

Белорусский государственный университет информатики и радиоэлектроники, ул.  
П.Бровки 6, Минск, Республика Беларусь  
Кафедра электронных вычислительных средств, e-mail: [palex@it.org.by](mailto:palex@it.org.by)

**Реферат.** В работе рассматриваются возможности формализации синтеза параллельно-поточных структур процессоров цифровой обработки сигналов. Предлагается алгоритм структурного синтеза процессора, обеспечивающий достижение обработки сигнала в реальном времени при изменении параметров потока входных данных (частоты дискретизации, размерности) и для различных реализаций вычислительного элемента базовой операции алгоритма БПФ. Особое внимание уделяется анализу реализации в виде параллельно-поточной структуры алгоритма БПФ по расщепленному основанию 2-4. Предложенное описание структуры процессора БПФ характеризуется легкостью аппаратной реализации на микросхемах программируемой логики (FPGA) и позволяет автоматизировать процесс синтеза процессоров БПФ для произвольного потока входных данных.

В последнее время появилось много новых направлений развития цифровой обработки сигналов, таких как синтез и кодирование речи, управление голосом, безтрубочные телефоны и другие [1]. Перечисленные мультимедиа приложения, требуют обеспечения обработки многомерных сигналов в реальном масштабе времени. Большая часть алгоритмов цифровой обработки сигналов, так или иначе, использует вычисление дискретного преобразования Фурье (ДПФ). Более того, качество обработки сигнала сильно зависит от пропускной способности системы. Поэтому эффективная реализация вычисления векторного ДПФ приобретает большое значение.

### 1. Алгоритм БПФ по расщепленному основанию

Среди множества алгоритмов вычисления ДПФ наиболее распространены алгоритмы БПФ по постоянным основаниям, поскольку они обеспечивают рекурсивность, регулярность вычислений, а так же возможности гибкого изменения формата преобразования. В последнее время были предложены алгоритмы вычисления ДПФ, обеспечивающие большую эффективность вычислений по сравнению с алгоритмами БПФ по постоянному основанию [2],[3]. Так алгоритм БПФ по расщепленному основанию 2-4 [4] базируется на следующем разложении основной формулы вычисления ДПФ:

$$\begin{aligned}
 X_k &= \sum_{n=0}^{N-1} x_n W^{nk} = \\
 X_{2k} &= \sum_{n=0}^{N/2-1} (x_n + x_{n+N/2}) W^{2nk} ; \\
 X_{4k+1} &= \sum_{n=0}^{N/4-1} [(x_n - x_{n+N/2}) + j(x_{n+N/4} - x_{n+3N/4})] W^n W^{4nk} ; \\
 X_{4k+3} &= \sum_{n=0}^{N/4-1} [(x_n - x_{n+N/2}) - j(x_{n+N/4} - x_{n+3N/4})] W^{3n} W^{4nk}
 \end{aligned} \quad (1)$$

Подобная схема организации вычислений позволяет значительно уменьшить вычислительную сложность алгоритма [3]. Уменьшение сложности сопровождается сохранением многих черт алгоритма БПФ по основанию 2. В частности, на всех этапах алгоритма БПФ по расщепленному основанию используется базовая операция, аналогичная базовой операции алгоритма БПФ по основанию 4. Кроме того, сохраняется возможность расчета ДПФ последовательностей длины  $2^n$ . Тем не менее, управление передачей данных при реализации этого алгоритма затруднено вследствие нерегулярности его этапов, что связано с использованием базовой операции L-формы.

В работе [5] было продемонстрировано, что простая перестановка выходов по выполнению базовой операции алгоритма БПФ по расщепленному основанию 2-4 (см. Рисунок 1) может значительно улучшить регулярность графа этого алгоритма. Перестановка выходов приводит к образованию так называемой базовой операции С-формы. С точки зрения аппаратной реализации, использование модифицированной базовой операции позволяет создать универсальный вычислительный элемент, который может применяться на любых этапах алгоритма БПФ по расщепленному основанию.

