IEICE Electronics Express, Vol.8, No.17, 1437–1443

On the efficient computation of single-bit input word length pipelined FFTs

Saima Athar[1a)], **Oscar Gustafsson**[1b)], **Fahad Qureshi**[1c)], **and Izzet Kale**[2d)]

[1] *Department of Electrical Engineering*
Linköping University
SE-581 83 Linköping, Sweden

[2] *Applied DSP and VLSI Research Group*
University of Westminster, 115 New Cavendish Street
London W1W 6UW, United Kingdom

a) *saima@isy.liu.se*
b) *oscarg@isy.liu.se*
c) *fahadq@isy.liu.se*
d) *kalei@westminster.ac.uk*

Abstract: This letter describes an efficient architecture for the computation of fast Fourier transform (FFT) algorithms with single-bit input. The proposed architecture is aimed for the first stages of pipelined FFT architectures, processing one sample per clock cycle, hence making it suiable for real-time FFT computation. Since natural input order pipeline FFTs use large memories in the early stages, it is important to keep the word length shorter in the beginning of the pipeline. By replacing the initial butterflies and rotators of an architecture with that of the proposed block, the memory requirements can be significantly reduced. Comparisons with the commonly used single delay feedback (SDF) architecture show that more than 50% of the required memory can be saved in some cases.

Keywords: FFT algorithm, complexity, single-bit, word length
Classification: Integrated circuits

References

[1] J. Bao-Yen Tsui, *Fundamentals of Global Positioning System Receivers: A Software Approach*, John Wiley and Sons, 2000.
[2] E. D. Kaplan and C. J. Hegarty, *Understanding GPS Principles and Applications*, Artech House, 2006.
[3] A. Molino, G. Girau, M. Nicola, M. Fantino, and M. Pini, "Evaluation of FFT-based acquisition in real time hardware and software GNSS receivers," *Proc. IEEE ISSSTA*, pp. 32–36, Aug. 2008.
[4] L. Wanhammar, *DSP Integrated Circuits*, Academic Press, 1999.
[5] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, Tata McGraw-Hill, 2006.
[6] S. He and M. Torkelson, "Design and Implementation of a 1024-point

DOI: 10.1587/elex.8.1437
Received August 01, 2011
Accepted August 12, 2011
Published September 10, 2011

Pipeline FFT Processor," *Proc. IEEE CICC*, pp. 131–134, May 1998.
[7] O. Gustafsson and F. Qureshi, "Addition aware quantization for low complexity and high precision constant multiplication," *IEEE Signal Processing Lett.*, vol. 17, no. 2, pp. 173–176, Feb. 2010.

1 Introduction

In digital radio communications, spread spectrum (SS) techniques are gaining more importance. In such systems, each user is identified by a unique spreading sequence. Global navigation satellite systems (GNSS) such as GPS and Galileo use direct sequence spread spectrum (DSSS) modulation techniques. In digital GNSS signal processing, the first step is signal acquisition. The purpose of acquisition is to determine from which satellite the received signal originates. The GPS data sequence is combined with a pseudo random noise (PRN) code and then modulated by the carrier wave, forming the DSSS signal to be transmitted. At the receiving end, the incoming signal is correlated by locally generated codes [1, 2] to determine the satellite. Due to the cyclic nature of PRN codes, this aquisition can be efficiently implemented using discrete Fourier transforms (DFT) [3]. Typically, these DFTs are computed using the class of algorithms known as fast Fourier transforms (FFTs) [4, 5] as the lengths are powers of 2. One commonly used architecture class for real-time FFT computation is the pipelined FFT. These are highly regular and characterized by continuous processing of the input data [6].

In this work, we propose an efficient architecture for the initial stages of the radix$-r$ single delay feedback (RrSDF) pipelined FFT architecture when the input wordlength is short, such as for the local code in a GNSS acquisition system. The initial butterflies and rotators of RrSDF are replaced by a Look-Up Table (LUT) and, as opposed to the traditional SDF architecture, only the input data is stored resulting in reduced storage requirements. We present two different approaches for mapping the initial stages of the RrSDF using decimation in time (DIT) radix$-r$ stage. This approach would find use in practical applications for low complexity and low power portable as well as other applications. Furthermore, we discuss the radix trade-off for the first stage and how it can be applied for general short input word length DFT computation.

