ПЛАТФОРМА МУЛЬТИКОР

ПРИКЛАДНАЯ БИБЛИОТЕКА

БИБЛИОТЕКА ФУНКЦИЙ ПО ОБРАБОТКЕ ИЗОБРАЖЕНИЙ ДЛЯ МС-12

Руководство программиста

Листов 71

2005

Порядок использования настоящей документации

Настоящая документация охраняется действующим законодательством Российской Федерации об авторском праве и смежных правах, в частности, законом Российской Федерации «Об авторском праве и смежных правах». ГУП НПЦ «ЭЛВИС» является единственным правообладателем исключительных авторских прав на настоящую документацию.

Настоящую документацию без предварительного согласия ГУП НПЦ «ЭЛВИС» запрещается:

- воспроизводить, т.е. изготавливать один или более экземпляров настоящей документации, ее части, в любой форме, любым способом;

- сдавать в прокат;

- публично показывать, исполнять или сообщать для всеобщего сведения;

- переводить;

- переделывать или другим образом перерабатывать (дорабатывать).

ГУП НПЦ «ЭЛВИС» оставляет за собой право в любой момент вносить изменения (дополнения) в настоящую документацию без предварительного уведомления о таком изменении (дополнении).

ГУП НПЦ «ЭЛВИС» не несет ответственности за вред, причиненный при использовании настоящей документации.

Передача настоящей документации не означает передачи каких-либо авторских прав ГУП НПЦ «ЭЛВИС» на нее.

Возникновение каких-либо прав на материальный носитель, на котором передается настоящая документация, не влечет передачи каких-либо авторских прав на данную документацию.

Все указанные в настоящей документации товарные знаки принадлежат их владельцам.

АННОТАЦИЯ

В документе “Библиотека функций по обработке изображений” приведено описание прикладных функций по обработке изображений, выполняющих преобразование компонент изображения, прямое/обратное вейвлет-преобразование, квантование/декватнование, кодирование/декодирование. Прикладные функции по обработке изображений предназначены для обработки данных представленных как в формате с плавающей точкой - 24E8 (стандарт IEEE 754), так и в формате с фиксированной точкой - целые числа со знаком в дополнительном коде.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Необходимое программное обеспечение для функционирования библиотеки

Для функционирования библиотеки необходим программный симулятор платформы MultiCore, например система MultiCore Studio.

1.2. Языки программирования, используемые для написания библиотеки

Ассемблер для ядра ELcore платформы МУЛЬТИКОР.

2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

2.1. Назначение программы

Прикладная библиотека функций по обработке изображений предназначена для программной реализации различных алгоритмов сжатия изображения на основе вейвлет-преобразования для использования при обработке изображений ядром ELcore платформы МУЛЬТИКОР.

В состав библиотеки входят функции, необходимые для реализации алгоритмов сжатия изображения на основе вейвлет-преобразования в соответствии с фильтром Ле Галла 5/3: RGBtoYUV, YUVtoRGB, FDWTint_HL, IDWTint_HL, FDWT2int_decomp, IDWT2int_recomp, – перечисленные функции выполняют обработку данных, представленных в формате с фиксированной точкой - целые числа со знаком в дополнительном коде.

В состав библиотеки входят функции, необходимые для реализации алгоритмов сжатия изображения на основе вейвлет-преобразования в соответствии с фильтром Добеши 9/7: RGBtoYCrCb, YCrCbtoRGB, FDWTreal, IDWTreal, FDWT2real_decomp, IDWT2real_recomp, FWTreal_NL_decomp, IWTreal_NL_recomp, FQnt_real_int, DQnt_real_int, FQnt_real_NL_int, DQnt_real_NL_int, - перечисленные функции выполняют обработку данных, представленных в формате с плавающей точкой - 24E8 (стандарт IEEE 754).

2.2. Условия применения программы

Библиотека функций по обработке изображений представляет собой файл – libVIDEO.a, в котором находятся объектные файлы, содержащие готовый к компоновке исполняемый код.

Для подключения и использования библиотеки в проекте необходимо:

1) В диалоге *Tools* > *Settings* > для компоновщика DSP добавить директиву -L с указанием адреса библиотеки и её именем, заменяя обязательный префикс “lib” на префикс “l”: –L<путь к библиотеке> –lVIDEO
2) Откорректировать скрипт, используемый компоновщиком DSP – файл с расширением .xl. Предварительно файл необходимо переименовать, чтобы при

компиляции его не затерла MCStudio, и прописать новое имя файла в строке вызова компоновщика. Сам скрипт изменить следующим образом:

```
SECTIONS {
_dsp_LMA_ = . ;

.dsp_text 0x00000000 : AT(_dsp_LMA_)
{
 <ваш файл с программой DSP>.*(.text);
 <имя функции>.*(.text);
}

.dsp_data 0x00000000 : AT(_dsp_LMA_  + SIZEOF(.dsp_text))
{
 <ваш файл с программой DSP >.*(.data);
 <имя функции>.*(.data);
}

.dsp_bss :
{
 *(COMMON)
 <имя функции>.*(.bss);
}
}
```

3) Вызов функции в программе ядра ELcore осуществляется инструкцией:

```
BS <имя функции>
```

4) Передача входных параметров и возврат результатов выполнения функции осуществляется через регистры, указанные в описании конкретной функции.

3. СОСТАВ БИБЛИОТЕКИ

В настоящее время библиотека включает следующие функции:

1) Функция **RGBtoYUV** – прямое обратимое преобразование компонент (RCT) из цветового пространства RGB в YUV;
2) Функция **YUVtoRGB** – обратное преобразование компонент (RCT) из цветового пространства YUV в RGB;
3) Функция **RGBtoYC$_r$C$_b$** – прямое необратимое преобразование компонент (ICT) из цветового пространства RGB в YC_rC_b;
4) Функция **YC$_r$C$_b$toRGB** – обратное преобразование компонент (ICT) из цветового пространства YC_rC_b в RGB;
5) Функция **FDWTint_HL** – прямое дискретное вейвлет-преобразование одномерного массива в соответствии с фильтром Ле Галла 5/3;
6) Функция **IDWTint_HL** – обратное дискретное вейвлет-преобразование одномерного массива в соответствии с фильтром Ле Галла 5/3;
7) Функция **FDWTreal** – прямое дискретное вейвлет-преобразование одномерного массива в соответствии с фильтром Добеши 9/7;
8) Функция **IDWTreal** – обратное дискретное вейвлет-преобразование одномерного массива в соответствии с фильтром Добеши 9/7;
9) Функция **FDWT2int_decomp** – прямое дискретное вейвлет-преобразование двумерного массива в соответствии с фильтром Ле Галла 5/3 (декомпозиция исходной матрицы);
10) Функция **IDWT2int_recomp** – обратное дискретное вейвлет-преобразование двумерного массива в соответствии с фильтром Ле Галла 5/3 (рекомпозиция исходной матрицы);
11) Функция **FDWT2real_decomp** – прямое дискретное вейвлет-преобразование двумерного массива в соответствии с фильтром Добеши 9/7 (декомпозиция исходной матрицы);
12) Функция **IDWT2real_recomp** – обратное дискретное вейвлет-преобразование двумерного массива в соответствии с фильтром Добеши 9/7 (рекомпозиция исходной матрицы);

13) Функция **FWTreal_NL_decomp** – N_L-уровневая декомпозиция исходной матрицы - N_L-уровней прямого дискретного вейвлет-преобразования двумерного массива в соответствии с фильтром Добеши 9/7;
14) Функция **IWTreal_NL_recomp** – N_L-уровневая рекомпозиция исходной матрицы - N_L-уровней обратного дискретного вейвлет-преобразования двумерного массива в соответствии с фильтром Добеши 9/7;
15) Функция **FQnt_real_int** – квантование двумерного массива в соответствии с заданным шагом квантования;
16) Функция **DQnt_real_int** – деквантование двумерного массива в соответствии с заданным шагом квантования;
17) Функция **FQnt_real_NL_int** – равномерное скалярное квантование поддиапазонов коэффициентов вейвлет-преобразования, полученных после N_L-уровневой декомпозиции исходной матрицы;
18) Функция **DQnt_real_NL_int** – равномерное скалярное деквантование поддиапазонов квантованных коэффициентов вейвлет-преобразования, полученных после N_L-уровневой декомпозиции исходной матрицы;
19) Функция **HENC** – кодирование по Хаффману двумерного массива;
20) Функция **HDEC** – декодирование по Хаффману двумерного массива;

4. ХАРАКТЕРИСТИКА ПРОГРАММЫ, ОБРАЩЕНИЕ К ПРОГРАММЕ, ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

4.1. Функция RGBtoYUV

4.1.1. Описание функции

Функция выполняет:

- сдвиг каждой компоненты изображения (DC level shift) по яркости;
- преобразование компонент изображения из цветового пространства RGB в YUV в соответствии с прямым обратимым преобразованием компонент RCT.

4.1.2. Входные данные

Функция обрабатывает массив пикселей в формате RGB, представленных 32-разрядными словами PIX[31:0], где PIX[31:24] = 0,
PIX[23:16] = R-компонента, PIX[15:8] = B-компонента, PIX[7:0] = G-компонента;
каждая компонента изображения задается целым беззнаковым 8-разрядным значением.

Регистр A0: базовый адрес исходного массива пикселей в формате RGB.

Регистр R18 (short): N – число пикселей в RGB-массиве. Число пикселей должно быть четным и находится в диапазоне: 2...32764.

Регистр A1: адрес массива значений Y-компоненты.

Регистр A2: адрес массива значений U- компоненты.

Регистр A3: адрес массива значений V- компоненты.

4.1.3. Выходные данные

В результате обработки исходного RGB-массива функция формирует три массива: массив Y-компонент, массив U-компонент, массив V-компонент. При этом полученные целые знаковые 16-разрядные значения (short) компонент изображения упаковываются в 32-разрядные слова: $\{Y_{i+1},Y_i\}$, $\{U_{i+1},U_i\}$, $\{V_{i+1},V_i\}$, где i - номер элемента в массиве.