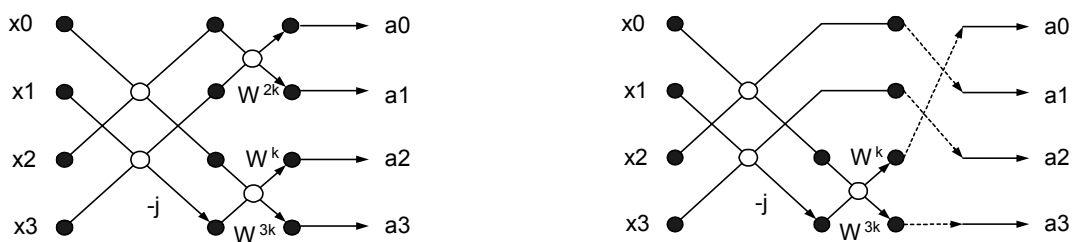


Рисунок 1. Базовые операции алгоритмов БПФ по основанию 4 (слева) и по расщепленному основанию 2-4 с перестановкой выходов (справа)

**2. Параллельно-поточная структура алгоритма БПФ по расщепленному основанию**

Обобщенная структура однопоточного процессора алгоритма БПФ по расщепленному основанию представлена на Рисунке 2 [5]. Вычислительные элементы (Computing Element - CE) соединяются с использованием элементов межпроцессорной связи (Interconnection Element - IE). Последние обеспечивают необходимый порядок поступления данных на вычислительные элементы. Внутренняя структура вычислительного элемента подразумевает параллельное выполнение операций умножения текущего этапа алгоритма и операций сложения-вычитания последующего этапа вычислений, что отражено на Рисунке 2. На последнем этапе алгоритма выполняются только базовые операции алгоритма БПФ по основанию 2. В силу описанных особенностей однопоточной структуры, входной и выходной этап вычислений отличаются от промежуточных, что отражается в различиях внутренней структуры соответствующих вычислительных элементов. Тем ни менее, подобный однопоточный процессор алгоритма БПФ по расщепленному основанию 2-4 может стать основой для построения переконфигурируемого векторного процессора для обработки многомерных данных.

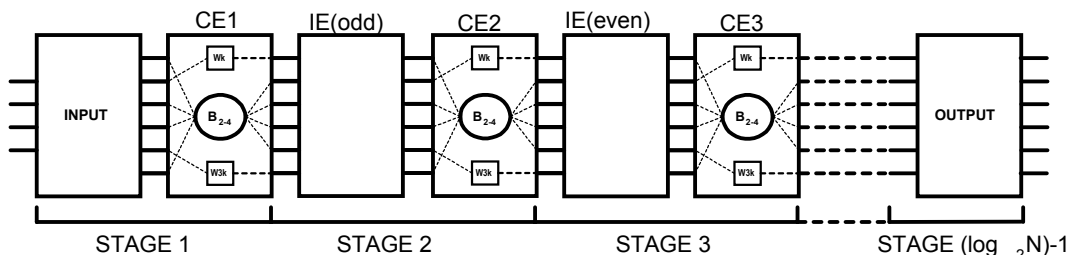


Рисунок 2. Структура однопоточного процессора, реализующего алгоритм БПФ по расщепленному основанию 2-4 для преобразования длины N

Предположим, что количество конвейеров в параллельно-поточной структуре, реализующей алгоритм БПФ по расщепленному основанию равно  $m$ , и  $m$  является степенью числа 2. В этом случае, до определенного этапа, алгоритм обеспечивает независимую обработку  $m$  потоков данных. Начиная с этапа  $\log_2 N/m - 1$  и заканчивая этапом, предшествующим выходному, необходимо осуществлять предварительный обмен данными между потоками. Другими словами,  $\log_2 m$  этапов в параллельно-поточной структуре (от этапа с номером  $(\log_2 N/m) - 1$  и до этапа с номером  $(\log_2 N) - 2$ ) начинают свою работу с выполнения процедуры межпоточной коммутации, которая определяется графом модифицированного алгоритма БПФ по расщепленному основанию 2-4 и обеспечивает правильный порядок поступления результатов работы предыдущих этапов на элементы межпроцессорной связи.

