Heuristic-Based Algorithms for the Ordering of Gray Encoded Twiddle Factors of FFT Architectures

[1]Angelo G. da Luz, [2]Eduardo A.C da Costa, [3]Sidinei Ghissoni
angelogl@gmail.com, ecosta@ucpel.tche.br, sidinei.ghissoni@unipampa.edu.br

[1]National Service of Commercial Knowledge – SENAC – Pelotas, Brazil
[2]Catholic University of Pelotas – UCPel – Pelotas, Brazil
[3]Federal University of Pampa– UNIPAMPA – Alegrete, Brazil

Abstract

This paper addresses the exploration of different heuristic-based algorithms for a better manipulation of coefficients in twiddle factors of Fast Fourier Transform (FFT). Due to the characteristics of the FFT algorithms, which involve multiplications of input data with appropriate coefficients, the best ordering of these operations can contribute for the reduction of the switching activity, what leads to the minimization of power consumption in the FFTs. The heuristic-based algorithm named Bellmore and Nemhauser and a new proposed one named Anedma are used to get as near as possible to the optimal solution for the ordering of coefficients in FFTs with larger number of points. Since some low-power techniques for global communication in CMOS VLSI using data encoding methods are used for the decrease of power consumed for transmitting information over buses by reducing the switching activity, we have used Gray encoding technique in the coefficients. As will be shown, the appropriate ordering of coefficients, based on the guidance given by the Anedma heuristic algorithm, can contribute for the reduction of switching activity of the encoded twiddle factors.

1. Introduction

Fast Fourier Transform (FFT) is the largely implementation of the Discrete Fourier Transform (DFT), because this algorithm needs less computation due its recursive operator named butterfly. This operator performs the calculation of complex terms, which involves multiplication of input data by appropriate coefficients [1].

Coefficient ordering is used in [2] as a technique for low power, where all coefficients are ordered in a Fully-Sequential circuit so as to minimize the transitions in the multiplier input and data bus. Thus, the problem is related to finding a better ordering for each coefficient by calculating the minimum Hamming distance between the coefficients. If the number of coefficients is reduced, the cost function can be calculated for all the combinations over the coefficients, since the total number of permutations is still reasonable. However, for a higher number of coefficients this exhaustive algorithm is less attractive due to the time necessary to process the large number of combinations. In this case, heuristic algorithms should be used to get as near as possible to the optimal solution. In this paper, we explore the use of different heuristic-based algorithms in order to search the best ordering of the coefficients for the power reduction of sequential FFT architectures. Two heuristic algorithms are used for this purpose, where one of them is named Bellmore and Nemhauser [3] and the other one is a new proposed heuristic named Anedma [4], which were implemented in order to get as near as possible to the optimal solution for the ordering of larger FFT instances. The main results show that based on the guidance given by the Anedma heuristic algorithm, the switching activity of the twiddle factors can be reduced significantly.

Since the data encoding method has been used for the decrease of power consumed over buses, by reducing the switching activity, we have used Gray encoding technique in order to verify the impact on reducing the switching activity of the coefficients. The results show that depending on the heuristic-based algorithm used, the number of transitions in the encoded twiddle factors can be reduced considerably.

2. FFT Structure

The most popular among the algorithms for the FFT calculation is named common factor FFT [5]. In this work, the main focus is the radix-2 common factor algorithm with decimation in frequency. In this algorithm, the frequency samples are decimated during each stage of the FFT. The operations are realized at each pair of input signals. The basic structure of this algorithm is shown in Fig. 1 for a 16-point example, where the computational structure of the FFT is divided into stages, groups and butterflies. The amount of stages of a radix-2 common factor FFT is given by log2N, where N represents the number of points of the FFT.