4.1.4. Алгоритм вычисления функции

Сдвиг каждой компоненты изображения (DC level shift) по яркости:
выполняется путем вычитания из целых беззнаковых 8-разрядных значений компоненты величины $2^{P-1} = 128$, где P=8 – точность представления компоненты изображения.
В результате сдвига каждая из компонент задается целыми знаковыми 8-разрядными значениями.
Прямое обратимое преобразование компонент изображения RCT из цветового пространства RGB в YUV:

$$\begin{cases} Y = [(R + 2G + B) / 4], \\ V = R - G, \\ U = B - G, \qquad (\text{где } [x] = \text{целая часть } x). \end{cases}$$

4.1.5. Затраты памяти

Функция использует регистры: R0…R6.
PRAM: 22 32-разрядных слов.
XRAM: (N×3)/2 32-разрядных слов.

4.1.6. Количество машинных тактов

5 тактов/пиксель.

4.1.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0     ; source RGB-array addr
        MOVE x,R18    ; source RGB array length
        MOVE x,A1     ; Y-component addr
        MOVE x,A2     ; U-component addr
        MOVE x,A3     ; V-component addr
; Вызов функции
        BS RGBtoYUV
        NOP
        ...
```

4.2. Функция YUVtoRGB

4.2.1. Описание функции

Функция выполняет:

- преобразование компонент изображения из цветового пространства YUV в RGB в соответствии с обратным преобразованием компонент RCT;
- обратный сдвиг каждой компоненты изображения (DC level shift) по яркости.

4.2.2. Входные данные

Функция обрабатывает три массива значений компонент: массив Y-компонент, массив U-компонент, массив V-компонент. При этом целые знаковые 16-разрядные значения (short) компонент упакованы в 32-разрядные слова:
$\{Y_{i+1},Y_i\}$, $\{U_{i+1},U_i\}$, $\{V_{i+1},V_i\}$, где i - номер элемента в массиве.

Регистр A1: адрес массива значений Y-компоненты.
Регистр A2: адрес массива значений U- компоненты.
Регистр A3: адрес массива значений V- компоненты.
Регистр R18 (short): N – число пикселей. Число пикселей должно быть четным и находится в диапазоне: 2...32764.
Регистр A0: адрес получаемого массива пикселей в формате RGB.

4.2.3. Выходные данные

В результате обработки массивов Y, U, V компонент функция формирует массив пикселей в формате RGB, представленных 32-разрядными словами PIX[31:0], где PIX[31:24] = 0, PIX[23:16] = R-компонента, PIX[15:8] = B-компонента, PIX[7:0] = G-компонента; каждая компонента изображения задается целым беззнаковым 8-разрядным значением.

4.2.4. Алгоритм вычисления функции

Обратное преобразование компонент изображения RCT из цветового пространства YUV в RGB:

$$\begin{cases} G = Y - [(U + V) / 4], \\ R = V + G, \\ B = U + G, \end{cases}$$ (где [x] = целая часть x).

Обратный сдвиг каждой компоненты изображения (DC level shift) по яркости: выполняется путем прибавления к целым знаковым 8-разрядным значениям компонент величины $2^{P-1} = 128$, где P=8 – точность представления компоненты изображения. В результате обратного сдвига каждая из компонент изображения представлятся беззнаковым целым 8-разрядным значением.

4.2.5. Затраты памяти

Функция использует регистры: R0…R8.

PRAM: 22 32-разрядных слов.

XRAM: N 32-разрядных слов.

4.2.6. Количество машинных тактов

5 тактов/пиксель.

4.2.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A1     ; Y-component addr
        MOVE x,A2     ; U-component addr
        MOVE x,A3     ; V-component addr
        MOVE x,R18    ; reconsructed RGB array length
        MOVE x,A0     ; reconsructed RGB-array addr
; Вызов функции
        BS YUVtoRGB
        NOP
        ...
```

4.3. Функция RGBtoYCrCb

4.3.1. Описание функции

Функция выполняет:

- сдвиг каждой компоненты изображения (DC level shift) по яркости;
- преобразование компонент изображения из цветового пространства RGB в YC_rC_b в соответствии с прямым необратимым преобразованием компонент ICT.

4.3.2. Входные данные

Функция обрабатывает массив пикселей в формате RGB, представленных 32-разрядными словами PIX[31:0], где PIX[31:24] = 0,
PIX[23:16] = R-компонента, PIX[15:8] = B-компонента, PIX[7:0] = G-компонента;
каждая компонента изображения задается целым беззнаковым 8-разрядным значением.

Регистр A0: базовый адрес исходного массива пикселей в формате RGB.

Регистр R18 (short): N – размер фрагмента изображения (N×N пикселей), представленного RGB-массивом. Значение N должно быть четным и находится в диапазоне: 2...16382.

Регистр A4: адрес буфера для размещения констант в YRAM (размер буфера – 9 ячеек памяти). Значение регистра A4 должно быть кратно 8, т.е. младшие 3 бита значения A4 должны быть = 0.

Регистр A1: адрес массива значений Y-компоненты.

Регистр A2: адрес массива значений U- компоненты.

Регистр A3: адрес массива значений V- компоненты.

4.3.3. Выходные данные

В результате обработки исходного RGB-массива функция формирует три массива: массив Y-компонент, массив U-компонент, массив V-компонент. Элементами массивов являются 32-разрядные значения компонент, представленные числами в формате с плавающей точкой (float).

4.3.4. Алгоритм вычисления функции

Сдвиг каждой компоненты изображения (DC level shift) по яркости:
выполняется путем вычитания из целых беззнаковых 8-разрядных значений компоненты величины $2^{P-1} = 128$, где P=8 – точность представления компоненты изображения.
В результате сдвига каждая из компонент задается целыми знаковыми 8-разрядными значениями.
Прямое необратимое преобразование компонент изображения ICT из цветового пространства RGB в YC_rC_b:

$$\begin{cases} Y = 0{,}299 \times R + 0{,}587 \times G + 0{,}114 \times B; \\ C_r = -0{,}16875 \times R - 0{,}33126 \times G + 0{,}5 \times B; \\ C_b = 0{,}5 \times R - 0{,}41869 \times G - 0{,}08131 \times B; \end{cases}$$

4.3.5. Затраты памяти

Функция использует регистры: R0 ... R13, AT, M4.
Перед запуском функции значение регистра MT должно быть: MT=FFFF.
PRAM: 62 32-разрядных слов.
XRAM: N×3 32-разрядных слов.
YRAM: 9 32-разрядных слов, начиная с адреса A4.

4.3.6. Количество машинных тактов

14 тактов/пиксель.

4.3.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0     ; source RGB-array addr
        MOVE x,R18    ; source RGB-array size
        MOVE x,A4     ; constants buffer addr
        MOVE x,A1     ; Y-component addr
        MOVE x,A2     ; U-component addr
        MOVE x,A3     ; V-component addr
; Вызов функции
        BS RGBtoYCrCb
        NOP
        ...
```

4.4. Функция YCrCbtoRGB

4.4.1. Описание функции

Функция выполняет:

- преобразование компонент изображения из цветового пространства YC_rC_b в RGB в соответствии с обратным преобразованием компонент ICT;
- обратный сдвиг каждой компоненты изображения (DC level shift) по яркости.

4.4.2. Входные данные

Функция обрабатывает три массива значений компонент: массив Y-компонент, массив U-компонент, массив V-компонент. Элементами массивов являются 32-разрядные значения компонент, представленные числами в формате с плавающей точкой (float).

Регистр A1: адрес массива значений Y-компоненты.

Регистр A2: адрес массива значений U- компоненты.

Регистр A3: адрес массива значений V- компоненты.

Регистр A4: адрес буфера для размещения констант в XRAM, либо YRAM (размер буфера – 4 ячейки памяти). Значение регистра A4 должно быть кратно 4, т.е. младшие 2 бита значения A4 должны быть = 0.

Регистр R18 (short): N – размер фрагмента изображения (N×N пикселей),
Значение N должно быть четным и находится в диапазоне: 2...16382.

Регистр A0: адрес получаемого массива пикселей в формате RGB.

4.4.3. Выходные данные

В результате обработки массивов Y, U, V компонент функция формирует массив пикселей в формате RGB, представленных 32-разрядными словами PIX[31:0], где PIX[31:24] = 0, PIX[23:16] = R-компонента, PIX[15:8] = B-компонента, PIX[7:0] = G-компонента;
каждая компонента изображения задается целым беззнаковым 8-разрядным значением.

4.4.4. Алгоритм вычисления функции

Обратное преобразование компонент изображения RCT из цветового пространства YC_rC_b в RGB:

$$\begin{cases} R = Y + 1{,}402 \times C_b; \\ G = Y - 0{,}34413 \times C_r - 0{,}71414 \times C_b; \\ B = Y + 1{,}772 \times C_r; \end{cases}$$

Обратный сдвиг каждой компоненты изображения (DC level shift) по яркости: выполняется путем прибавления к целым знаковым 8-разрядным значениям компонент величины $2^{P-1} = 128$, где P=8 – точность представления компоненты изображения. В результате обратного сдвига каждая из компонент изображения представлятся беззнаковым целым 8-разрядным значением.

4.4.5. Затраты памяти

Функция использует регистры: R0…R11, M4.

PRAM: 56 32-разрядных слов.

XRAM: N 32-разрядных слов.

XRAM /YRAM: 4 32-разрядных слов.

4.4.6. Количество машинных тактов

21 тактов/пиксель.

4.4.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A1     ; Y-component addr
        MOVE x,A2     ; U-component addr
        MOVE x,A3     ; V-component addr
        MOVE x,A4     ; constants buffer addr
        MOVE x,R18    ; reconsructed RGB-array size
        MOVE x,A0     ; reconsructed RGB-array addr
; Вызов функции
        BS YCrCbtoRGB
        NOP
        ...
```

4.5. Функция FDWTint_HL

4.5.1. Описание функции

Функция выполняет прямое дискретное обратимое вейвлет-преобразование (FDWT) исходного одномерного массива в соответствии с фильтром Ле Галла 5/3.

4.5.2. Входные данные

Функция обрабатывает одномерный массив целых знаковых 16-разрядных чисел (short), являющийся столбцом исходного двумерного массива. При этом 16-разрядные значения элементов одномерного массива упакованы в 32-разрядные слова: {Ai, Xi}, либо {Xi, Ai}, где Xi - i-ый элемент исходного массива, а Ai - 16-разрядное значение, не являющееся элементом исходного массива.

Регистр A0: базовый адрес исходного одномерного массива.

Регистр R18 (short): N – число элементов в массиве - должно быть четным и находится в диапазоне: 8...32764.