Рассматривая наиболее простой случай двух конвейеров ( $m=2$ ), становится очевидно, что дополнительная межпоточная коммутация необходима перед этапом номер  $\log_2 N/2 - 1$ . Другими словами, - на этапе, предшествующем выходному. Пусть  $i=0$  - это номер первого конвейера, а  $j=1$  - это номер второго конвейера. Процедура межпоточной коммутации может быть определена как преобразование вектора  $O = [O_{i,0}, O_{i,1}, O_{i,2}, O_{i,3}, O_{i,4}, O_{i,5}, O_{j,0}, O_{j,1}, O_{j,2}, O_{j,3}, O_{j,4}, O_{j,5}]$ , представляющего значения на выходе предыдущего этапа алгоритма в вектор значений, поступающих на вход последующего этапа  $I = [I_{i,0}, I_{i,1}, I_{i,2}, I_{i,3}, I_{i,4}, I_{i,5}, I_{j,0}, I_{j,1}, I_{j,2}, I_{j,3}, I_{j,4}, I_{j,5}]$ . Соответствующая коммутация определяется графом модифицированного алгоритма БПФ по расщепленному основанию 2-4 и изображена на Рисунке 3.

При разбиении вычислений на несколько параллельных потоков можно установить, что граф модифицированного алгоритма БПФ по расщепленному основанию обладает следующими особенностями:

- 1) На любом этапе выполнения алгоритма передачи данных происходят исключительно между парами потоков.
- 2) Схема передачи данных внутри любой пары потоков всегда постоянна и не изменяется от этапа к этапу алгоритма.

Эти особенности графа модифицированного алгоритма БПФ по расщепленному основанию 2-4 сводят проблему формализации процесса синтеза соответствующей параллельно-поточной структуры к задаче

определения пар потоков, между которыми будет происходить передача данных на каждом из этапов алгоритма. Пусть  $s$  - это номер этапа, на котором производится определение пар потоков для передач данных, а  $i$  и  $j$  - номера этих потоков. Тогда обобщенная процедура межпоточной коммутации на каждом из этапов алгоритма БПФ по расщепленному основанию определяется следующей формулой:

$$j=i+N/2^{s+1}, \quad \text{где } i \in [0, 1, 2, \dots, (N/2^{s+1}-1)] \text{ и } \log_2 N/m-1 \leq s < (\log_2 N)-1 \quad (2)$$

Выражение 2 определяет возможность создания параллельно-поточных структур с произвольным количеством конвейеров при реализации алгоритма БПФ по расщепленному основанию 2-4.

Аппаратная реализация процедуры межпоточной коммутации возможна с использованием буферной памяти. Для перестановки данных внутри ограниченного количества потоков достаточно простых схем, построенных на регистрах. Но высокие степени параллелизма вычислений могут потребовать использования двухпортовой памяти.

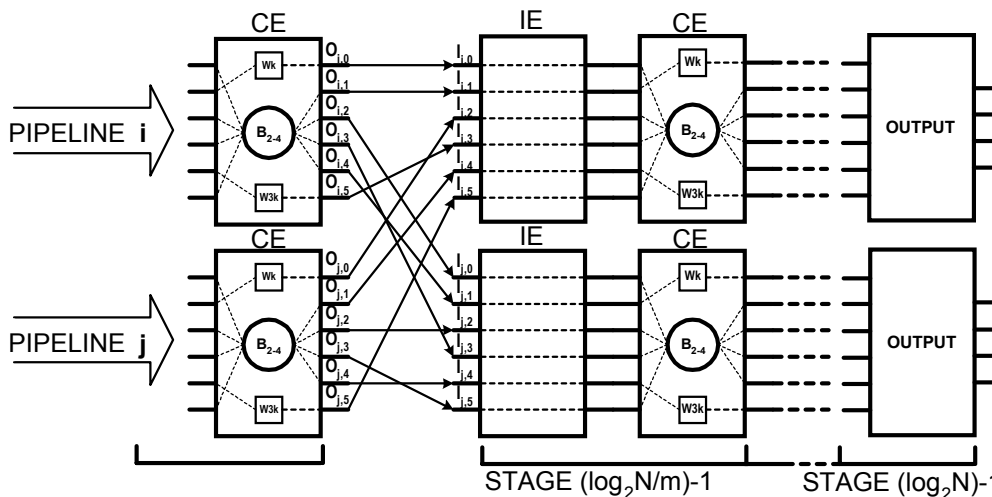


Рисунок 3. Реализация процедуры межпоточной коммутации для произвольной пары потоков параллельно-поточной структуры с  $m$  вычислительными элементами на этапе при вычислении преобразования длины  $N$ .