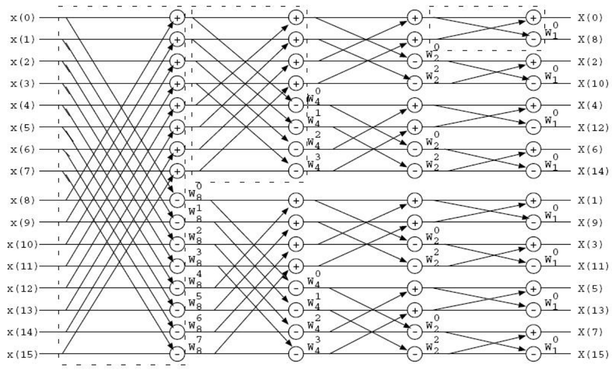

Fig. 1 - Data flow for 16-point radix-2 common factor FFT with decimation in frequency

In the structure presented in Fig. 1, 32 real and 32 imaginary terms are performed in the butterfly (4 stages with 8 butterflies). The butterfly plays a central role in the FFT computation. For the common factor FFT algorithm with decimation in frequency, the butterfly allows the calculation of complex terms according to eq. (1) and (2).

$$C_{complex} = A_{complex} + B_{complex} \tag{1}$$

$$D_{complex} = (A_{complex} - B_{complex})W_{complex} \tag{2}$$

As can be observed in the equations above, one complex addition, one complex subtraction and one complex multiplication are involved in the butterfly block. In the complex multiplication, the input samples are multiplied by fixed coefficients named twiddle factors (W) These coefficients represent values multiple of $\frac{2\pi}{N}$ factor, and they are represented by: $W_N = e^{-j\frac{2\pi K}{N}} = \cos\frac{2\pi k}{N} - j\sin\frac{2\pi k}{N}$, where *k* is the level of the butterfly at each stage of the FFT.

2.1 Ordering of the Twiddle Factors

Tab.1 shows an example of the calculation of the first column (stage) of the FFT, based on the structure previously presented in Fig. 1. This example considers a fully sequential structure with both normal (sequential with no ordering) and after using ordering of the coefficients. In this work we use Gray encoding technique for both i) verify the impact on reducing the switching activity in the original FIR filter coefficients, and ii) verify the best ordering of the encoded coefficients when using the heuristic-based algorithms.

Tab. 1 – Ordering of coefficients in the FFT algorithm

Clock cycles	Normal operation	Operation after ordering the coefficients
1	xo + x8 (x0 – x8) x W_8^0	Xo + x8 (x0 – x8) x W_8^0
2	x1 + x9 (x1 – x9) x W_8^1	x6 + x14 (x6 – x14) x W_8^6
3	x2 + x10 (x2 – x10) x W_8^2	x2 + x10 (x2 – x10) x W_8^2
4	x3 + x11 (x3 – x11) x W_8^3	x5 + x13 (x5 – x13) x W_8^5
5	x4 + x12 (x4 – x12) x W_8^4	x3 + x11 (x3 – x11) x W_8^3
6	x5 + x13 (x5 – x13) x W_8^5	x1 + x9 (x1 – x9) x W_8^1
7	x6 + x14 (x6 – x14) x W_8^6	x4 + x12 (x4 – x12) x W_8^4
8	x7 + x15 (x7 – x15) x W_8^7	x7 + x15 (x7 – x15) x W_8^7

3. Heuristic-based algorithms

This section summarizes the main aspects of the heuristic-based algorithms used in this work.

3.1. Bellmore and Nemhauser Algorithm

Tab.2 presents a random example with the symmetric cost to add each element to the sequence of decisions. Considering the same element values from Tab.2, we would have the following solution ([E2] [E6] [E1] [E3] [E4] [E5]), which would be formed by the steps shown in Fig. 2 (starting from the initial element E1).

Tab. 2 – Costs for the insertion of elements in the solution

Element	E1	E2	E3	E4	E5	E6
E1	0	2	1	4	9	1
E2	2	0	5	9	7	2
E3	1	5	0	3	8	6
E4	4	9	3	0	2	5
E5	9	7	8	2	0	2
E6	1	2	6	5	2	0

Step 1: Insert E3, since it is the best neighbor of E1. Current list: ([E1] [E3]).

