

# Быстрое прототипирование ДКП-процессора для встраиваемых систем реального времени

Ключеня В.В.  
Кафедра ЭВС, ФКП, БГУИР  
Минск, Беларусь  
e-mail: lucky\_victor@rambler.ru

**Аннотация** – в данной работе предлагается FPGA (Field Programmable Gate Array) реализация процессора ДКП (дискретного косинусного преобразования) на основе лифтинг шагов. Алгоритм 16-точечного ДКП представляется в виде произведения матриц и отображается на архитектуру FPGA виде многоступенчатой конвейерной схемы. Используя различные битовое представление коэффициентов лифтинг шагов, можно быстро реализовать прототипы ДКП-процессора. Методом множителей Лагранжа можно выбрать оптимальную структуру ДКП-процессора по таким параметрам как площадь кристалла, точность, отношение сигнал-шум, частота и т. д.

**Ключевые слова:** ДКП, лифтинг шаги

## I. ВВЕДЕНИЕ

На сегодняшний день широко распространены мобильные мультимедийные системы, которые используют стандарты H.261/3/4/5, MPEG-1/2/4 и JPEG для кодирования/декодирования видео, аудио и изображений. Ядром этих стандартов является дискретное косинусное преобразование (ДКП) -I, -II, -III – -VIII типа. ДКП<sub>II</sub> чаще всего используется в данных системах.

Вычислительная сложность ДКП очень важна для встраиваемых систем мультимедиа реального времени. Поэтому актуальной задачей является разработка быстрых алгоритмов ДКП, которые позволяют уменьшить число умножений или избавиться от них. Одним из наиболее эффективных методов реализации ДКП является алгоритм Лофлера [1], который представляет 8-точечное

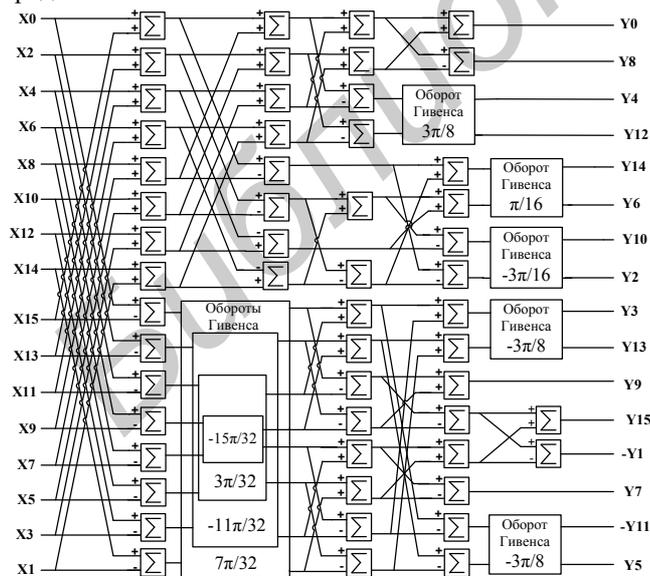


Рис.1. Модифицированная схема Лофлера 16-точечного ДКП<sub>II</sub> или 16-точечное ДКП (16-ДКП<sub>II</sub>) в виде схемы с минимальным количеством умножений. В данной статье рассматривается реализация 16-ДКП<sub>II</sub>. Постоянные коэффициенты ДКП могут быть вынесены в блок квантования в кодерах и схема Лофлера может представлена как показано на рис.1. Основная

операция, которая вызывает трудности при аппаратной реализации, является операция вращения или оборот Гивенса (“Givens rotation”). Математически эта операция может быть записана как умножение вектора на матрицу.

Одной из распространенных реализаций оборота Гивенса в схеме Лофлера является аппроксимация значений косинусов и синусов к числам, легко получаемым путем небольшого количества операций сдвига и сложения, что позволяет избежать операций умножения и тем самым повысить эффективность вычисления.

## II. МАТРИЧНОЕ И АРХИТЕКТУРНОЕ ПРЕДСТАВЛЕНИЕ

Матрицу оборотов Гивенса, используя лифтинг шаги, можно разложить следующим образом

$$\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} = \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ u & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix}, \quad (1)$$

где  $p = \frac{\cos\alpha - 1}{\sin\alpha}$ ,  $u = \sin\alpha$ . На рис.2 показана

структура оборота Гивенса, используя лифтинг шаги.