### 3. Алгоритм структурного синтеза ППБПФ

В [6],[7] был предложен подход к универсальному описанию параллельно-поточного БПФ-процессора. Полученные результаты позволяют использовать эту методику для реализации БПФ-процессоров по расщепленному основанию 2-4 в виде параллельно-поточных структур.

Основным свойством процессоров реального времени является необходимость обработки непрерывного потока данных. Выполнение указанного условия достигается, если удастся решить две основные проблемы структурного синтеза:

1) Определение необходимого количества вычислительных элементов на каждом этапе алгоритма, которое может обеспечить вычисление векторного ДПФ входного потока данных в реальном времени при минимальной структурной сложности процессора.

2) Определение структуры БПФ-процессора для данного количества вычислительных элементов на каждом этапе алгоритма.

Определим переменную  $T_B$  как  $\max [2T_a, T_m]$ , где  $T_a$  - время выполнения операций сложения-вычитания, а  $T_m$  - время затрачиваемое на выполнение умножения на поворачивающий множитель. Обобщенная параллельно-поточная структура (см. Рисунок 3), определенная в соответствии с формулой (2) позволяет выполнить расчет минимального количества вычислительных элементов на каждом этапе алгоритма, которое обеспечит достижение обработки заданного потока данных в реальном масштабе времени:

$$m_{\min} = \begin{cases} 1, r = 0, \text{ if } DT_B < 2\Delta t, \\ 2^r, r = \lceil \log(DT_B / 2\Delta t) \rceil, \text{ if } DT_B \geq 2\Delta t, \end{cases} \quad (3)$$

где  $\lceil a \rceil$  - ближайшее целое, большее или равное  $a$ ;  $T_B$  - время выполнения базовой операции алгоритма БПФ;  $\Delta t$  - период дискретизации;  $m$  - количество вычислительных элементов на этапе алгоритма;  $D$  - количество компонент векторного процесса;  $N$  - размер компоненты.

При известном количестве конвейеров, структура параллельно-поточного процессора легко определяется на основании соотношений, представленных в предыдущих разделах.

#### Выводы

Гибкость получаемых аппаратных решений может быть улучшена за счет выбора соответствующего аппаратного базиса. Предлагаемое исполнение алгоритма БПФ по расщепленному основанию характеризуется небольшим количеством логических операций, но требует использования элементов памяти. Поэтому, выбор в качестве аппаратного базиса микросхем программируемой логики FPGA представляется достаточно оправданным. Универсальный алгоритм синтеза и полученное общее описание параллельно-поточной структуры позволяют создавать переконфигурируемые процессоры, которые могут встраиваться в широкий набор устройств, требующих вычисления многомерного дискретного преобразования Фурье в реальном масштабе времени.

#### Литература

1. Mark Cummings, Shinichiro Haruyama. "FPGA in the Software Radio" // IEEE Communications Magazine, February, 1999 - pp.108-112..
2. Raider,С.М. and Brenner,N.M. "A New Principle for Fast Fourier Transformation" // IEEE Trans., 1976, ASSP-24, - pp.264-265.
3. Preuss,R.D. "Very Fast Computation of the Radix-2 Discrete Fourier Transform" // IEEE Trans., 1982, ASSP-30, - pp.595-607.
4. Duhamel,P., Hollmann,H. "Split Radix FFT Algorithm" // Electronics Letters, Vol.20 No.1, 5th January, 1984 - pp.14-16
5. Jesus Garcia, Juan A.Michell, Angel M.Buron. "VLSI Configurable Delay Commutator for a Pipeline Split Radix FFT Architecture" // IEEE Transactions on Signal Processing, Vol.47, No.11, November, 1999 - pp.3098-3107.
6. Petrovsky,A.A., Kachinsky,M.V. "Design Method of Real-Time Parallel-Pipeline Processors Computing the Vector DFT" // International conference on Parallel Computing in Electrical Engineering, September 2-5, 1998, Bialystok, Poland - pp.242-247.
7. Petrovsky,A., Kachinsky,M. "Automated parallel-pipeline structure of FFT hardware design for real-time multidimensional signal processing" // Proc. of the 9<sup>th</sup> European Signal processing conference, EUSIPCO'98, vol.II, Sep.8-11 1998, Greece - pp.491-494.



