SASE

FPGAs HDLs y ASICs

Low size FFT core for OFDM communications

Andres D. Cassagnes*, Federico G. Zacchigna*, Octavio Alpago* and Ariel Lutenberg* **

*Laboratorio de Sistemas Embebidos (FIUBA) **CONICET-GICSAFe

Introduction and objectives

One of the most used modulation schemes in communication systems that fulfills the current data bandwith demand is the Orthogonal Frequency Division Multiplexing (OFDM), which transmits the data over multiple orthogonal carriers. The main advantage of the OFDM over other multi-carrier modulations is the carrier orthogonality, which permits to overlap the spectrum of the carriers without getting any inter-carrier interference.
The OFDM modulation scheme can be expressed as:

$$s_k(t-kT) = \sum_{i=-N/2}^{N/2-1} x_{i,k} e^{j2\pi\left(\frac{i}{T}\right)(t-kT)} \qquad (1)$$

where $x_{i,k}$ is the ith data symbol associated to the kth sub-carrier, and T is the symbol period. It is easy to recognize in (1) an Inverse Discrete Fourier Transform. Assuming that $x_{i,k}$ is constant along the symbol period T it is possible to use an Inverse Discrete Fourier Transform/Discrete Fourier Transform (IDFT/DFT) blocks to modulate/demodulate the OFDM signal, which might be computed by using the efficient Fast Fourier Transform (FFT) algorithm, which reduces the complexity of the algorithm from $\mathcal{O}(n^2)$ to $\mathcal{O}(n\,logn)$
The objective of this work is to obtain an FFT core, small enough to be included in a complete OFDM transceiver without consuming too much resources, but efficient and accurate enough to be useful in an ISDB-T television system.
The main requirements for the implementation can be summarized in:

- Run-time configurable FFT length, including at least 2K, 4K and 8K samples.
- $8,126,984$ sampling frequency guaranteed.
- Continuous input and output (not burst mode).
- Fixed point arithmethics.
- Run-time configurable and step-selectible scaling, with rounding and clipping options.
- Lower space consuption than other implementations, taking as references Xilinx IP and an open FFT for ISDB-T OFDM implementation.

Implementation

Radix-r algorithms reduces the calculation of one n-point FFT to the calculation of ν smaller r-point sub-FFTs where $N = r^\nu$. This leads to the posibility of reuse the same modules for all the r-point FFT computation.
As the main objective is to achieve a low size core, an iterative version is chosen for implementation because it uses only one r-point FFT core for all the ν sub-FFTs computations. The limitation of this approach is given by the operation frequency. As the same module is used for every FFT stage, in a ν stage implementation a $(f*\nu)$Hz clock is needed to achieve a fHz sample frequency.
Two architechtures are implemented, a radix-2 and a radix-4, in order to compare them and bring the possibility to choose depending on the requirements of the specific application.
For twiddle factors multiplication two variants are implemented, an iterative cordic and an efficient complex multiplicator.
The escaling unit is implemented right after the butterfly / dragonfly unit in order to provide the optional rounding/clipping function. This is run-time configurable and brings the possibility to choose the stages where the scaling is applied.
Both implementation scheme are showed in Figures 1 and 2.