Регистр R20 (short): длина строки исходного двумерного массива.

Регистр R21 (short): если R21=0, то 16-разрядные значения элементов одномерного массива упакованы в младшей части 32-разрядные слова;
если R21=1, то 16-разрядные значения элементов одномерного массива упакованы в старшей части 32-разрядные слова.

Регистр A1: адрес массива коэффициентов вейвлет-преобразования.

4.5.3. Выходные данные

Функция формирует одномерный массив коэффициентов вейвлет-преобразования длиной N. При этом целые знаковые 16-разрядные значения (short) коэффициентов упакованы в 32-разрядные слова: {H,L}.

4.5.4. Алгоритм вычисления функции

Исходный одномерный массив X(i), $0 \le i \le n$, тогда расширенный массив:

$$X_{ext}(i),\ -2 \le i \le n+1.$$

Результатом вейвлет-преобразования расширенного массива является массив коэффициентов вейвлет-преобразования – Y(k), $0 \le k \le n$.

Нечетные коэффициенты (Highpass - высокочастотные):

$$Y(2m+1) = X_{ext}(2m+1) - \left\lfloor \frac{X_{ext}(2m) + X_{ext}(2m+2)}{2} \right\rfloor, \quad -1 \le 2m+1 \le n \quad ;$$

Четные коэффициенты (Lowpass - низкочастотные):

$$Y(2m) = X_{ext}(2m) + \left\lfloor \frac{Y(2m-1) + Y(2m+1) + 2}{4} \right\rfloor, \quad 0 \le 2m \le n-1 \quad ;$$

4.5.5. Затраты памяти

Функция использует регистры: R0…R9.

PRAM: 32 32-разрядных слов.

XRAM: N/2 32-разрядных слов.

4.5.6. Количество машинных тактов

(6×N) тактов.

4.5.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0     ; array addr
        MOVE x,R18    ; array length
        MOVE x,R20    ; string length
        MOVE x,R21    ; select high or low
        MOVE x,A1     ; WT-coefficients array addr
; Вызов функции
        BS FDWTint_HL
        NOP
        ...
```

4.6. Функция IDWTint_HL

4.6.1. Описание функции

Функция выполняет обратное дискретное вейвлет-преобразование (IDWT) одномерного массива в соответствии с фильтром Ле Галла 5/3.

4.6.2. Входные данные

Функция обрабатывает два одномерных массива коэффициентов вейвлет-преобразования. При этом целые знаковые 16-разрядные значения (short) коэффициентов упакованы в 32-разрядные слова: {Ai, WCi}, либо {WCi, Ai}, где WCi - i-ый элемент массива коэффициентов, а Ai - 16-разрядное значение, не являющееся элементом массива коэффициентов.

Регистр A0: адрес массива низкочастотных (lowpass) коэффициентов вейвлет-преобразования.

Регистр A1: адрес массива высокочастотных (highpass) коэффициентов вейвлет-преобразования.

Регистр R18 (short): N – число элементов в восстанавливаемом одномерном массиве, должно быть четным и находится в диапазоне: 8...32764.

Регистр R20 (short): длина строки исходного двумерного массива.

Регистр R21 (short): если R21=0, то 16-разрядные значения коэффициентов упакованы в младшей части 32-разрядные слова; если R21=1, то 16-разрядные значения коэффициентов упакованы в старшей части 32-разрядные слова.

Регистр A2: адрес восстанавливаемого одномерного массива.

4.6.3. Выходные данные

Функция восстанавливает одномерный массив, при этом целые знаковые 16-разрядные значения (short) элементов восстановленного массива упакованы в 32-разрядные слова: $\{X_{i+1}, X_i\}$.

4.6.4. Алгоритм вычисления функции

Исходный одномерный массив коэффициентов Y(k), $0 \le k \le n$, тогда расширенный массив коэффициентов: Yext(k), $-1 \le k \le n+2$.

Результатом обратного вейвлет-преобразования расширенного массива коэффициентов является восстановленный массив – X(i), $0 \le i \le n$.

Четные элементы массива:

$$X(2m) = Y_{ext}(2m) - \left\lfloor \frac{Y_{ext}(2m-1) + Y_{ext}(2m+1) + 2}{4} \right\rfloor, \quad 0 \le 2m \le n+1 \quad ;$$

Нечетные элементы массива:

$$X(2m+1) = Y_{ext}(2m+1) + \left\lfloor \frac{X(2m) + X(2m+2)}{2} \right\rfloor, \quad 1 \le 2m+1 \le n \quad ;$$

4.6.5. Затраты памяти

Функция использует регистры: R0…R9.

PRAM: 30 32-разрядных слов.

XRAM: N 32-разрядных слов.

4.6.6. Количество машинных тактов

(6×N) тактов.

4.6.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0     ; lowpass coefficients array addr
        MOVE x,A1     ; highpass coefficients array addr
        MOVE x,R18    ; array length
        MOVE x,R20    ; string length
        MOVE x,R21    ; select high or low
        MOVE x,A2     ; reconstructed array addr
; Вызов функции
        BS IDWTint_HL
        NOP
        ...
```

4.7. Функция FDWTreal

4.7.1. Описание функции

Функция выполняет прямое дискретное необратимое вейвлет-преобразование (FDWT) исходного одномерного массива в соответствии с фильтром Добеши 9/7.

4.7.2. Входные данные

Функция обрабатывает одномерный массив, 32-разрядные значения элементов которого представлены числами в формате с плавающей точкой (float).

Регистр A0: базовый адрес исходного одномерного массива.

Регистр R18 (short): N – число элементов в массиве - должно быть четным и находится в диапазоне: 8...32764.

Регистр R20 (short): смещение адреса между соседними элементами массивов.

Регистр A1: адрес массива низкочастотных (lowpass) коэффициентов вейвлет-преобразования.

Регистр A2: адрес массива высокочастотных (highpass) коэффициентов вейвлет-преобразования.

4.7.3. Выходные данные

Функция формирует два одномерных массива коэффициентов вейвлет-преобразования длиной N/2 каждый. При этом 32-разрядные значения коэффициентов представлены числами в формате с плавающей точкой (float).

4.7.4. Алгоритм вычисления функции

Исходный одномерный массив X(i), $0 \le i \le n$, тогда расширенный массив:

$X_{ext}(i),\ -4 \le i \le n+3$.

Результатом вейвлет-преобразования расширенного массива является массив коэффициентов вейвлет-преобразования – Y(k), $0 \le k \le n$.

$$
\begin{cases}
Y_0(2m+1) = X_{ext}(2m+1) + \alpha \times \left(X_{ext}(2m) + X_{ext}(2m+2)\right), & -3 \le 2m+1 \le n+2 \quad ; \\
Y_0(2m) = X_{ext}(2m) + \beta \times \left(Y_0(2m-1) + Y_0(2m+1)\right), & -2 \le 2m \le n+1 \quad ; \\
Y_1(2m+1) = Y_0(2m+1) + \gamma \times \left(Y_0(2m) + Y_0(2m+2)\right), & -1 \le 2m+1 \le n \quad ; \\
Y_1(2m) = Y_0(2m) + \delta \times \left(Y_1(2m-1) + Y_1(2m+1)\right), & 0 \le 2m \le n-1 \quad ;
\end{cases}
$$

Нечетные коэффициенты (Highpass - высокочастотные):

$$Y(2m+1) = -K \times Y_1(2m+1), \qquad 1 \le 2m+1 \le n \quad ;$$

Четные коэффициенты (Lowpass - низкочастотные):

$$Y(2m) = \frac{1}{K} \times Y_1(2m), \qquad 0 \le 2m \le n-1 \quad ;$$

В приведенных выше формулах коэффициенты $\alpha, \beta, \gamma, \delta$ имеют следующие значения:

$$
\begin{cases}
\alpha = -1{,}586134342; & \beta = -0{,}052980118; \\
\gamma = 0{,}882911075; & \delta = 0{,}443506852.
\end{cases}
$$

4.7.5. Затраты памяти

Функция использует регистры: R0…R9.

PRAM: 84 32-разрядных слов.

XRAM: N 32-разрядных слов.

4.7.6. Количество машинных тактов

(8×N) тактов.

4.7.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0     ; array addr
        MOVE x,R18    ; array length
        MOVE x,R20    ; addr inc step
        MOVE x,A1     ; lowpass coefficients array addr
        MOVE x,A2     ; highpass coefficients array addr
; Вызов функции
        BS FDWTreal
        NOP
        ...
```

4.8. Функция IDWTreal

4.8.1. Описание функции

Функция выполняет обратное дискретное вейвлет-преобразование (IDWT) одномерного массива в соответствии с фильтром Добеши 9/7.

4.8.2. Входные данные

Функция обрабатывает два одномерных массива коэффициентов вейвлет-преобразования. При этом 32-разрядные значения коэффициентов представлены числами в формате с плавающей точкой (float).

Регистр A0:	адрес массива низкочастотных (lowpass) коэффициентов вейвлет-преобразования.
Регистр A1:	адрес массива высокочастотных (highpass) коэффициентов вейвлет-преобразования.
Регистр R18 (short):	N – число элементов в восстанавливаемом одномерном массиве, должно быть четным и находится в диапазоне: 8...32764.
Регистр R20 (short):	смещение адреса между соседними элементами массивов.
Регистр A2:	адрес восстанавливаемого одномерного массива.

4.8.3. Выходные данные

Функция восстанавливает одномерный массив. При этом 32-разрядные значения элементов восстановленного массива представлены числами в формате с плавающей точкой (float).

4.8.4. Алгоритм вычисления функции

Исходный одномерный массив коэффициентов Y(k), $0 \le k \le n$, тогда расширенный массив коэффициентов: Yext(k), $-3 \le k \le n+4$.

Результатом обратного вейвлет-преобразования расширенного массива коэффициентов является восстановленный массив – X(i), $0 \le i \le n$.

$$
\begin{cases}
X_0(2m) = K \times Y_{ext}(2m), & -2 \le 2m \le n+3 \quad ; \\
X_0(2m+1) = -\dfrac{1}{K} \times Y_{ext}(2m+1), & -3 \le 2m+1 \le n+4 \quad ; \\
X_1(2m) = X_0(2m) - \delta \times \left(X_0(2m-1) + X_0(2m+1)\right), & -2 \le 2m \le n+3 \quad ; \\
X_1(2m+1) = X_0(2m+1) - \gamma \times \left(X_1(2m) + X_1(2m+2)\right), & -3 \le 2m+1 \le n+4 \quad ; \\
\text{Четные элементы массива:} \\
X(2m) = X_1(2m) - \beta \times \left(X_1(2m-1) + X_1(2m+1)\right), & 0 \le 2m \le n+1 \quad ; \\
\text{Нечетные элементы массива:} \\
X(2m+1) = X_1(2m+1) - \alpha \times \left(X_1(2m) + X_1(2m+2)\right), & 1 \le 2m+1 \le n \quad ;
\end{cases}
$$

В приведенных выше формулах коэффициенты $\alpha, \beta, \gamma, \delta$ имеют следующие значения:

$$
\begin{cases}
\alpha = -1{,}586134342; & \beta = -0{,}052980118; \\
\gamma = 0{,}882911075; & \delta = 0{,}443506852.
\end{cases}
$$

4.8.5. Затраты памяти

Функция использует регистры: R0…R9.

PRAM: 84 32-разрядных слов.

XRAM: N 32-разрядных слов.

4.8.6. Количество машинных тактов

(8×N) тактов.

4.8.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0     ; lowpass coefficients array addr
        MOVE x,A1     ; highpass coefficients array addr
        MOVE x,R18    ; array length
        MOVE x,R20    ; addr inc step
        MOVE x,A2     ; reconstructed array addr
; Вызов функции
        BS IDWTreal
        NOP
        ...
```

