VHDL Core for 1024-Point Radix-4 FFT Computation

Vite-Frias Jose Alberto1     Romero-Troncoso Rene de Jesus2     Ordaz-Moreno Alejandro3
Facultad de Ingenieria Mecanica Electrica y Electronica, FIMEE, Universidad de Guanajuato
Tampico No. 912, CP. 36730, Salamanca, Gto.
Tel. :(464)6480911 y 6480386     Fax:(4)6472400
1 albertovite@hotmail.com
2 troncoso@salamanca.ugto.mx
3 a_ordaz_m@hotmail.com

Abstract
This paper shows the development of a 1024-point radix-4 FFT VHDL core for applications in hardware signal processing, targeting low-cost FPGA technologies. The developed core is targeted into a Xilinx® Spartan™-3 XC3S200 FPGA with the inclusion of a VGA display interface and an external 16-bit data acquisition system for performance evaluation purposes. Several tests were performed in order to verify FFT core functionality, besides the time performance analysis highlights the core advantages over commercially available DSPs and Pentium-based PCs. The core is compared with similar third party IP cores targeting resourceful FPGA technologies. The novelty of this work is to provide a low-cost, resource efficient core for spectrum analysis applications.

Keywords: FFT, HSP, VHDL, IP cores

1. Introduction

There are several methodologies and technologies that already offers hardware and software solutions for fast Fourier transform (FFT) computation having specific advantages for specific applications. Among the solutions are: PC, DSP, application specific integrated circuits (ASIC) and HDL cores which are intended to be implemented into reconfigurable logic having the advantages of ASIC in parallel architecture and time performance over microprocessor based solutions like DSP.

1024-point FFT computation is considered the main basic algorithm for several DSP applications, [1]. Different FFT algorithms have been proposed to exploit certain signal properties to improve the trade-off between computation time and hardware requirements. Radix-4 based algorithms improve computation time in a factor of two, compared with radix-2 based algorithms, increasing hardware requirements by the same factor.

Considering that low-cost, high-density reconfigurable devices are already available, an optimized price/performance HDL core development of the 1024-point radix-4 FFT is feasible.

2. Radix-4 FFT

The N-point discrete Fourier transform (DFT) is defined as state equations (1 - 2)

\[X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}\]

\[W_N^k = \cos\left(\frac{2\pi nk}{N}\right) - j\sin\left(\frac{2\pi nk}{N}\right)\]

Where \(x(n)\) is the time-domain discrete input signal and \(X(k)\) is the DFT. \(n\) represents the discrete time-domain index, while \(k\) is the normalized frequency-domain index.

From the implementation point of view, DFT computation is highly inefficient; therefore, a divide-and-conquer algorithm has been proposed to improve computation efficiency. This algorithm is known as the fast Fourier transform or FFT, [2 - 3]. Among the approaches followed for the FFT computation, radix-4 time-decimation has been widely used for a number of practical applications and this is the approach developed in the present work. The algorithm consists in the decomposition of the computation in 4x4 multiplicative and additive processing, named 4x4 butterflies or dragonflies that can be expressed as equation (3).

\[
\begin{bmatrix}
A(1) \\
A(1+4^C) \\
A(1+2\cdot4^C) \\
A(1+3\cdot4^C)
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & 1 & W_N^{04}\ A(1) \\
1 & -j & -1 & jW_N^{04}\ A(1+4^C) \\
1 & -1 & -1 & W_N^{02}\ A(1+2\cdot4^C) \\
1 & j & -1 & jW_N^{02}\ A(1+3\cdot4^C)
\end{bmatrix}
\]

For a radix-4 1024-point FFT, it is necessary to compute 5 stages of 256 dragonflies. In order to re-use the dragonfly...
computational block, it must be established the addressing index \((q\) and \(l)) for each block as stated in equations (4 – 5).

\[
q = \left\lfloor \frac{R}{4^{C}} \right\rfloor \cdot 4^{C}
\]

\[
l = \text{Re} \left( \frac{R}{4^{C}} \right) \cdot 4^{C+1} + \left\lfloor \frac{R}{4^{C}} \right\rfloor
\]

Where \(C\) is the column index \((C=0,1,..4)\) and \(R\) is the R-dragonfly \((R=0,1,2,...255)\) at stage \(C\).

3. HSP unit

The main block for the FFT computation is the dragonfly processor which contains complex fixed-point multipliers and adders. In order to minimize resources for the computation units, complex multipliers can be implemented as stated in equation (6) with the structure shown in Fig. 1 for 25% savings in real multipliers.

\[
(a + bj)(c + dj) = ac - bd + j[(a + b)(c + d) - ac - bd]
\]

Equation (3) can be rewritten using equation (7), saving 33% of adders required, that gives the structure shown in Fig. 2.

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 - j & -1 & j \\
1 - l & 1 & -1 \\
1 & j & -1 & -j
end{bmatrix}
\rightarrow
\begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 0 & -j \\
1 & 0 & -1 & 0 \\
0 & 1 & 0 & 1
end{bmatrix}
\begin{bmatrix}
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 1 & 0 & -1
end{bmatrix}
\]

Input data and intermediate results are stored in an internal dual-port double-bank RAM, while the kernel coefficients are stored in an internal ROM. The overall algorithm computation is supervised by a sequencer with a finite-state machine (FSM) as shown in Fig. 3. [4]
4. Implementation

The radix-4 1024-point FFT core was described under VHDL and simulated in Active-HDL™ in order to verify its functionality. Results were compared with MatLab® simulations under the same conditions to validate the algorithm performance. For overall performance test, a case study was proposed to be implemented as a direct application of the developed core.