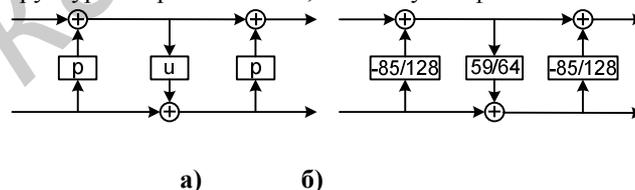


Рис.2. Реализация прямого (а) оборота Гивенса на лифтинг шагах, целочисленное представление (б) оборота Гивенса на лифтинг шагах

Подставляя структуры оборотов Гивенса в схему ДКП (рис.1), получаем 16-ДКП<sub>II</sub> на основе лифтинг шагов. Для эффективной реализации на FPGA 16-ДКП<sub>II</sub> можно разложить на одиннадцать простых матриц. Каждая матрица является отдельным вычислительным шагом и вся схема 16-ДКП<sub>II</sub> представляется в виде конвейерной архитектуры [2]. Матричная факторизация 16-ДКП<sub>II</sub> на основе лифтинг шагов имеет вид:

$$16\text{-ДКП}_{II} = K \cdot J \cdot I \cdot H \cdot G \cdot F \cdot E \cdot D \cdot C \cdot B \cdot A, \quad (2)$$

где A, B, C, D, E, F, G, H, I, J и K являются разреженными матрицами. В целом, архитектура каждой ступени содержит два набора регистров и специальные арифметические модули для вычисления матричных умножений. Каждая ступень (рис.3) имеет 16 входов и 16 выходов, а также имеет свои управляющие сигналы. Коэффициенты всех матриц представляются в двоичном коде, матричные умножения можно представить виде сдвигов и суммирования, операции суммирования и вычитания выполняются в дополнительном коде.

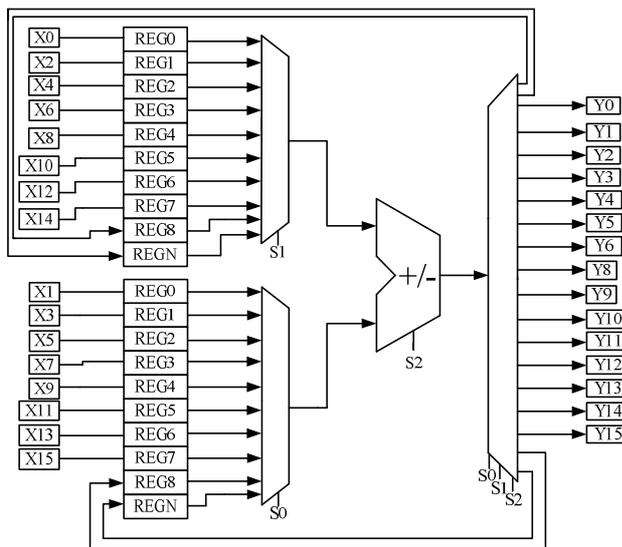


Рис.3. Обобщенная схема архитектуры каждой ступени.

С использованием VHDL-описаний был проведен синтез систем 16-ДКП<sub>II</sub> в среде Xilinx ISE 10.1 (табл.1), а также в MATLAB были промоделированы данные структуры в матричном представлении для монохромных изображений и подсчитано среднеквадратическое отклонение (MSE) и отношение сигнал-шум (PSNR).