4.9. Функция FDWT2int_decomp

4.9.1. Описание функции

Функция выполняет прямое дискретное обратимое вейвлет-преобразование (FDWT) исходного двумерного массива в соответствии с фильтром Ле Галла 5/3 (декомпозиция исходной матрицы).

4.9.2. Входные данные

Функция обрабатывает двумерный массив, элементами которого являются знаковые целые 16-разрядных числа (short). При этом 16-разрядные значения элементов массива упакованы в 32-разрядные слова: {Xi(j+1) , Xij} , то есть соседние элементы j-ый и (j+1)-ый в строке i упакованы в одно 32-разрядное слово.

Регистр R14 (short): N – размер исходного двумерного массива (N×N элементов), значение N должно быть кратно 4 и находится в диапазоне: 8...32764.
Регистр R15 (short): базовый адрес исходного двумерного массива.
Регистр R19 (short): адрес буфера данных в XRAM. Объем буфера равен объему исходного двумерного массива.
Регистр R16 (short): адрес массива HH, HL коэффициентов вейвлет-преобразования.
Регистр R17 (short): адрес массива LH, LL коэффициентов вейвлет-преобразования.

4.9.3. Выходные данные

В результате декомпозиции исходного двумерного массива функция формирует два двумерных массива коэффициентов вейвлет-преобразования. При этом целые знаковые 16-разрядные значения (short) коэффициентов упакованы в 32-разрядные слова:

{HH, HL} – в первом массиве;

{LH, LL} – во втором массиве.

4.9.4. Алгоритм вычисления функции

Прямое вейвлет-преобразование двумерного массива заключается в применении прямого вейвлет-преобразования одномерного массива в соответствии с фильтром Ле Галла 5/3 сначала ко всем столбцам исходного двумерного массива, то есть фильтрация по вертикали, а затем ко всем строкам полученного двумерного массива, то есть фильтрация по горизонтали. Таким образом, прямое вейвлет-преобразование двумерного массива базируется на использовании функции прямого вейвлет-преобразования одномерного массива в соответствии с фильтром Ле Галла 5/3 – *FDWTint_HL*.

4.9.5. Затраты памяти

Функция использует регистры: R10, R11.

PRAM: 22 32-разрядных слов.

XRAM: (N×N) 32-разрядных слов;

4.9.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,R14    ; matrix size
        MOVE x,R15    ; source matrix addr
        MOVE x,R19    ; buffer addr
        MOVE x,R16    ; HH, HL coefficients array addr
        MOVE x,R17    ; LH, LL coefficients array addr
; Вызов функции
        BS FDWT2int_decomp
        NOP
        ...
```

4.10. Функция IDWT2int_recomp

4.10.1. Описание функции

Функция выполняет обратное дискретное вейвлет-преобразование (IDWT) двумерного массива в соответствии с фильтром Ле Галла 5/3 (рекомпозиция исходной матрицы).

4.10.2. Входные данные

Функция обрабатывает четыре двумерных массива коэффициентов вейвлет-преобразования. При этом целые знаковые 16-разрядные значения (short) коэффициентов упакованы в 32-разрядные слова:

$\{LL_{i+1}, LL_i\}$ – в первом массиве;

$\{LH_{i+1}, LH_i\}$ – во второом массиве;

$\{HL_{i+1}, HL_i\}$ – в третьем массиве;

$\{HH_{i+1}, HH_i\}$ – в четвертом массиве;

Регистр R14 (short): N – размер восстанавливаемого двумерного массива (N×N элементов), значение N должно быть кратно 4 и находится в диапазоне: 8...32764.
Регистр R15 (short): адрес массива LL-коэффициентов вейвлет-преобразования.
Регистр R16 (short): адрес массива LH-коэффициентов вейвлет-преобразования.
Регистр R17 (short): адрес массива HL-коэффициентов вейвлет-преобразования.
Регистр R19 (short): адрес массива HH-коэффициентов вейвлет-преобразования.
Регистр A2: адрес буфера данных в XRAM. Объем буфера равен объему восстанавливаемого двумерного массива.
Регистр A3: адрес восстанавливаемого двумерного массива.

4.10.3. Выходные данные

Функция восстанавливает двумерный массив размером N×N в транспонированном виде, при этом целые знаковые 16-разрядные значения (short) элементов восстановленного массива упакованы в 32-разрядные слова: {Xi(j+1) , Xij} , то есть соседние элементы j-ый и (j+1)-ый в строке i упакованы в одно 32-разрядное слово.

4.10.4. Алгоритм вычисления функции

При выполнении обратного вейвлет-преобразования двумерного массива четыре исходных массива коэффициентов вейвлет-преобразования объединяются в один двумерный массив, к полученному массиву применяется обратное вейвлет-преобразование одномерного массива в соответствии с фильтром Ле Галла 5/3 сначала ко всем строкам массива, то есть фильтрация по горизонтали, а затем ко всем столбцам полученного двумерного массива, то есть фильтрация по вертикали. Таким образом, обратное вейвлет-преобразование двумерного массива базируется на использовании функции обратного вейвлет-преобразования одномерного массива в соответствии с фильтром Ле Галла 5/3 – *IDWTint_HL*.

4.10.5. Затраты памяти

Функция использует регистры: R10…R12, R15…R17, R19.

PRAM: 32 32-разрядных слов.

XRAM: (N×N) 32-разрядных слов.

4.10.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,R14    ; reconstructed matrix size
        MOVE x,R15    ; LL coefficients array addr
        MOVE x,R16    ; LH coefficients array addr
        MOVE x,R17    ; HL coefficients array addr
        MOVE x,R19    ; HH coefficients array addr
        MOVE x,A2     ; buffer addr
        MOVE x,A3     ; reconstructed matrix addr

; Вызов функции
        BS IDWT2int_recomp
        NOP
        ...
```

4.11. Функция FDWT2real_decomp

4.11.1. Описание функции

Функция выполняет прямое дискретное необратимое вейвлет-преобразование (FDWT) исходного двумерного массива в соответствии с фильтром Добеши 9/7 (декомпозиция исходной матрицы).

4.11.2. Входные данные

Функция обрабатывает двумерный массив, 32-разрядные значения элементов которого представлены числами в формате с плавающей точкой (float).

Регистр R16 (short): N – размер исходного двумерного массива (N×N элементов), значение N должно быть кратно 4 и находится в диапазоне: 8...32764.
Регистр A4: базовый адрес исходного двумерного массива.
Регистр A3: адрес буфера 1 данных в XRAM.
Регистр I3: адрес буфера 2 данных в XRAM.
Регистр R15 (short): адрес массива LL-коэффициентов вейвлет-преобразования.
Регистр R17 (short): адрес массива LH-коэффициентов вейвлет-преобразования.
Регистр R19 (short): адрес массива HL-коэффициентов вейвлет-преобразования.
Регистр R21 (short): адрес массива HH-коэффициентов вейвлет-преобразования.

4.11.3. Выходные данные

В результате декомпозиции исходного двумерного массива функция формирует четыре двумерных массива коэффициентов вейвлет-преобразования : массив LL-коэффициентов, массив LH-коэффициентов, массив HL-коэффициентов, массив HH-коэффициентов. При этом 32-разрядные значения коэффициентов представлены числами в формате с плавающей точкой (float).

4.11.4. Алгоритм вычисления функции

Прямое вейвлет-преобразование двумерного массива заключается в применении прямого вейвлет-преобразования одномерного массива в соответствии с фильтром Добеши 9/7 сначала ко всем столбцам исходного двумерного массива, то есть фильтрация по вертикали, а затем ко всем строкам полученного двумерного массива, то есть фильтрация по горизонтали. Таким образом, прямое вейвлет-преобразование двумерного массива базируется на использовании функции прямого вейвлет-преобразования одномерного массива в соответствии с фильтром Добеши 9/7 – *FDWTreal*.

4.11.5. Затраты памяти

Функция использует регистры: R14, R23, R25.

PRAM: 26 32-разрядных слов.