Step 2: Insert E6 because it is among the elements not yet added. Choosing the best neighbors to each end of the list, E6 appears as the best neighbor of E1, with a cost of 1. At the other end of the list, E4 is the best neighbor of E3 with a cost of 3. Then, E6 is added to the extreme left of the route. Current list: ([E6] [E1] [E3]).

Step 3: Insert E2 at the left end of the list. From the elements which are not added yet, the best neighbor of E6 is E2 with a cost of 2, and the element with lower cost from E3 is E4 with cost 3. Current list: ([E2] [E6] [E1] [E3]).

Step 4: Insert E4 in the far right of the list, because the best neighbor of E2 is E5 at a cost of 7, and the element with lower cost compared to E3 is E4 with a cost of 3. Current list: ([E2] [E6] [E1] [E3] [E4]).

Step 5: As the only remaining element is the E5, we just must see which end their inclusion has lower cost. If E5 is inserted at the end of E2, then the cost is equal to 7. On the other hand, if inserted as a neighbor of E4 then the cost is 2. Final list: ([E2] [E6] [E1] [E3] [E4] [E5]).

The total cost of the elements of the list is given by:

TotalCost = cost(E2,E6) + cost(E6, E1) + cost(E1, E3) + cost(E3, E4) + cost(E4, E5) + cost(E5, E2) = 16.

Fig. 2 – Bellmore and Nemhauser steps

3.2. Anedma Algorithm

By using this heuristic, the purpose is that a coefficient which has P candidates with the same Hamming distance to be chosen as a neighbor, make the choice of the lowest important coefficient for the others neighbors, thus making it possible to achieve better results. In this heuristic, P threads are started and each one produces an ordering resulting from the application of the algorithm using a random point as initial element. In Tab.3, a hypothetic example with 10 elements randomly generated representing the coefficients and the Hamming distance between them. The total Hamming distance with lexicographical ordering (E1,.. ,E10) is equal to 27. The list of candidates of each element, considering E1 as the initial element is shown in Fig. 3.

Tab. 3 – A hypothetic example with 10 coefficients

Elements	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
E1	0	2	1	4	9	1	1	3	8	1
E2	2	0	5	9	7	2	3	2	2	5
E3	1	5	0	3	8	6	8	9	2	3
E4	4	9	3	0	2	5	1	6	7	2
E5	9	7	8	2	0	2	5	3	2	8
E6	1	2	6	5	2	0	3	2	4	3
E7	1	3	8	1	5	3	0	7	6	2
E8	3	2	9	6	3	2	7	0	1	6
E9	8	2	2	7	2	4	6	1	0	2
E10	1	5	3	1	8	3	2	6	2	0

E1C = ([E3], [E6], [E7], [E10]);
E2C = ([E6], [E8], [E9]);
E3C = ([E9]);
E4C = ([E7]);
E5C = ([E4], [E6], [E9]);
E6C = ([E2], [E5], [E8]);
E7C = ([E4]);
E8C = ([E9]);
E9C = ([E8]);
E10C = ([E4]).

(a)

E1C = ([E3], [E6], [E10]);
E2C = ([E6]);
E3C = ([E9]);
E4C = ([E7]);
E5C = ([E6]);
E6C = ([E2], [E5]);
E7C = (); E8C = ();
E9C = ([E8]); E10C = ([E4]);
Finally, the list EBloq is given by::
Ebloq=([E1],[E9],[E7],[E4],[E8])

(b)

E1C = ([E3], [E10]);
E2C = ([E6]);
E3C = ([E9]);
E4C = ([E7]);
E5C = ();
E6C = ([E2], [E5]);
E7C = (); E8C = ();
E9C = ([E8]); E10C = (E4).
Ebloq = ([E1], [E9], [E7], [E4], [E8], [E6])

(c)

Fig. 3. Steps of the Anedma algorithm

In addition, E1 is already entered in a locked elements list named EBloq that now is EBloq = ([E1]). This list prevents that these elements will be selected as candidates from other factors. E1C represents the list of elements candidate of (coefficient) E1, E2C represents the candidate list of element E2, and successively. Fig. 3(a) shows this procedure. The partial Ebloq list is shown in Fig. 3(b). The final Ebloq list, with the total Hamming of this ordering reduced to 20 is shown in Fig. 3(c).