2 Proposed architecture

An N-point DFT can be expressed as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, k = 0, 1, 2, \ldots, N-1 \quad (1)$$

where $W_N^{nk} = e^{-j2\pi nk/N}$ is the twiddle factor, n is the time index and k is the frequency index [5]. Figure 1 shows the mapping of the first two stages of a R2SDF pipelined FFT architecture onto the proposed architecture. In

DOI: 10.1587/elex.8.1437
Received August 01, 2011
Accepted August 12, 2011
Published September 10, 2011

FFT architectures, a multiplier is commonly denoted by the twiddle factor resolution, such that a resolution of N points around the unit circle is denoted a W_N multiplier. A W_4 multiplier only performs multiplications with the trivial coefficients $\{\pm 1, \pm j\}$. For a W_8 multiplier, multiplication either by 1 or $\sin \pi/4 = \cos \pi/4$ is required plus possible negation, to obtain the result for all eight possible twiddle factors.

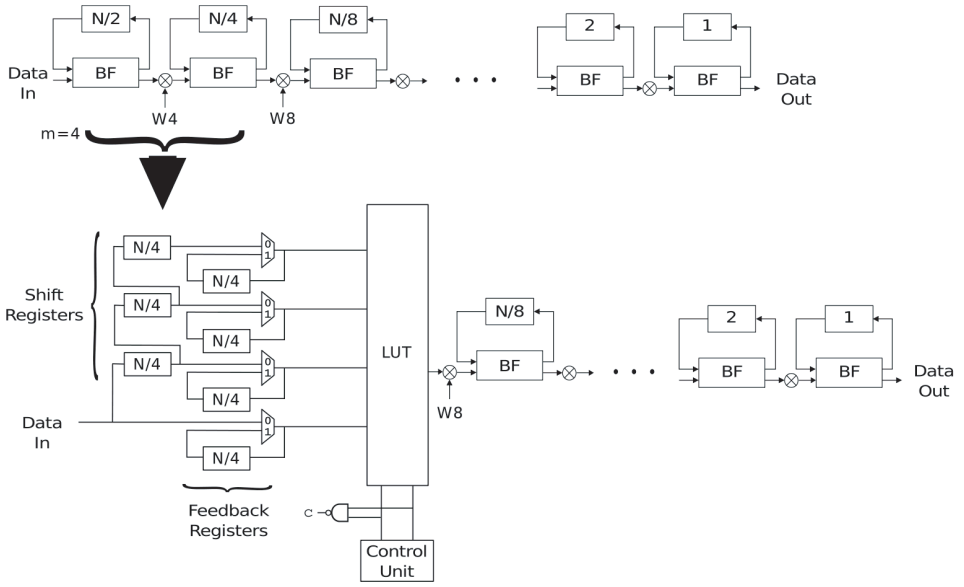

Fig. 1. Mapping of the first two stages of a R2SDF architecture onto the proposed architecture.

The proposed architecture consists of $(m-1)$ shift registers and m feedback registers of size N/m each (where N is the FFT length), m multiplexers controlled by a signal c, a Look-Up Table (LUT), and a control unit. The number of stages of R2SDF that are mapped onto the proposed architecture are related to the size of the LUT by the relation $m = r^s$. Here s corresponds to the number of mapped radix$-r$ stages and m corresponds to radix of the resulting building block. The number of inputs to the LUT is $m + sr/2$. The control unit can be the same binary counter typically used to control the rest of the pipelined FFT architecture, using the $s \log_2 r$ most significant bits.

Considering Fig. 1 ($m = 4$, $s = 2$, $r = 2$), we can describe the behavior of the proposed architecture as follows. During the first $3N/4$ cycles, $c = 1$ and data from the input sequence is directed towards shift registers until they are filled. For the next $N/4$ cycles, $c = 0$, and data from the shift registers will appear at the input of the LUT, and the first output of a 4-point DFT is determined with the incoming data and the data from the shift registers. During these $N/4$ cycles, data is also fed to the feedback registers. On the next $3N/4$ cycles when $c = 1$, again 4-point DFTs are calculated with the data from the feedback registers and the next input frame is directed towards shift registers and so on. The two signals from the control unit, obtained from a binary counter increasing by one every $N/4$ cycle, determines which of the

DOI: 10.1587/elex.8.1437
Received August 01, 2011
Accepted August 12, 2011
Published September 10, 2011

four precomputed DFT outputs the LUT should put at the output.