XRAM: (N×N) 32-разрядных слов;

буфер_1: (N/2×N) 32-разрядных слов;

буфер_2: (N/2×N) 32-разрядных слов;

4.11.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,R16    ; matrix size
        MOVE x,A4     ; source matrix addr
        MOVE x,A3     ; buffer_1 addr
        MOVE x,I3     ; buffer_2 addr
        MOVE x,R15    ; LL coefficients array addr
        MOVE x,R17    ; LH coefficients array addr
        MOVE x,R19    ; HL coefficients array addr
        MOVE x,R21    ; HH coefficients array addr
; Вызов функции
        BS FDWT2real_decomp
        NOP
        ...
```

4.12. Функция IDWT2real_recomp

4.12.1. Описание функции

Функция выполняет обратное дискретное вейвлет-преобразование (IDWT) двумерного массива в соответствии с фильтром Добеши 9/7 (рекомпозиция исходной матрицы).

4.12.2. Входные данные

Функция обрабатывает четыре двумерных массива коэффициентов вейвлет-преобразования размером N/2×N/2. При этом 32-разрядные значения коэффициентов представлены числами в формате с плавающей точкой (float).

Регистр R16 (short): N – размер восстанавливаемого двумерного массива (N×N элементов), значение N должно быть кратно 4 и находится в диапазоне: 8...32764.
Регистр R15 (short): адрес массива LL-коэффициентов вейвлет-преобразования.
Регистр R17 (short): адрес массива LH-коэффициентов вейвлет-преобразования.
Регистр R19 (short): адрес массива HL-коэффициентов вейвлет-преобразования.
Регистр R21 (short): адрес массива HH-коэффициентов вейвлет-преобразования.
Регистр A3: адрес буфера данных в XRAM.
Регистр A4: адрес восстанавливаемого двумерного массива.

4.12.3. Выходные данные

Функция восстанавливает двумерный массив размером N×N, при этом 32-разрядные значения элементов восстановленного массива представлены числами в формате с плавающей точкой (float).

4.12.4. Алгоритм вычисления функции

При выполнении обратного вейвлет-преобразования двумерного массива четыре исходных массива коэффициентов вейвлет-преобразования объединяются в один двумерный массив, к полученному массиву применяется обратное вейвлет-преобразование одномерного массива в соответствии с фильтром Добеши 9/7 сначала ко всем строкам массива, то есть фильтрация по горизонтали, а затем ко всем столбцам полученного двумерного массива, то есть фильтрация по вертикали. Таким образом, обратное вейвлет-преобразование двумерного массива базируется на использовании функции обратного вейвлет-преобразования одномерного массива в соответствии с фильтром Добеши 9/7 – *IDWTreal*.

4.12.5. Затраты памяти

Функция использует регистры: R14, R23, R25, I3.

PRAM: 27 32-разрядных слов.

XRAM: (N×N) 32-разрядных слов.

буфер: (N×N) 32-разрядных слов;

4.12.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,R16    ; matrix size
        MOVE x,R15    ; LL coefficients array addr
        MOVE x,R17    ; LH coefficients array addr
        MOVE x,R19    ; HL coefficients array addr
        MOVE x,R21    ; HH coefficients array addr
        MOVE x,A3     ; buffer addr
        MOVE x,A4     ; reconstructed matrix addr
; Вызов функции
        BS IDWT2real_recomp
        NOP
        ...
```

4.13. Функция FWTreal_NL_decomp

4.13.1. Описание функции

Функция выполняет N_L-уровневую декомпозицию исходного двумерного массива (фрагмента изображения), то есть N_L уровней прямого дискретного вейвлет-преобразования двумерного массива в соответствии с фильтром Добеши 9/7.

4.13.2. Входные данные

Функция обрабатывает двумерный массив, 32-разрядные значения элементов которого представлены числами в формате с плавающей точкой (float).

Регистр R22 (short): N – размер исходного двумерного массива (N×N элементов), значение N должно быть кратно 2^{N_L} и находится в диапазоне: 8...32764.

Регистр R27 (short): N_L – число уровней декомпозиции.

Регистр A5: базовый адрес исходного двумерного массива / адрес массива коэффициентов вейвлет-преобразования.

Регистр A6: адрес буфера данных в XRAM. Объем буфера равен объему исходного двумерного массива.

4.13.3. Выходные данные

В результате N_L-уровневой декомпозиции исходного двумерного массива функция формирует массив коэффициентов вейвлет-преобразования, состоящий из коэффициентов LL, LH, HL, HH, полученных на каждом уровне декомпозиции. При этом 32-разрядные значения коэффициентов представлены числами в формате с плавающей точкой (float).

4.13.4. Алгоритм вычисления функции

Выполнение N_L уровней прямого вейвлет-преобразования двумерного массива базируется на использовании функции прямого дискретного вейвлет-преобразования двумерного массива в соответствии с фильтром Добеши 9/7 – *FDWT2real_decomp*.
На каждом уровне декомпозиции в результате преобразования двумерного массива получаем четыре поддиапазона коэффициентов вейвлет-преобразования: $LL_{(lev)}$, $LH_{(lev)}$, $HL_{(lev)}$, $HH_{(lev)}$, где lev обозначает уровень декомпозиции, на котором получен данный поддиапазон. На каждом последующем уровне декомпозиции (lev+1) преобразованию подвергается поддиапазон коэффициентов $LL_{(lev)}$, полученный на предыдущем уровне декомпозиции. Общее число поддиапазонов коэффициентов вейвлет-преобразования, полученных в результате выполнения N_L уровней декомпозиции: $(3\times N_L+1)$.

4.13.5. Затраты памяти

Функция использует регистры: R24, R26, R28.

PRAM: 16 32-разрядных слов.

XRAM: (N×N) 32-разрядных слов;

4.13.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,R22     ; matrix size
        MOVE x,R27     ; number of decomposition levels
        MOVE x,A5      ; matrix addr
        MOVE x,A6      ; buffer addr
; Вызов функции
        BS FWTreal_NL_decomp
        NOP
        ...
```

4.14. Функция IWTreal_NL_recomp

4.14.1. Описание функции

Функция выполняет N_L-уровневую рекомпозицию исходного двумерного массива (фрагмента изображения), то есть N_L уровней обратного дискретного вейвлет-преобразования двумерного массива в соответствии с фильтром Добеши 9/7.

4.14.2. Входные данные

Функция обрабатывает массив коэффициентов вейвлет-преобразования, состоящий из коэффициентов LL, LH, HL, HH, соответствующих каждому из N_L уровней рекомпозиции. При этом 32-разрядные значения коэффициентов представлены числами в формате с плавающей точкой (float).

Регистр R22 (short): N – размер восстанавливаемого двумерного массива (N×N элементов), значение N должно быть кратно 2^{N_L} и находится в диапазоне: 8...32764.
Регистр R27 (short): N_L – число уровней рекомпозиции.
Регистр A5: адрес массива коэффициентов вейвлет-преобразования / адрес восстанавливаемого двумерного массива.
Регистр A6: адрес буфера данных в XRAM. Объем буфера равен объему исходного двумерного массива.

4.14.3. Выходные данные

Функция восстанавливает двумерный массив (фрагмент изображения) размером N×N, при этом 32-разрядные значения элементов восстановленного массива представлены числами в формате с плавающей точкой (float).

4.14.4. Алгоритм вычисления функции

Выполнение NL уровней обратного вейвлет-преобразования двумерного массива базируется на использовании функции обратного дискретного вейвлет-преобразования двумерного массива в соответствии с фильтром Добеши 9/7 – *IDWT2real_recomp*.
На каждом уровне рекомпозиции восстанавливается поддиапазон $LL_{(lev-1)}$ из четырех поддиапазонов коэффициентов вейвлет-преобразования: $LL_{(lev)}$, $LH_{(lev)}$, $HL_{(lev)}$, $HH_{(lev)}$, где lev обозначает уровень декомпозиции, которому соответствует данный поддиапазон, а $LL_{(lev)}$ – поддиапазон коэффициентов, восстановленный на предыдущем уровне рекомпозиции. В результате выполнения последнего уровня рекомпозиции получаем восстановливаемый двумерный массив.

4.14.5. Затраты памяти

Функция использует регистры: R24, R26.

PRAM: 14 32-разрядных слов.

XRAM: (N×N) 32-разрядных слов.

4.14.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,R22    ; reconstructed matrix size
        MOVE x,R27    ; number of recomposition levels
        MOVE x,A5     ; matrix addr
        MOVE x,A6     ; buffer addr
; Вызов функции
        BS IWTreal_NL_recomp
        NOP
        ...
```

4.15. Функция FQnt_real_int

4.15.1. Описание функции

Функция выполняет скалярное квантование двумерного массива коэффициентов вейвлет-преобразования в соответствии с заданным шагом квантования.

4.15.2. Входные данные

Функция обрабатывает двумерный массив коэффициентов вейвлет-преобразования, 32-разрядные значения коэффициентов представлены числами в формате с плавающей точкой (float).

Регистр A0: адрес двумерного массива коэффициентов вейвлет-преобразования.

Регистр R18 (short): N – размер двумерного массива коэффициентов (N×N элементов), значение N должно быть четным и находится в диапазоне: 8...32764.

Регистр R20 (short): шаг квантования.

Регистр A1: адрес двумерного массива квантованных коэффициентов вейвлет-преобразования.

4.15.3. Выходные данные

Функция формирует двумерный массив кватованных коэффициентов вейвлет-преобразования. При этом значения квантованных коэффициентов представлены целыми знаковыми 32-разрядными числами (long).

4.15.4. Алгоритм вычисления функции

Каждый коэффициент вейвлет-преобразования WCi массива коэффициентов квантуется в соответствии со следующим выражением:

$$QWCi = \mathrm{sign}(WCi) \cdot \left\lfloor \frac{|WCi|}{\Delta} \right\rfloor$$, где Δ - заданный шаг квантования.

4.15.5. Затраты памяти

Функция использует регистры: R0…R10.

PRAM: 16 32-разрядных слов.

XRAM: (N×N) 32-разрядных слов.

4.15.6. Количество машинных тактов

7 тактов/коэффициент.