4. Results

In this section, we present the results obtained with the two heuristic-based algorithms and with the encoding technique described in the previous sections. We used 16 bit-width FFT architectures with 128, 256, 512 and 1024 points. As the twiddle factors are composed by complex terms, the coefficients were divided into real and imaginary parts.

Tab.4 shows the obtained results, in terms of number of transitions, after applying Gray encoding in the original and ordered coefficients (after applying Bellmore and Nemhauser and Anedma algorithms). As should be seen, both algorithms achieved significant reduction in terms of Hamming distance, mainly on the set of real coefficients with the Anedma algorithm.. It occurs because this algorithm uses a list of candidates that provides (when more than one coefficient with the same Hamming of the best candidate is found), the possibility of not making a random choice between them, but keep them in a candidate list. In case of any of these elements is the unique best candidate to one of the following coefficients analyzed, it can be removed from this list to be added as best candidate of another element. Thus, each coefficient with more than one candidate always will select the candidate that is the one which has the less importance to others coefficients, thus providing further optimization of the obtained results.

Tab. 4 – Applying the ordering on the Gray encoded coefficients

Twiddles factors size	Number of transitions between the coefficients (Hamming distance)									
	Original Coefficients		Ordered Real Coefficients				Ordered Imaginary Coefficients			
	Real	Imaginary	Bellmore and Nemhauser		Anedma		Bellmore and Nemhauser		Anedma	
	Ham.	Ham.	Ham	Red. (%)	Ham	Red. (%)	Ham	Red. (%)	Ham	Red. (%)
128	508	232	206	59,4	200	60,6	142	38,7	138	40,5
256	926	436	376	59,4	372	59,8	258	40,8	250	42,7
512	1702	816	700	58,9	694	59,2	446	45,3	440	46,1
1024	3138	1542	1248	60,2	1246	60,3	748	51,4	740	52,0
AVG				59,5		60,0		44,1		45,3

5. Conclusions

In this work two heuristic-based algorithms named Anedma and Bellmore and Nemhauser were applied to the ordering of Gray encoded FFT twiddle factors. The results shows that the algorithms can find a good cost function for the ordering of the coefficients and the Hamming distance between consecutive coefficients could be reduced significantly. As future work we intend to implement the FFT architectures with original and encoded ordered coefficients in order to observe the impact on power reduction of the FFT with the reduction of the number of transitions in the coefficients.

6. References

[1] F. Qureshi, and O Gustafsson.” Low-complexity reconfigurable complex constant multiplication for FFTs ” Circuits and Systems. ISCAS 2009. IEEE International Symposium on, pp. 1137-1140, May 2009.

[2] M. Mehendale, S. Sherlekar, and G. Venkatesh.. “Algorithmic and Architectural Transformations for Low Power Realization of FIR Filters”. 11th Int. Conf. on VLSI Design, pp. 12-17, 1998.

[3] Reeves, C.R. Modern Heuristic Techniques for Combinatorial Problems. Blackwell Scientif Publications, 1993.

[4] A. Luz, E. Costa, M. Aguiar. “Ordering and Partitioning of Coefficients Based on Heuristic Algorithms for Low Power FIR Filter Realization”. 23rd Symposium on Integrated Circuits and Systems Design – SBCCI, pp. 180-185, 2010.

[5] J. W. Cooley and J. W. Tukey, “An Algorithm for the Machine Calculation of Complex Fourier Series. Mathematics of Computation”, [S.l.], v.19, n.90, p.297–301, 1965.