PARALLEL-PIPELINE STRUCTURES IMPLEMENTING SPLIT RADIX 2-4 FFT ALGORITHM

Petrovsky A., Shkredov S.

Belarusian State University of Informatics and Radioelectronics,  
6, P.Brovky St., Minsk, 220027, Republic of Belarus  
Computer Engineering Department, e-mail: palex@it.org.by

**Annotation.** The article is devoted to creating a complete methodology for automatic synthesis of real-time FFT-processors at structural level under the given restrictions: speed of input data receipt, structure of the computing element, and the time of the butterfly operation execution. Special attention is paid to creating parallel-pipeline structures for modified split radix 2-4 FFT algorithm. The structures employed in the design show good possibilities for scaling the degree of parallelization, thus changing the overall throughput of the system. They are particularly suited for implementing in programmable logic basis (FPGA).

There have been a lot of interesting developments in the field of digital signal processing lately [1]. Such multimedia applications as speech encoding, voice recognition systems, hands-free telephones and other tend to employ real-time data processing approach [1]. Main part of various signal processing algorithms employ DFT calculation. Moreover, the quality of the signal processing greatly depends on the throughput of the system, which makes calculation of DFT of vector signal a critical point in many real-time solutions.

**1. Split radix FFT algorithm**

There are algorithms of calculating DFT providing better efficiency in comparison with fixed radix FFTs [2],[3]. Thus, algorithm suggested by Duhamel and Hallmann [4] uses the following decomposition of the basic formula:

$$X_k = \sum_{n=0}^{N-1} x_n W^{nk} = \begin{cases} X_{2k} = \sum_{n=0}^{N/2-1} (x_n + x_{n+N/2}) W^{2nk} ; \\ X_{4k+1} = \sum_{n=0}^{N/4-1} [(x_n - x_{n+N/2}) + j(x_{n+N/4} - x_{n+3N/4})] W^n W^{4nk} ; \\ X_{4k+3} = \sum_{n=0}^{N/4-1} [(x_n - x_{n+N/2}) - j(x_{n+N/4} - x_{n+3N/4})] W^{3n} W^{4nk} . \end{cases} \quad (1)$$

This scheme of calculations provides substantial reduction of computational complexity [3].

**2. Split-radix 2-4 parallel-pipeline FFT structure**

General SR FFT pipeline structure based on the modified butterfly operation consists of Computing Elements (CEs) cascaded with Interconnection Elements (IEs), which are routing the data between the CEs [5]. The structure of Computing Element presumes that the butterfly multiplications are done in parallel with the add-subtract operations of the following stage of computation. This split radix FFT pipeline offers decent possibilities for multi-dimensional data flow processing.

Let us suppose that number of SR FFT pipelines  $m$  is a power of 2. In this case, until certain point, the modified SR FFT algorithm processes  $m$  data flows independently. Starting from stage  $\log_2 N/m-1$  and up to the stage  $\log_2 N-2$  (the stage preceding the last one), outputs of every previous stage should be permuted. In other words,  $\log_2 m$  stages (from stage  $(\log_2 N/m)-1$  till the stage with number  $(\log_2 N)-2$ ) start their work from making a certain switching procedure implementing all permutations, required by the graph of split radix FFT algorithm. The switching procedure is always made between pairs of pipelines and its pattern is always the same. The corresponding mapping is demonstrated on Figure 1. Any pair of the pipelines for making the switching procedure at any stage of the split radix FFT algorithm can be determined by the following formula:

$$j=i+N/2^{s+1} , \quad \text{where } i=[0, 1, 2, \dots (N/2^{s+1}-1)] \text{ и } \log_2 N/m-1 \leq s < (\log_2 N)-1 \quad (2)$$

where  $s$  - number of a stage and  $i, j$  are numbers of pipelines. Equation 2 enables creating parallel-pipeline structures implementing split-radix FFT algorithm.