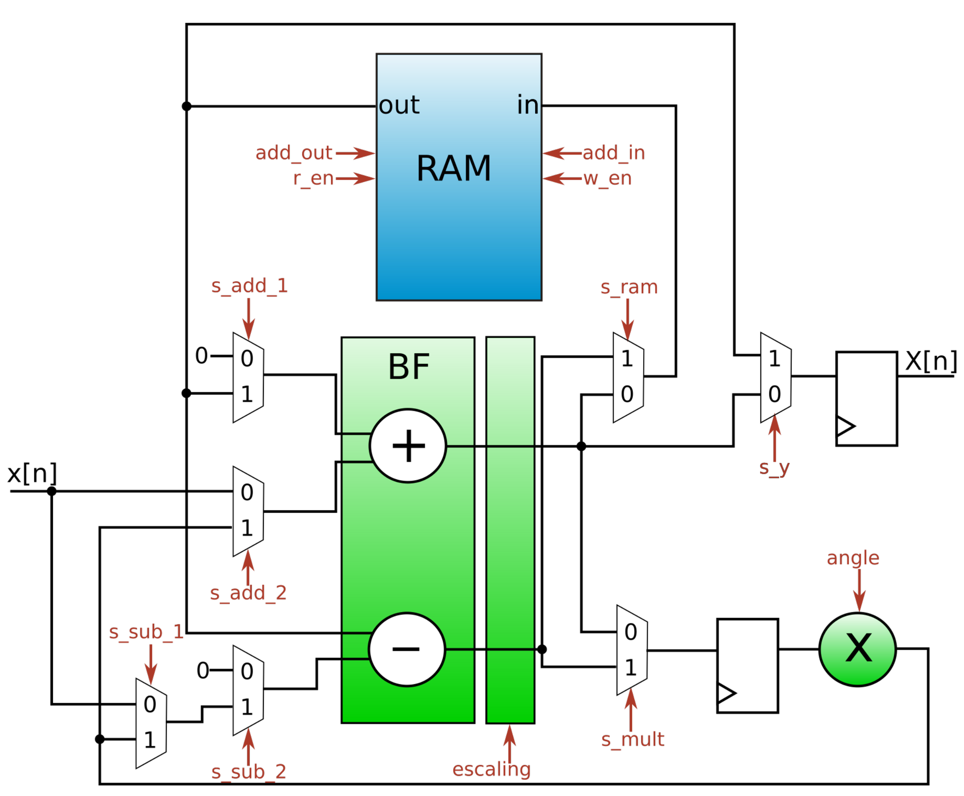

Figure 1: Radix-2 implementation diagram

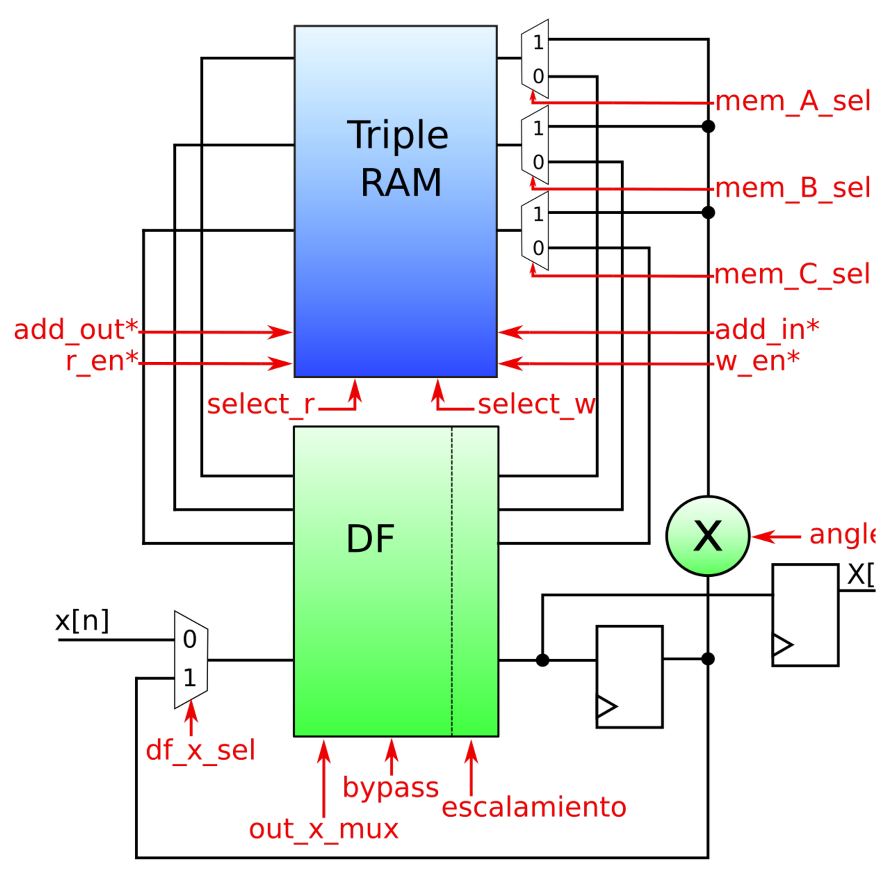

Figure 2: Radix-4 implementation diagram

Characterization

Beside of the individual tests for every composing unit, a set of tests is performed over the entire architectures in order to verify and validate the design. For a complete description of tests and results, refer to the article related to this poster.
In order to measure the architecture error, the 64 bits floating point Matlab FFT is taken as a benchmark.
Two metrics are used for error measuring, maximum relative error, E_∞, and root mean square error, E_2:

$$E_\infty = \max\left(\frac{X_o[n] - X_{dut}[n]}{X_o[n]}\right) \qquad (2)$$

$$E_2 = \left\| \frac{X_o[n] - X_{dut}[n]}{X_o[n]} \right\|_2 \qquad (3)$$

where $X_o[n]$ is the Matlab FFT output and $X_{dut}[n]$ is the design under test output. It is also measured the performance of a 16 bit integer C++ FFT, known as *Kiss FFT*, as a testbench for the proposed core.

Table 1: E_∞ for 1024 simulations, random inputs

	1024 points	4096 points
R-2, Cordic	0.006	0.008
R-2, Mult.	0.003	0.108
R-4, Cordic	0.003	0.007
R-4, Mult.	0.002	0.105
Kiss FFT	0.017	0.035

Table 2: E_2 for 1024 simulations, random inputs

	1024 points	4096 points
R-2, Cordic	0.007	0.053
R-2, Mult.	0.004	0.131
R-4, Cordic	0.002	0.027
R-4, Mult.	0.003	0.126
Kiss FFT	0.017	0.035

The main requirement for the design is the low space/resource occupation. 16 bits iterative radix-2 and radix-4 architectures are synthesized and compared with a 16 bits radix-2 sdf and Xilinx's LogiCORE FFT v7.1.

Table 3: Resource occupation for 1024 points

	Slices	LUTs	Reg	LUTRAM
r2, cordic	855	2712	164	1024
r2, mult	659	1884	163	1024
r4, cordic	916	2862	165	1152
r4, mult	824	2241	260	1152
r2	3369	11386	1425	1056
Xilinx FFT	1050	2541	3684	

In order to meassure the ISDB-T utilization potential, the core is compared with an implementation made specifically for this use.
The proposed core has lower resource ocupation and provides scaling options.

Table 4: Comparison with ISDB-T oriented FFT IP Core

	Iter. r-2	Ref. IP core
FF	533	1334
LUT	3046	4133
BRAM	62	62
MUL		48
MHz	107	61

Conclusion and future work

This paper presented two iterative radix-r FFT computing cores, designed for OFDM communication systems, as are detailed below:

- Radix-2 iterative architecture.
- Radix-4 iterative architecture.
- Cordic algorithm for twiddle factors multiplications, for radix-2 and radix-4.
- Efficient complex multiplier for twiddle factors multiplications, for radix-2 and radix-4, as an alternative to cordic algorithm.
- Run time, stage selectible rounding/clipping module.

The cores fulfill the implementation requirements in terms of number of samples, run time configuration and scaling options.
The low space/resource requirement is achieved, which made them suitable for integration in large systems without impacting in the resource distribution, in case of FPGA implementation, or space in case of ASIC implementation.
The architectures were implemented in Verilog HDL code. Also there were developed several testing tools comprising Matlab, C++ programs and Verilog testbenchs.
For future work, it can be considered to add a dithering system, in order to reduce the noise generated by the architectures, and to implement a pipelined cordic without modifying the global architecture timing, in order to improve the throughput.

References:

[1] Cassagnes, A., Lutenberg, A., Giordano Zacchigna, F. (2016) "Implementacion y analisis de algoritmos de calculo de Transformada Rapida de Fourier para su aplicacian en sistemas OFDM" http://laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-Tesis-Grado-Andres-Cassagnes-2016.pdf
[2] Prasad, R. (2004), Orthogonal Frequency-Division Multiplexing. In *OFDM for Wireless Communications Systems* (p.p. 11-15), UK: Artech House.
[3] Melo, R., Salomón, F., Valinoti, B., (2016) "IP core FFT configurable en Runtime"

Contact:

Laboratorio de Sistemas Embebidos
Facultad de Ingeniería - UBA
Paseo Colón 850, CABA
E-mail: andres.cass@gmail.com