4.15.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0      ; source coefficients subband addr
        MOVE x,R18     ; subband size
        MOVE x,R20     ; subband quantization coefficient
        MOVE x,A1      ; quantized coefficients subband addr
; Вызов функции
        BS FQnt_real_int
        NOP
        ...
```

4.16. Функция DQnt_real_int

4.16.1. Описание функции

Функция выполняет скалярное деквантование двумерного массива квантованных коэффициентов вейвлет-преобразования в соответствии с заданным шагом квантования.

4.16.2. Входные данные

Функция обрабатывает двумерный массив квантованных коэффициентов вейвлет-преобразования. При этом значения квантованных коэффициентов представлены целыми знаковыми 32-разрядными числами (long).

Регистр A0: адрес двумерного массива квантованных коэффициентов вейвлет-преобразования.

Регистр R18 (short): N – размер двумерного массива коэффициентов (N×N элементов), значение N должно быть четным и находится в диапазоне: 8...32764.

Регистр R20 (short): шаг квантования.

Регистр A1: адрес двумерного массива восстановленных коэффициентов вейвлет-преобразования.

4.16.3. Выходные данные

Функция восстанавливает двумерный массив коэффициентов вейвлет-преобразования. При этом 32-разрядные значения восстановленных коэффициентов представлены числами в формате с плавающей точкой (float).

4.16.4. Алгоритм вычисления функции

Деквантование каждого из квантованных коэффициентов вейвлет-преобразования QWCi массива коэффициентов выполняется в соответствии со следующим выражением:

$$\begin{cases} WC_i^{REC} = \text{sign}(QWCi) \cdot \left(|QWCi| + 0{,}5 \right) \cdot \Delta, & \text{если} \quad QWCi \neq 0; \\ WC_i^{REC} = 0, & \text{если} \quad QWCi = 0; \end{cases}$$

Где Δ - заданный шаг квантования.

4.16.5. Затраты памяти

Функция использует регистры: R0…R10.

PRAM: 20 32-разрядных слов.

XRAM: N 32-разрядных слов.

4.16.6. Количество машинных тактов

8 тактов/коэффициент.

4.16.7. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0      ; quantized coefficients subband addr
        MOVE x,R18     ; subband size
        MOVE x,R20     ; subband dequantization coefficient
        MOVE x,A1      ; dequantized coefficients subband addr
; Вызов функции
        BS DQnt_real_int
        NOP
        ...
```

4.17. Функция FQnt_real_NL_int

4.17.1. Описание функции

Функция выполняет равномерное скалярное квантование поддиапазонов коэффициентов вейвлет-преобразования, полученных после N_L-уровневой декомпозиции исходного двумерного массива.

4.17.2. Входные данные

Функция обрабатывает массив коэффициентов вейвлет-преобразования, состоящий из коэффициентов LL, LH, HL, HH, соответствующих каждому из N_L уровней декомпозиции. При этом 32-разрядные значения коэффициентов представлены числами в формате с плавающей точкой (float).

Регистр R22 (short): N – размер исходного двумерного массива (N×N элементов), значение N должно быть кратно 2^{N_L} и находится в диапазоне: 8...32764.
Регистр R27 (short): N_L – число уровней декомпозиции.
Регистр R24 (short): коэффициент квантования = 1/Δ, где Δ - шаг квантования.
Регистр A5: адрес массива коэффициентов вейвлет-преобразования / адрес массива квантованных коэффициентов вейвлет-преобразования.

4.17.3. Выходные данные

В результате квантования двумерного массива коэффициентов вейвлет-преобразования функция формирует массив квантованных коэффициентов вейвлет-преобразования, состоящий из квантованных коэффициентов LL, LH, HL, HH, соответствующих каждому уровню декомпозиции. При этом значения квантованных коэффициентов представлены целыми знаковыми 32-разрядными числами (long).

4.17.4. Алгоритм вычисления функции

Поддиапазоны коэффициентов вейвлет-преобразования, полученные после N_L-уровневой декомпозиции исходного двумерного массива подвергаются равномерному скалярному квантованию. При этом коэффициенты вейвлет-преобразования каждого из поддиапазонов квантуется с соответствующим данному поддиапазону шагом квантования. Для квантования каждого из поддиапазонов коэффициентов используется функция скалярного квантования двумерного массива – *FQnt_real_int*.

4.17.5. Затраты памяти

Функция использует регистры: R26, R28, R15, R17, R19, R25.

PRAM: 32 32-разрядных слов.

4.17.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,R22    ; coefficients matrix size
        MOVE x,R27    ; number of decomposition levels
        MOVE x,R24    ; quntization coefficient
        MOVE x,A5     ; coefficients matrix addr
; Вызов функции
        BS FQnt_real_NL_int
        NOP
        ...
```

4.18. Функция DQnt_real_NL_int

4.18.1. Описание функции

Функция выполняет равномерное скалярное деквантование поддиапазонов квантованных коэффициентов вейвлет-преобразования, полученных после N_L-уровневой декомпозиции двумерного массива.

4.18.2. Входные данные

Функция обрабатывает массив квантованных коэффициентов вейвлет-преобразования, состоящий из коэффициентов LL, LH, HL, HH, соответствующих каждому из N_L уровней декомпозиции. При этом значения квантованных коэффициентов представлены целыми знаковыми 32-разрядными числами (long).

Регистр R22 (short): N – размер восстанавливаемого двумерного массива (N×N элементов), значение N должно быть кратно 2^{N_L} и находится в диапазоне: 8...32764.
Регистр R27 (short): N_L – число уровней декомпозиции.
Регистр R24 (short): шаг деквантования = 1/Δ, где Δ - шаг квантования.
Регистр A5: адрес массива квантованных коэффициентов вейвлет-преобразования / адрес массива восстановленных коэффициентов вейвлет-преобразования.

4.18.3. Выходные данные

Функция восстанавливает двумерный массив коэффициентов вейвлет-преобразования, состоящий из коэффициентов LL, LH, HL, HH, соответствующих каждому из N_L уровней декомпозиции. При этом 32-разрядные значения восстановленных коэффициентов представлены числами в формате с плавающей точкой (float).

4.18.4. Алгоритм вычисления функции

Для деквантования поддиапазонов квантованных коэффициентов вейвлет-преобразования, полученных после N_L-уровневой декомпозиции двумерного массива квантованные коэффициенты вейвлет-преобразования каждого из поддиапазонов деквантуется с соответствующим данному поддиапазону шагом деквантования.
Для деквантования каждого из поддиапазонов коэффициентов используется функция скалярного деквантования двумерного массива – *DQnt_real_int*.

4.18.5. Затраты памяти

Функция использует регистры: R26, R28, R15, R17, R19, R25.
PRAM: 26 32-разрядных слов.

4.18.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,R22    ; coefficients matrix size
        MOVE x,R27    ; number of decomposition levels
        MOVE x,R24    ; dequntization step
        MOVE x,A5     ; coefficients matrix addr
; Вызов функции
        BS DQnt_real_NL_int
        NOP
        ...
```

4.19. Функция HENC

4.19.1. Описание функции

Функция выполняет кодирование по Хаффману двумерного массива.

4.19.2. Входные данные

Функция обрабатывает двумерный массив, элементы которого представлены целыми знаковыми 32-разрядными числами (long).

Регистр A0: адрес исходного двумерного массива.

Регистр I0: смещение адреса между соседними элементами массива.

Регистр R18 (short): N – размер двумерного массива (N×N элементов),
значение N должно быть четным и находится в диапазоне: 8...32764.

Регистр R20 (short): адрес таблицы кодирования Хаффмана.

Регистр A1: адрес кодового массива.

4.19.3. Выходные данные

В результате кодирования функция формирует одномерный кодовый массив, состоящий из 32-разрядных кодовых слов.

Регистр R16 (long): N_{CW} – число элементов в кодовом массиве.

4.19.4. Алгоритм вычисления функции

Процедура кодирования состоит из двух последовательно выполняемых операций:

- групповое кодирование;
- кодирование Хаффмана.

При выполнении группового кодирования каждому ненулевому элементу двумерного массива Qi ставится в соответствие пара {ZRL,SIZE(Qi)}, где Qi – значение элемента, ZRL – число нулевых элементов, предшествующих данному ненулевому элементу Qi.

$$-1023 \le Qi \le 1023 ; \qquad 0x0 \le SIZE(Qi) \le 0xA ;$$

$$0x0 \le ZRL \le 0xF ;$$

$$0x00 \le \{ZRL,SIZE(Qi)\} \le 0xFA ;$$

При кодировании Хаффмана каждой паре значений, полученной в результате группового кодирования, ставится в соответствие код Хаффмана. Код Хаффмана является кодом переменной длины, таким образом, последовательно получаемые коды переменной длины объединяются в кодовую последовательность.

4.19.4.1. Формат таблицы кодирования Хаффмана

Таблица кодирования Хаффмана представляет собой последовательность 32-разрядных слов – элементов таблицы кодирования – EncTE[31:0], каждое из которых содержит код Хаффмана и длину кода Хаффмана. Адрес слова в таблице кодирования Хаффмана относительно начала таблицы соответствует значению {ZRL,SIZE(Qi)}, полученному в результате группового кодирования исходного массива.

Если $-1023 \le Qi \le 1023$; то $0x0 \le SIZE(Qi) \le 0xA$;

$0x00 \le \{ZRL,SIZE(Qi)\} \le 0xFA$;

Тогда неиспользуемые элементы таблицы по адресам: 0x0B, … , 0x10, 0x1B, … , 0x20, 0x2B, … , 0x30, 0x3B, … , 0x40, 0x4B, … , 0x50, 0x5B, … , 0x60, 0x6B, … , 0x70, 0x7B, … , 0x80, 0x8B, … , 0x90, 0x9B, … , 0xA0, 0xAB, … , 0xB0, 0xBB, … , 0xC0, 0xCB, … , 0xD0, 0xDB, … , 0xE0, 0xEB, … , 0xEF – заполняются нулевыми значениями.

Адрес элемента EncTE в таблице = {ZRL,SIZE(Qi)}	Значение элемента EncTE таблицы кодирования Хаффмана	
	число бит в коде Хаффмана – EncTE [31:16]	код Хаффмана – EncTE [15:0]
0x00	xxxx	xxxx
0x01	xxxx	xxxx
…	…	…
0x0A	xxxx	xxxx
0x0B	0000	0000
0x0C	0000	0000
0x0D	0000	0000
0x0E	0000	0000
0x0F	0000	0000
0x10	0000	0000
0x11	xxxx	xxxx
…	…	…
0xEA	xxxx	xxxx
0xEB	0000	0000

0xEC	0000	0000
0xED	0000	0000
0xEE	0000	0000
0xEF	0000	0000
0xF0	xxxx	xxxx
0xF1	xxxx	xxxx
…	…	…
0xFA	xxxx	xxxx
0xFB	0000	0000
0xFC	0000	0000
0xFD	0000	0000
0xFE	0000	0000
0xFF	0000	0000

Таблица 1. Формат таблицы кодирования Хаффмана.

В Приложении 1 приведен пример таблицы кодирования Хаффмана.

4.19.5. Затраты памяти

Функция использует регистры: R0...R12, A2.

PRAM: 58 32-разрядных слов.

XRAM: N_{CW} 32-разрядных слов.