The case study consists in a basic spectrum analyzer for base-band signals, using a low-cost development platform. Xilinx® Spartan-3™ [5-6] XC3S200-based with 1 Mb external RAM development board was used for case study prototyping. A 16-bit 100 kSPS data acquisition system was developed for input data, based on the Texas Instruments ADS7809 [7] analog to digital converter. VGA display capabilities of the development board were used to display the spectrum estimation results.

The spectrum analyzer makes use of the radix-4 1024-point FFT core, along with a data acquisition system controller, a square root computation unit for magnitude estimation, a VGA display driver and a memory management unit for data addressing. Fig. 4 shows the spectrum analyzer block diagram, showing each component.

Considering a complex number, \( p = p_r + p_j \), its modulus (magnitude) is defined as follows:

\[
|p| = \sqrt{(p_r)^2 + (p_j)^2} \tag{8}
\]

As eq. (8) states, FFT complex sequence modulus computation is based on the square root computation. Taking into account inherited parallel performance is easy to conclude that overall system performance is limited by the slowest process (1024-point FFT), this fact allows to use a simple sequential unit for square root computation based on a successive approximation algorithm (SAR). The structure of the square root computation unit is shown in figure 5.

5. Results

Once implemented the FFT core for spectrum analysis, several tests were performed in order to verify its functionality. Fig. 6 shows the VGA display of the system for a spectrum estimation of a 33.42 kHz sine input waveform which is consistent with the expected theoretical spectrum estimation.

Implementation resources, as reported by Xilinx® ISE 6.2 synthesis tool, are summarized in table 1 for the FFT core only and the spectrum analyzer case study.

Time-computation performance of the FFT core can be estimated by the clock cycle number, required to compute a full 1024-point input signal. There are 5 butterfly stages, \( s \), with 256 individual complex dragonflies, \( d \). Each
dragonfly requires 6 clock cycles. Therefore, the number of clock cycles, $L$, is given by equation (9).

$$L = sdc = 5(256)6 = 7680 \quad (9)$$

For a 50 MHz master clock with a Spartan™-3 FPGA, the 1042-point FFT, the computation time is 153.84 $\mu$s.

![Figure 6. Spectrum for a 33.42kHz sine input waveform.](image)

### Table 1. Used resources report.

<table>
<thead>
<tr>
<th>Device: XC3S200</th>
<th>FFT core</th>
<th>Case study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slices</td>
<td>47%</td>
<td>93%</td>
</tr>
<tr>
<td>Slice Flip Flops</td>
<td>9%</td>
<td>17%</td>
</tr>
<tr>
<td>LUT</td>
<td>43%</td>
<td>84%</td>
</tr>
<tr>
<td>RAM blocks</td>
<td>16%</td>
<td>66%</td>
</tr>
<tr>
<td>18x18 Multipliers</td>
<td>75%</td>
<td>100%</td>
</tr>
<tr>
<td>Average</td>
<td>38%</td>
<td>72%</td>
</tr>
</tbody>
</table>

Note: report based on ISE v6.2 synthesis

### 6. Conclusions

FFT core functionality has been shown through the spectrum analyzer case study by demonstrating the system integration with the core.

Raw computation efficiency can be compared with similar systems, to determine advantages and disadvantages of the core. Commercially available DSPs have been optimized for several common DSP tasks like radix-4 1024-point FFT; and as an example, the TMS320C30 DSP from Texas Instruments [8] report a peak performance of 2.533 ms for FFT computation which is 19.5 times slower than the FFT core implementation under the Xilinx® Spartan™-3 50 MHz FPGA. Pentium-based PCs are even slower than the DSP with a computation performance average of 120ms on an 1.7 GHz Intel® Celeron® processor based PC. The developed FFT core can be compared also with already commercially available FFT cores like the IP core offered by Xilinx® for Virtex™ II FPGA family which reports a peak performance of 1319 clock cycles, being faster than the developed core, but it makes use of higher resources.

As it can be easily seen, the developed core is faster than similar commercially available DSPs and Pentium-based PCs, even considering that the full application makes use of the core computation, besides interfacing and additional computations like complex modulus, making the FFT core much more efficient because of its intrinsic parallel structure. The commercially available core for Virtex™-II FPGA [9] family reports a peak time performance factor of 5.8, compared with the developed core; however, it requires 7 times more RAM blocks and approximately an 165.3 % more slice resources making it unsuited for Spartan™-3 FPGA family.

The developed FFT core has a higher time-performance than similar DSP and PC systems, and although it has lower time-performance compared with the commercially available core from Xilinx®, it has the advantage of using lower resources making it feasible to be implemented in lower-cost FPGA families like the Spartan™-3.

### Acknowledgements

The authors wish to thank Texas Instruments for providing the ADS7809 samples used in the data acquisition system, and the Facultad de Ingenieria at Universidad Autonoma de Queretaro for the manufactured printed circuit board prototype.

### References

[8] Regional Technology Center, An Implementation of FFT, DCT, and Other Transforms on the TMS320C30 datasheet, Texas Instruments, 1997