Табл.1. Синтез 16-ДКП<sub>II</sub>-Nbit на FPGA Spartan3 xc3s1500I-4fg676

	Slices	PSNR, dB	MSE	Frequency,MHz
16-ДКП <sub>II</sub> -8bit	4173	73.5919	$2.8437 \cdot 10^{-3}$	154.369
16-ДКП <sub>II</sub> -7bit	3893	73.5907	$2.8445 \cdot 10^{-3}$	177.153
16-ДКП <sub>II</sub> -6bit	3625	73.5811	$2.8508 \cdot 10^{-3}$	177.580
16-ДКП <sub>II</sub> -5bit	3153	73.545	$2.8746 \cdot 10^{-3}$	181.263
16-ДКП <sub>II</sub> -4bit	3037	73.4168	$2.9607 \cdot 10^{-3}$	180.660
16-ДКП <sub>II</sub> -3bit	2909	73.0671	$3.209 \cdot 10^{-3}$	179.175
16-ДКП <sub>II</sub> -2bit	2722	71.6302	$4.4674 \cdot 10^{-3}$	180.660
16-ДКП <sub>II</sub> -1bit	2682	69.6754	$7.0072 \cdot 10^{-3}$	180.660

### III. ВЫБОР ОПТИМАЛЬНОГО СТРУКТУРНОГО РЕШЕНИЯ

Проанализировав важнейшие характеристики полученных схем, задачу поиска оптимального решения можно сформулировать следующим образом: найти такое значение N-бит коэффициентов лифтинг шагов, при которых схема 16-ДКП<sub>II</sub> имела бы минимальное MSE, минимальную площадь кристалла (Slices), максимальное PSNR и максимальную частоту (Frequency).

Так как в основном все данные экспериментальные и нам надо найти экстремум некой целевой функции, то производную такой функций будет найти сложно. Следовательно, решать данную проблему надо

методом, который не требует нахождения производной для поиска минимума целевой функции, например, одним из методов нелинейного программирования – методом множителей Лагранжа [3]. Задача минимизации может быть сформулирована следующим образом:

$$\text{найти минимум функции } f(N) = 2^{-N} \quad (3)$$

при заданных ограничениях:

$$MSE(N) - MSE_{\min} \geq 0; Slices(N) - Slices_{\min} \geq 0; \quad (4)$$

$$Freq(N) - Freq_{\max} \leq 0; PSNR(N) - PSNR_{\max} \leq 0;$$

Целевая функция представляется формулой (3), которая использует оптимизационную переменную  $2^{-N}$ , все значения берутся из табл.1. Составляем Лагранжиан:

$$L(N) = ef(N) + \lambda_1 \max(0, g_1) + \lambda_2 \max(0, g_2) + \lambda_3 \max(0, g_3) + \lambda_4 \max(0, g_4) \quad (5)$$

где  $\lambda_1, \lambda_2, \lambda_3, \lambda_4$  – множители Лагранжа,  $e$  – масштабирующий коэффициент ( $0 < e < 1$ ).

Подсчитав Лагранжиан для каждого  $N_i$ , где  $i$  от 8 до 1 бита, мы получаем локальный минимум при  $N_4 = 5$ -бит, то есть схема, в которой лифтинг шаги представляются 5-ю битами является оптимальным структурным решением.

### IV. ЗАКЛЮЧЕНИЕ

В данной работе исследуются реализации процессора 16-точечного ДКП II-типа на основе метода лифтинг шагов. Приведены результаты синтеза схем с различным битовым представлением лифтинг шагов. А также, используя один из методов нелинейной программирования – метод множителей Лагранжа, показано, что схема 16-ДКП<sub>II</sub>-5bit является оптимальной для обработки монохромных изображений в реальном времени из всех реализованных прототипов ДКП-процессора. В итоге, используя данный подход, можно найти компромиссное структурное решение процессора ДКП среди заданных параметров схемы, таких как точность вычисления (MSE), занимаемая площадь кристалла, отношение сигнал-шум, частота и т. д.

[1] Loeffler C., Ligtenberg A., Moshy G.S., "Practical fast 1-D DCT algorithms with 11 multiplications" Proc. ICASSP. 1989, pp. 998-991.

[2] Philip P. Dang, Paul M. Chau and Truong Q. Nguyen, Trac D. Tran, "BinDCT and Its Efficient VLSI Architectures for Real-Time Embedded Applications" Journal of imaging science and technology, Volume 49, Number 2, March/April 2005.

[3] T. W. Fox, L. E. Turner, "Rapid Prototyping of Field Programmable Gate Array-Based Discrete Cosine Transform Approximations," EURASIP Journal on Applied Signal Processing, pp. 543-554, 2003:6.