4.19.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0      ; source matrix addr
        MOVE x,I0      ; addr step
        MOVE x,R18     ; source matrix size
        MOVE x,R20     ; Haffman_encode_table addr
        MOVE x,A1      ; code_stream addr
; Вызов функции
        BS HENC
        NOP
        MOVE R16,x     ; code_stream length
        ...
```

4.20. Функция HDEC

4.20.1. Описание функции

Функция выполняет декодирование по Хаффману одномерного кодового массива.

4.20.2. Входные данные

Функция обрабатывает одномерный кодовый массив, состоящий из 32-разрядных кодовых слов.

Регистр A0: адрес одномерного кодового массива.

Регистр R18 (short): N – размер восстанавливаемого двумерного массива (N×N элементов), значение N должно быть четным и находится в диапазоне: 8...32764.

Регистр R20 (short): адрес таблицы декодирования Хаффмана.

Регистр A1: адрес восстанавливаемого двумерного массива.

Регистр I1: смещение адреса между соседними элементами массива.

4.20.3. Выходные данные

Функция восстанавливает двумерный массив, элементы которого задаются целыми знаковыми 32-разрядными числами (long).

4.20.4. Алгоритм вычисления функции

Процедура декодирования состоит из двух последовательно выполняемых операций:

- декодирование кода Хаффмана;
- обратное групповое кодирование.

При декодировании элементов двумерного массива коды Хаффмана восстанавливаются в пары значений {ZRL, SIZE(Qi)}, где Qi – значение элемента, ZRL – число нулевых элементов, предшествующих данному ненулевому элементу Qi. Декодированные пары значений {ZRL, SIZE(Qi)} затем при выполнении обратного группового кодирования восстанавливаются в элементы двумерного массива Qi.

Когда $-1023 \le Qi \le 1023$; то $0x0 \le SIZE(Qi) \le 0xA$;

$0x0 \le ZRL \le 0xF$;

$0x00 \le \{ZRL,SIZE(Qi)\} \le 0xFA$;

4.20.4.1. Формат таблицы декодирования Хаффмана

Таблица декодирования Хаффмана формируется на основе таблицы кодирования Хаффмана и представляет собой последовательность 32-разрядных слов – элементов таблицы декодирования – DecTE[31:0], каждое из которых содержит значение SYM = {ZRL,SIZE(Qi)} и длину кода Хаффмана соответствующего данному значению SYM. По значению кода Хаффмана определяется адрес слова в таблице декодирования Хаффмана относительно начала таблицы, которое соответствует данному коду. Элементы DecTE располагаются в таблице по возрастанию длины соответствующих элементам кодов Хаффмана.

Таблица декодирования Хаффмана строится следующим образом:

- по адресу 0x00 – элемент, соответствующий коду 00;
- по адресу 0x01 – элемент, соответствующий коду 01;
- адреса 0x02, … , 0x0F – резерв;
- по адресам 0x10, … , 0x4F – элементы, соответствующие кодам Хаффмана, длина которых < 14 бит (код содержит в старшей части меньше девяти единиц подряд). Смещение OFS[5:0] адреса элемента относительно 0x10 – определяется по значению кода Хаффмана следующим образом:
  старшие 3 разряда смещения адреса OFS[5:3] = [число подряд идущих единиц в старшей части кода – 1]. Для получения младших разрядов смещения адреса OFS[2:0] из старшей части кода Хаффмана необходимо удалить подряд идущие единицы и следующий за ними нулевой бит. Оставшиеся биты кода равны младшей части смещения адреса OFS[2:0]. Если число оставшихся разрядов кода меньше чем 3, то их необходимо дополнить справа до 3 бит разрядами с произвольным значением, то есть, например:
  OFS[2:0] = {HC[1:0], x}. Таким образом, такому коду соответствует несколько адресов в таблице декодирования, определяемых с учетом всех возможных значений произвольных разрядов смещения адреса.
  Когда коду соответствует несколько адресов в таблице, то по каждому из этих адресов должен находиться элемент, соответствующий данному коду.
- адреса 0x50, … , 0x5F – резерв;

– по адресам начиная с 0x60, … – элементы, соответствующие кодам Хаффмана, длина которых 14, 15, 16 бит (код содержит в старшей части не менее девяти единиц подряд, то есть - 111111111xxx…). Смещение адреса элемента относительно 0x60 определяется по значению младших 7 разрядов кода Хаффмана – HC[6:0], при условии, что код 16-разрядный - HC[15:0]. Если код содержит меньше 16 бит, то его необходимо дополнить справа до 16 бит разрядами с произвольным значением. Тогда полученное для такого кода смещение адреса элемента в таблице будет содержать разряды с произвольным значением, например – {HC[4:0], xx}. Таким образом, такому коду соответствует несколько адресов в таблице декодирования, определяемых с учетом всех возможных значений произвольных разрядов смещения адреса.

  Когда коду соответствует несколько адресов в таблице, то по каждому из этих адресов должен находиться элемент, соответствующий данному коду.

Адрес элемента DecTE в таблице	Значение элемента DecTE таблицы декодирования Хаффмана	
	число бит в коде Хаффмана – DecTE [31:16]	значение SYM = {ZRL,SIZE(Qi)}, соответствующее коду Хаффмана
0x00	0002	xxxx
0x01	0002	xxxx
0x02 ÷ 0x0F	0000	0000
0x10	0003	xxxx
0x11	xxxx	xxxx
…	…	…
0x4F	xxxx	xxxx
0x50 ÷ 0x5F	0000	0000
0x60	xxxx	xxxx
0x61	xxxx	xxxx
…	…	…

Таблица 2. Формат таблицы декодирования Хаффмана.

В Приложении 1 приведен пример таблицы декодирования Хаффмана.

4.20.5. Затраты памяти

Функция использует регистры: R0...R13, A2.

PRAM: 74 32-разрядных слов.

XRAM: (N×N) 32-разрядных слов.

4.20.6. Синтаксис