The Look-Up Table (LUT) in Fig. 1 is used to store pre-computed results from the 4-point DFT computation. The transfer function from inputs to the outputs can be written as 8×8 real-valued matrix-vector multiplication. For each input combination value, the resulting output values for a 4-point DFT are stored in the LUT and the control signal selects the correct value of a 4-point DFT stored in the LUT. Since, the input word length is short it is reasonable to believe that this approach is efficient compared to using discrete arithmetic operators. The approach is related to distributed arithmetic, although while distributed arithmetic often operates using bit-serial data, here we only have single-bit data.

For a general value of $m = r^s$, the shift registers are filled during the first $(m-1)N/m$ cycles and in the next N/m cycles m-point DFTs are calculated with the incoming data and the data from the shift registers. Data is also fed to the feedback registers during this time. During the next $(m-1)N/m$ cycles, again m-point DFTs are calculated with the data from the feedback registers and the next input frame is directed towards the shift registers and so on. The control unit now consists of $s \log_2 r$ signals, forming a binary counter increasing every N/m cycles. The LUT now needs to store $2^{m+s \log_2 r} = m2^m$ different words.

Two different approaches for mapping the initial stages of the RrSDF are suggested. In the first approach, the proposed architecture will replace the initial stages of a R2SDF pipelined FFT architecture. For $m = 2$, only first stage of R2SDF will be mapped onto the proposed architecture. Similarly, first two stages ($m = 4$) and first 3 stages ($m = 8$) of R2SDF will be mapped and so on. While in the second approach, the first stage of R4SDF and R8SDF will be mapped onto the proposed architecture.

Consider an m-point DFT. The transfer function from the inputs to the outputs can be written as an $m \times m$ complex matrix-vector multiplication or as a $2m \times 2m$ real-valued matrix-vector multiplication (as discussed above). During each clock cycle, one real and one imaginary data are computed. Considering the R2SDF architecture, the required output word length, W_{out}, can be calculated for each stage. The number of output bits for $m = 2$ and 4 are $(W_{in} + 1)$ (only real output) and $2(W_{in} + 2)$ (both real and imaginary output), respectively. For $m \geq 8$, the number of output bits depends on the required accuracy. For a single-bit real-valued input, the number of output bits for $m = 2$ is two bits for the real output (the imaginary output is always zero). For $m = 4$ we have three real bits and two imaginary bits (one sign bit and one data bit). For both the cases we use the negated value of the output since this maps better to the two's complement representation (maximum positive and negative output values are 4 and -2 for $m = 4$, respectively). This can easily be compensated for at later stages if required.

Similarly, in the second approach, for R4SDF the output word length, W_{out}, for a single-bit real-valued input is $(W_{in} + 4)$ for $s = 1$. For R8SDF, the number of output bits depends on the required accuracy (but not for all outputs).

DOI: 10.1587/elex.8.1437
Received August 01, 2011
Accepted August 12, 2011
Published September 10, 2011

The proposed block as shown in Fig. 1 will efficiently use the register memory by utilizing the short word length at the input.

For single-bit real input data, the register memory required for intermediate storage of the input samples can be calculated as:

$$(m-1)\frac{NW_{in}}{m} + m\frac{NW_{in}}{m} = 2NW_{in} - \frac{NW_{in}}{m} \tag{2}$$

The mapping of the first two stages of the R2SDF (i.e., $m = 4$) is logically equivalent to the mapping of first stage of the R4SDF onto the proposed architecture as both corresponds to a DIT (decimation in time) radix−4 stage.

3 Results

A comparison of the number of registers required for the traditional RrSDF and the proposed architecture is shown in Table I.

Table I. Comparison of register memory bits for $W_{in} = 1$.