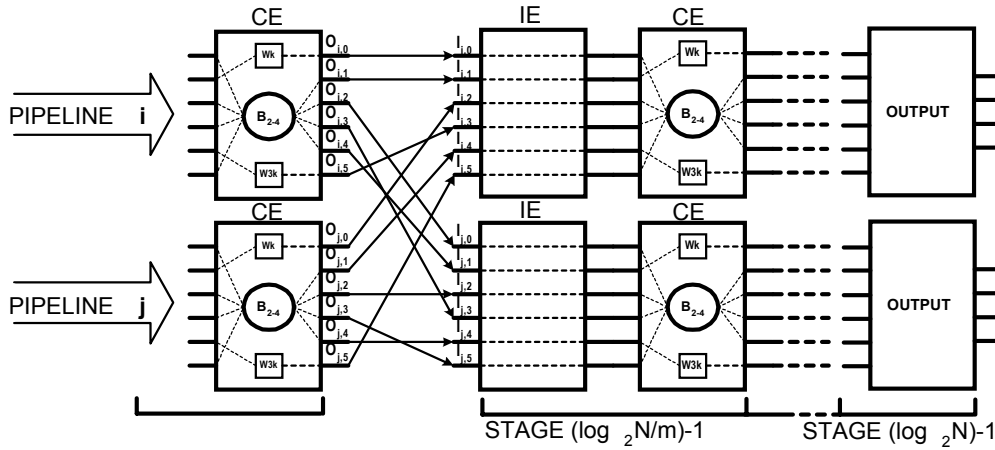


Figure 1. Split radix FFT pipeline for transform of size  $N$  with  $m$  Computing Elements at a stage

### 3. Structural synthesis of parallel-pipeline structure for split-radix FFT

Development of the FFT-processor on the basis of parallel-pipeline structure requires defining the optimum number of computing elements at a stage. The problem is normally reduced to finding the minimal number of CEs at a stage, that at the given time of butterfly operation execution can provide calculation of vector DFT in real time. Condition for calculation the vector DFT in real time for any FFT structure will be  $(DN/k)T_B \leq N\Delta t$ , where  $T_B$  is a time of butterfly operation execution of algorithm FFT;  $\Delta t$  is a sampling period;  $k$  is number of data samples processed by each pipeline;  $D$  is number of components of the vector process;  $N$  is size of a component. Then the minimum number of CEs at a stage  $m_{\min}$  is defined as follows:

$$m_{\min} = \begin{cases} 1, & r = 0, \text{ if } DT_B < 2\Delta t, \\ 2^r, & r = \lceil \log(DT_B/2\Delta t) \rceil, \text{ if } DT_B \geq 2\Delta t, \end{cases} \quad (3)$$

where  $\lceil a \rceil$  is nearest integer, greater than  $a$  or equal to it.

Having found the number of pipelines for the corresponding parallel-pipeline structure, the second problem of structural synthesis is easily solved according to the methodology suggested in previous sections.

### Conclusion

Considering the unified design algorithm and the general description of hardware structures received, mapping into FPGA basis brings ability of integrating the corresponding processor structures into a wide range of devices requiring real-time calculation of multidimensional DFT.

### References

1. Mark Cummings, Shinichiro Haruyama. "FPGA in the Software Radio" // IEEE Communications Magazine, February, 1999 - pp.108-112..
2. Raider,C.M. and Brenner,N.M. "A New Principle for Fast Fourier Transformation" // IEEE Trans., 1976, ASSP-24, - pp.264-265.
3. Preuss,R.D. "Very Fast Computation of the Radix-2 Discrete Fourier Transform" // IEEE Trans., 1982, ASSP-30, - pp.595-607.
4. Duhamel,P., Hollmann,H. "Split Radix FFT Algorithm" // Electronics Letters, Vol.20 No.1, 5th January, 1984 - pp.14-16
5. Jesus Garcia, Juan A.Michell, Angel M.Buron. "VLSI Configurable Delay Commutator for a Pipeline Split Radix FFT Architecture" // IEEE Transactions on Signal Processing, Vol.47, No.11, November, 1999 - pp.3098-3107.