC, R/W, Reset= -

QTBL1	位	描述	初始状态
Reserved	[31:8]	保留。	-
Q_val1	[7:0]	定义量化表1。在这里，用户必须写一些值。	

注：0x04作为地址偏移的增加（文字处理）。

22.3.12. JPEG 量化表 2 寄存器(QTBL2)

Off=0x600~0x6FC, R/W, Reset= -

QTBL2	位	描述	初始状态
Reserved	[31:8]	保留。	-
Q_val2	[7:0]	定义量化表2。在这里，用户必须写一些值。	

注：0x04作为地址偏移的增加（文字处理）。

22.3.13. JPEG 量化表 3 寄存器(QTBL3)

Off=0x700~0x7FC, R/W, Reset= -

QTBL3	位	描述	初始状态
Reserved	[31:8]	保留。	-
Q_val3	[7:0]	定义量化表3。在这里，用户必须写一些值。	

注：0x04作为地址偏移的增加（文字处理）。

22.3.14. JPEG 直流哈夫曼表 0 寄存器 (HDCTBL0)

Off=0x800~0x83C, W, Reset= -

HDCTBL0	位	描述	初始状态
Reserved	[31:8]	保留。	-
H_DC_val0	[7:0]	在直流哈夫曼表0中，定义每个编码长度的编码数目。 在这里，用户必须写一些值。	

注：0x04作为地址偏移的增加（文字处理）。

22.3.15. JPEG 组直流哈夫曼表 0 寄存器 (HDCTBLG0)

Off=0x840~0x86C, W, Reset= -

HDCTBLG0	位	描述	初始状态
Reserved	[31:8]	保留。	-
H_DC_G_val0	[7:0]	用于发生在直流哈夫曼表0中，定义顺序的组数目。在这里，用户必须写一些值。	

注：0x04作为地址偏移的增加（文字处理）。