	$m = 2$	$m = 4$	$m = 8$
Proposed	$1.5N$	$1.75N$	$1.875N$
R2SDF	N	$2.50N$	$N(\frac{W_{out}}{4} + 2.5)$
R4SDF	–	$3.75N$	–
R8SDF	–	–	$N(\frac{W_{out}}{2} + 3.375)$
Ratio[1]1	1.5	0.70	$\frac{7.5}{W_{out}+10}$
Ratio[2]2	–	0.46	$\frac{3.75}{W_{out}+4.75}$

[1] Ratio of proposed and R2SDF

[2] Ratio of proposed and RrSDF

The word length of the output can be estimated with the objective to get reduced memory size which leads to reduced chip area and power consumption. From Table I it is clear that the memory requirements for the first stage of R4SDF and R8SDF is greater than for R2SDF. This is caused by the larger increase in output word length for higher radix architectures. Furthermore, if we replace the first stage of the pipelined R2SDF with the proposed block ($m = 2$), there is an increase in memory requirement. If the first two stages of the pipelined R2SDF are replaced by the proposed block, i.e., $m = 4$, then $1.75N$ bits compared to $2.5N$ bits shows a significant gain, which would correspond to the FFT algorithm using a radix−4 stage. For higher order, ($m \geq 8$) the memory size reduction for the proposed block compared to the first three stages of the pipelined R2SDF architecture depends upon the chosen output word length. For an output word length of 5 bits, the reduction is 50% while if we increase the word length to 15 bits then, 70% reduction in memory size is obtained. This means that the relative saving increases by a longer output word length. While memory savings are expected for $m = 16$, the LUT will grow large, requiring 2^{20} words. For the previous reasons, only replacing two or three R2SDF stages with the proposed architecture is considered for hardware implementation.

DOI: 10.1587/elex.8.1437
Received August 01, 2011
Accepted August 12, 2011
Published September 10, 2011

Both the proposed and the R2SDF architectures have been described using VHDL and synthesized to an FPGA, in this particular case a Xilinx 4V-FX-12-SF-363. For $m = 4$, the corresponding FFT used a radix-2^2 stage and for $m = 8$ a radix-2^3 stage was used. In this way, the multipliers in the R2SDF architecture were minimized. Also, the functionality of the two blocks is exactly identical. The W_8-multiplier for the R2SDF architecture is implemented using a complexity optimized constant multiplier based on [7]. The resource utilization results for an FFT-size of $N = 1024$ is shown in Table II. Since the memory requirements for replacing three or more stages (i.e., $m \geq 8$) of R2SDF with the proposed architecture depends upon the chosen output word length, different W_{out} are considered. The results show that the logic (function generators) using the proposed architecture is sometimes slightly increased. However, the number of registers are, as expected, significantly decreased. Furthermore, the number of FPGA resources (CLB slices $\approx$ two function generators and registers) are significantly decreased. The difference between the number of registers reported in Table II and those that can be calculated from Table I, are pipeline registers used at the output of the look-up table and some minor logic functions.

Table II. Comparison of replacing the initial stages of R2SDF with the proposed architecture (with different W_{out}) for $N = 1024$.

Architecture	Hardware Resources			W_{out}
	CLB Slices	Registers	FG[3]	
First 2 stages of R2SDF are replaced by proposed architecture				
Proposed	901	1802	27	–
R2SDF	1282	2564	18	–
First 3 stages of R2SDF are replaced by proposed architecture				
Proposed	960	1920	226	8
	960	1920	228	10
	960	1920	232	12
R2SDF	2317	4634	176	8
	2575	5150	185	10
	2832	5664	272	12

[3]Function generator.

4 Conclusions

In the R2SDF pipelined architecture, one of the outputs from the butterfly is stored in a feedback register. Conversely, the proposed architecture, based on the R2SDF pipelined architecture, stores the input samples instead of the output samples. Since the pipelined architecture uses large memories in the early stages when natural input order is considered, the proposed architecture leads to significant register savings when operating on short input word length

DOI: 10.1587/elex.8.1437
Received August 01, 2011
Accepted August 12, 2011
Published September 10, 2011

sequences.

Acknowledgments

S. Athar was supported by NED University, Pakistan. F. Qureshi was supported by Higher Education Commission, Pakistan. O. Gustafsson was supported by CENIIT at Linköping University, Sweden. I. Kale was partially supported by the Innovative Navigation using new GNSS Signals with Hybridised Technologies (INSIGHT) funded by the UK Engineering and Physical Sciences Research Council (EPSRC).

DOI: 10.1587/elex.8.1437
Received August 01, 2011
Accepted August 12, 2011
Published September 10, 2011