```
        ...
; Передача входных параметров:
        MOVE x,A0      ; code_stream addr
        MOVE x,R18     ; reconstucted matrix size
        MOVE x,R20     ; Haffman_decode_table addr
        MOVE x,A1      ; reconstucted matrix addr
        MOVE x,I1      ; addr step
; Вызов функции
        BS HDEC
        ...
```

5. СООБЩЕНИЯ

Сообщения не выдаются.

6. ОБРАЩЕНИЕ К ПРОГРАММЕ

Функции библиотеки предназначены для использования в программе ядра ELcore платформы МУЛЬТИКОР. Для того чтобы были доступны обращения из программы к функциям библиотеки необходимо к проекту программы подключить библиотеку функций по обработке изображений – libVIDEO.a. Также для корректной компоновки программы необходимо скорректировать скрипт компоновщика – файл .xl

Перед вызовом функции необходимо подготовить соответствующие входные данные.

Вызов функции в программе выполняется инструкцией: `BS <имя функции>.`

7. РЕКОМЕНДАЦИИ ПО ИСПОЛЬЗОВАНИЮ ФУНКЦИЙ БИБЛИОТЕКИ ПРИ НАПИСАНИИ ПРОГРАММ

Для написания программы сжатия изображения на основе вейвлет-преобразования в соответствии с фильтром Добеши 9/7 рекомендуется использовать функции библиотеки в соотвествии со схемой представленной на рисунке 1:

Рисунок 1. Программа сжатия изображения.

Для написания программы восстановления ранее сжатого изображения на основе вейвлет-преобразования в соответствии с фильтром Добеши 9/7 рекомендуется использовать функции библиотеки в соотвествии со схемой представленной на рисунке 2:

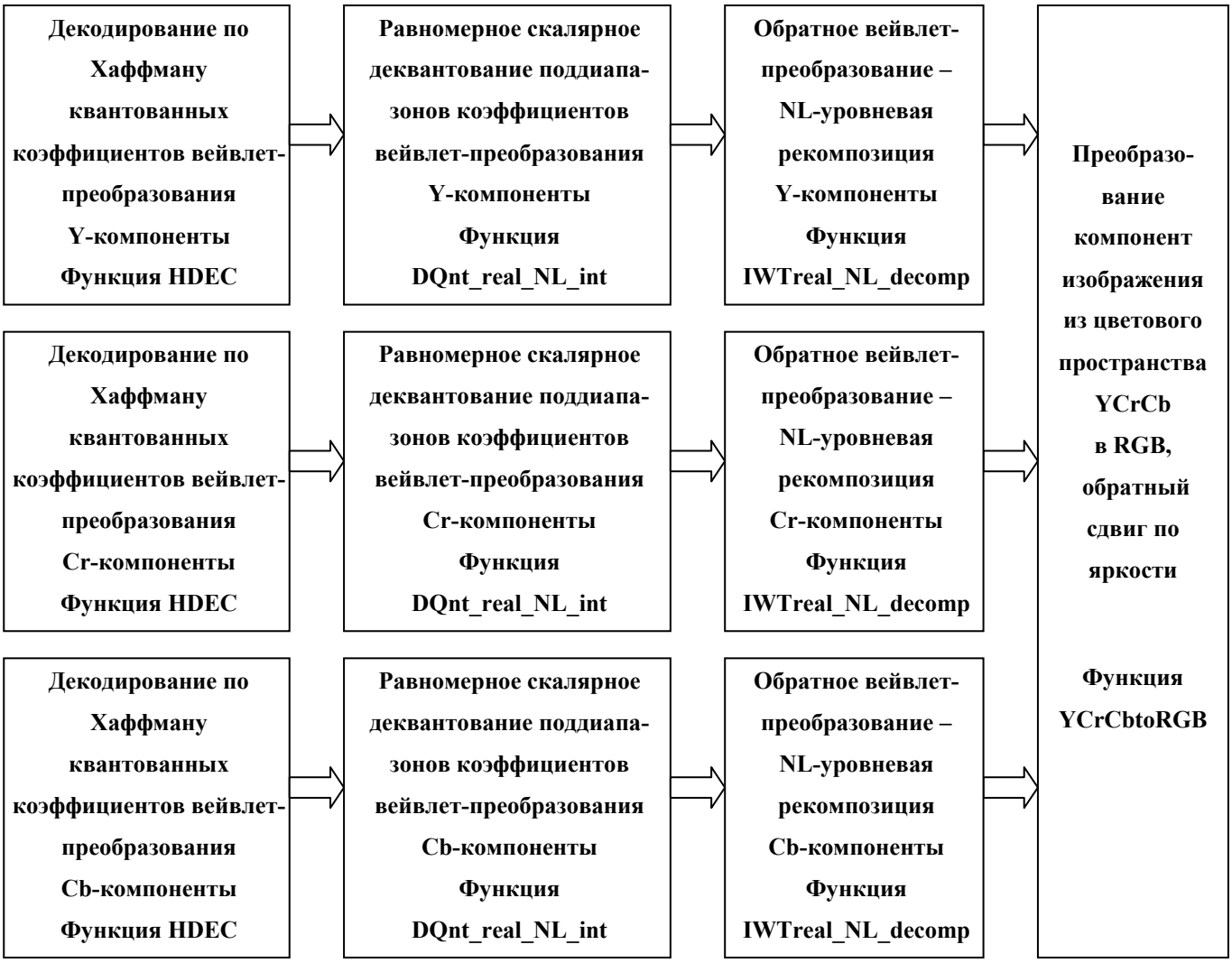

Рисунок 2. Программа восстановления сжатого изображения.

8. ПРИЛОЖЕНИЕ 1

Ниже приведен пример таблицы кодирования Хаффмана.

Адрес элемента EncTE в таблице	Значение элемента EncTE таблицы кодирования Хаффмана
00	00020000
01	00020001
02	00030004
03	0004000A
04	00050018
05	00050019
06	00060038
07	00070078
08	000901F4
09	000A03F6
0A	000C0FF4
0B	00000000
0C	00000000
0D	00000000
0E	00000000
0F	00000000
10	00000000
11	0004000B
12	00060039
13	000800F6
14	000901F5
15	000B07F6
16	000C0FF5
17	0010FF88
18	0010FF89
19	0010FF8A
1A	0010FF8B
1B	00000000
1C	00000000
1D	00000000
1E	00000000
1F	00000000
20	00000000
21	0005001A
22	000800F7
23	000A03F7
24	000C0FF6
25	000F7FC2
26	0010FF8C
27	0010FF8D
28	0010FF8E
29	0010FF8F
2A	0010FF90
2B	00000000
2C	00000000
2D	00000000

2E	00000000
2F	00000000
30	00000000
31	0005001B
32	000800F8
33	000A03F8
34	000C0FF7
35	0010FF91
36	0010FF92
37	0010FF93
38	0010FF94
39	0010FF95
3A	0010FF96
3B	00000000
3C	00000000
3D	00000000
3E	00000000
3F	00000000
40	00000000
41	0006003A
42	000901F6
43	0010FF97
44	0010FF98
45	0010FF99
46	0010FF9A
47	0010FF9B
48	0010FF9C
49	0010FF9D
4A	0010FF9E
4B	00000000
4C	00000000
4D	00000000
4E	00000000
4F	00000000
50	00000000
51	0006003B
52	000A03F9
53	0010FF9F
54	0010FFA0
55	0010FFA1
56	0010FFA2
57	0010FFA3
58	0010FFA4
59	0010FFA5
5A	0010FFA6
5B	00000000
5C	00000000
5D	00000000
5E	00000000
5F	00000000
60	00000000
61	00070079
62	000B07F7

63	0010FFA7
64	0010FFA8
65	0010FFA9
66	0010FFAA
67	0010FFAB
68	0010FFAC
69	0010FFAD
6A	0010FFAE
6B	00000000
6C	00000000
6D	00000000
6E	00000000
6F	00000000
70	00000000
71	0007007A
72	000B07F8
73	0010FFAF
74	0010FFB0
75	0010FFB1
76	0010FFB2
77	0010FFB3
78	0010FFB4
79	0010FFB5
7A	0010FFB6
7B	00000000
7C	00000000
7D	00000000
7E	00000000
7F	00000000
80	00000000
81	000800F9
82	0010FFB7
83	0010FFB8
84	0010FFB9
85	0010FFBA
86	0010FFBB
87	0010FFBC
88	0010FFBD
89	0010FFBE
8A	0010FFBF
8B	00000000
8C	00000000
8D	00000000
8E	00000000
8F	00000000
80	00000000
91	000901F7
92	0010FFC0
93	0010FFC1
94	0010FFC2
95	0010FFC3
96	0010FFC4
97	0010FFC5

98	0010FFC6
99	0010FFC7
9A	0010FFC8
9B	00000000
9C	00000000
9D	00000000
9E	00000000
9F	00000000
A0	00000000
A1	000901F8
A2	0010FFC9
A3	0010FFCA
A4	0010FFCB
A5	0010FFCC
A6	0010FFCD
A7	0010FFCE
A8	0010FFCF
A9	0010FFD0
AA	0010FFD1
AB	00000000
AC	00000000
AD	00000000
AE	00000000
AF	00000000
A0	00000000
B1	000901F9
B2	0010FFD2
B3	0010FFD3
B4	0010FFD4
B5	0010FFD5
B6	0010FFD6
B7	0010FFD7
B8	0010FFD8
B9	0010FFD9
BA	0010FFDA
BB	00000000
BC	00000000
BD	00000000
BE	00000000
BF	00000000
C0	00000000
C1	000901FA
C2	0010FFDB
C3	0010FFDC
C4	0010FFDD
C5	0010FFDE
C6	0010FFDF
C7	0010FFE0
C8	0010FFE1
C9	0010FFE2
CA	0010FFE3
CB	00000000
CC	00000000

CD	00000000
CE	00000000
CF	00000000
D0	00000000
D1	000B07F9
D2	0010FFE4
D3	0010FFE5
D4	0010FFE6
D5	0010FFE7
D6	0010FFE8
D7	0010FFE9
D8	0010FFEA
D9	0010FFEB
DA	0010FFEC
DB	00000000
DC	00000000
DD	00000000
DE	00000000
DF	00000000
E0	00000000
E1	000E3FE0
E2	0010FFED
E3	0010FFEE
E4	0010FFEF
E5	0010FFF0
E6	0010FFF1
E7	0010FFF2
E8	0010FFF3
E9	0010FFF4
EA	0010FFF5
EB	00000000
EC	00000000
ED	00000000
EE	00000000
EF	00000000
F0	000A03FA
F1	000F7FC3
F2	0010FFF6
F3	0010FFF7
F4	0010FFF8
F5	0010FFF9
F6	0010FFFA
F7	0010FFFB
F8	0010FFFC
F9	0010FFFD
FA	0010FFFE

Ниже приведен пример таблицы декодирования Хаффмана.

Адрес элемента DecTE в таблице	Значение элемента DecTE таблицы кодирования Хаффмана
00	00020000
01	00020001
02	00000000
03	00000000
04	00000000
05	00000000
06	00000000
07	00000000
08	00000000
09	00000000
0A	00000000
0B	00000000
0C	00000000
0D	00000000
0E	00000000
0F	00000000
10	00030002
11	00030002
12	00030002
13	00030002
14	00040003
15	00040003
16	00040011
17	00040011
18	00050004
19	00050004
1A	00050005
1B	00050005
1C	00050021
1D	00050021
1E	00050031
1F	00050031
20	00060006
21	00060006
22	00060012
23	00060012
24	00060041
25	00060041
26	00060051
27	00060051
28	00070007
29	00070007
2A	00070061
2B	00070061
2C	00070071

2D	00070071
2E	00080013
2F	00080022
30	00080032
31	00080032
32	00080081
33	00080081
34	00090008
35	00090014
36	00090042
37	00090091
38	000900A1
39	000900A1
3A	000900B1
3B	000900B1
3C	000900C1
3D	000900C1
3E	000A0009
3F	000A0023
40	000A0033
41	000A0033
42	000A0052
43	000A0052
44	000A00F0
45	000A00F0
46	000B0015
47	000B0062
48	000B0072
49	000B0072
4A	000B00D1
4B	000B00D1
4C	000C000A
4D	000C0016
4E	000C0024
4F	000C0034
50	00000000
51	00000000
52	00000000
53	00000000
54	00000000
55	00000000
56	00000000
57	00000000
58	00000000
59	00000000
5A	00000000
5B	00000000
5C	00000000

5D	00000000
5E	00000000
5F	00000000
60	000E00E1
61	000E00E1
62	000E00E1
63	000E00E1
64	000F0025
65	000F0025
66	000F00F1
67	000F00F1
68	00100017
69	00100018
6A	00100019
6B	0010001A
6C	00100026
6D	00100027
6E	00100028
6F	00100029
70	0010002A
71	00100035
72	00100036
73	00100037
74	00100038
75	00100039
76	0010003A
77	00100043
78	00100044
79	00100045
7A	00100046
7B	00100047
7C	00100048
7D	00100049
7E	0010004A
7F	00100053
80	00100054
81	00100055
82	00100056
83	00100057
84	00100058
85	00100059
86	0010005A
87	00100063
88	00100064
89	00100065
8A	00100066
8B	00100067
8C	00100068

8D	00100069
8E	0010006A
8F	00100073
80	00100074
91	00100075
92	00100076
93	00100077
94	00100078
95	00100079
96	0010007A
97	00100082
98	00100083
99	00100084
9A	00100085
9B	00100086
9C	00100087
9D	00100088
9E	00100089
9F	0010008A
A0	00100092
A1	00100093
A2	00100094
A3	00100095
A4	00100096
A5	00100097
A6	00100098
A7	00100099
A8	0010009A
A9	001000A2
AA	001000A3
AB	001000A4
AC	001000A5
AD	001000A6
AE	001000A7
AF	001000A8
A0	001000A9
B1	001000AA
B2	001000B2
B3	001000B3
B4	001000B4
B5	001000B5
B6	001000B6
B7	001000B7
B8	001000B8
B9	001000B9
BA	001000BA
BB	001000C2
BC	001000C3

BD	001000C4
BE	001000C5
BF	001000C6
C0	001000C7
C1	001000C8
C2	001000C9
C3	001000CA
C4	001000D2
C5	001000D3
C6	001000D4
C7	001000D5
C8	001000D6
C9	001000D7
CA	001000D8
CB	001000D9
CC	001000DA
CD	001000E2
CE	001000E3
CF	001000E4
D0	001000E5
D1	001000E6
D2	001000E7
D3	001000E8
D4	001000E9
D5	001000EA
D6	001000F2
D7	001000F3
D8	001000F4
D9	001000F5
DA	001000F6
DB	001000F7
DC	001000F8
DD	001000F9
DE	001000FA
DF	00000000

Лист регистрации изменений									
Изм	Номера листов (страниц)				Всего листов (страниц) в докум.	N документа	Входящий N сопрово-дительного докум.	Подп.	Дата
	измененных	замененных	новых	аннулированных					