

# Efficient Circuit Architecture and FPGA Implementation for LTE Single Carrier FDMA DFT

J. Greg Nash, Senior Member, IEEE  
 Centar LLC, Los Angeles, CA USA jgregnash@centar.net

**Abstract**— A new memory-based circuit architecture for computing the DFT is presented and applied to the LTE SC-FDMA DFT protocol requirements. The implementation focuses on efficiently using the LUT/register fabric of FPGA-based hardware. The fastest available commercial design uses 29%/52% more LUT/registers while the proposed design is 40% faster in computing LTE resource blocks, a measure that reflects both circuit throughput and latency. It is programmed by simply entering parameter values into a single ROM memory so that any number of transform sizes, including powers-of-two, can be accommodated. The architecture provides scalable throughput by increasing the array size, high dynamic range and leads to simple, regular implementations.

**Keywords**— LTE; SC-FDMA; FPGA; DFT; Fast Fourier transform; discrete Fourier transform; non-power-of-two.

## I. INTRODUCTION

Single carrier frequency division multiple access (SC-FDMA) is a part of the LTE protocol used for up-link data transmission. It involves a discrete Fourier transform (DFT) pre-coding of the transmitted signal, where the DFT can be any one of 35 transform sizes  $N$  from 12-points to 1296-points, and  $N=2^a3^b5^c$  ( $a,b,c$  are positive integers). The rationale for targeting FPGAs is due to the rapidly growing FPGA use in communications applications, e.g., base stations and remote radio heads at the top of cell phone towers. Here we provide results of mapping the architecture to Xilinx Virtex and Altera Stratix devices.

FPGAs as an implementation platform have unique features such as large numbers of embedded multipliers and memories, leading to very different design tradeoffs compared to ASIC designs. In particular, embedded elements in such quantities make them almost “free”, compared to their ASIC implementation costs. Consequently, the design goal is to produce a circuit that minimizes the “expensive” FPGA look-up-table (LUT) and register fabric usage rather than embedded element usage. An additional motivation for this goal is that this fabric is also the source of most of the FPGA dynamic power consumption, as opposed to the embedded elements, an important consideration since FPGAs are increasingly being used in mobile devices.

## II. BACKGROUND

### A. FFT computing models

The proposed “memory-based” FFT model departs considerably from traditional memory-based designs as illustrated in Fig. 1. Here a traditional high-performance

memory-based design (Fig. 1a) contains physically separate arithmetic (butterfly) and data units. The goal in such designs is to sequence data to/from the memories in such a way that data I/O rates are maximized. Alternatively, the model proposed in this paper does the same thing, but at a finer level of granularity such that data is placed in close proximity to computing resources. As shown in Fig. 1b, this is done using many very small “processing elements” (PEs), each containing a multiplier/adder and a few registers. Since each PE reads and writes to a small, simple dual-port memory, aggregate bandwidth is limited only by the number of PEs. Additionally, well-known scalability of array structures means that high bandwidths, and thus performance, are achieved by simply increasing the array size. It is far more difficult to do the same for traditional memory based designs (Fig. 1a).

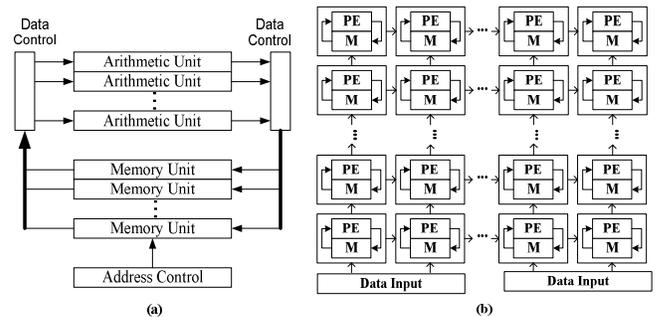


Fig. 1. (a) Traditional memory-based FFT architecture and (b) proposed fine-grained, locally connected, equivalent.

Fig. 1b also shows that each PE is locally connected to its (4) neighbors which keeps interconnections short, resulting in reduced power dissipation and higher clock speeds.

### B. Related Work

Both Xilinx and Altera [1] provide users of their FPGAs options to support the LTE SC-FDMA DFT protocol using a memory-based architecture as in Fig.1a, consisting of a single multi-port memory that sends/receives data to/from a single arithmetic unit that performs the required butterfly computations. For these designs the number of clock cycles per DFT is greater than the transform size  $N$ , so that it is not possible to continuously stream data into and out of the circuit.

A couple of other published designs are different from those discussed above in that they either use a higher radix memory-based design [2] or a pipelined architecture [3] to

reduce the overall number of cycles needed to compute a DFT to that of the actual transform size  $N$ .

### III. IMPLEMENTATION

For our proposed memory-based architecture transforms are performed using a 6x6 PE virtual array to compute the appropriate butterflies for the mixed radices needed. Since  $N$  can also be obtained from the expression  $N=2^a3^b4^c5^d6^e$ , where all exponents are positive integers, additional radices can be employed, improving computational efficiency compared to use of just 2,3 and 5 radices. Twiddle memory is minimized by “on-the-fly” generation of values. More implementation details are provided in [4].

### IV. COMPARITIVE ANALYSIS

#### A. Introduction

In order to provide a more relevant metric than throughput and latency numbers we calculate, where possible, the length of time necessary to compute an LTE resource block (RB). The RB is the minimum processing unit of data for the LTE protocol consisting of 7 symbols for (normal cyclic prefix). This is a better performance comparison metric in that it requires both low latency and high throughput for good results.

#### B. Commercial FPGAs

For comparison with Xilinx FPGAs, a Virtex-6 (XC6VLX75T-3) FPGA was used as the target hardware for both the Xilinx and the proposed circuit. The Xilinx LogiCORE IP version 3.1 was used to generate a 16-bit version of their DFT because the SQNR of 60.0 db (average over all 35 transform sizes) was comparable to the proposed circuit with average SQNR=61.3.

The resource comparisons in Table I use a Xilinx block RAM normalized to 18K bits, so that a Xilinx 36K block RAM is considered equal to two 18K RAMs. Also, the “RB Avg” column provides the average number of cycles (over all 35 DFT sizes) it takes to compute the DFT for the 7 symbols defined by a RB as a function of the transform size  $N$ . Finally, in Table I the Fmax (maximum clock frequency) value and the number of RB cycles are combined, providing a measure of the throughput (“Thrpt norm”), which is normalized to a value of “1” for the proposed design (higher is better). Table I then shows the Xilinx design uses 29%/52% more LUTs/registers while the proposed design provides a 40% higher RB computation throughput. So the overall combined gain is significant. The proposed design uses more embedded memory and multipliers, but this was less a consideration as discussed in Section I.

Altera does not offer a DFT LTE core as does Xilinx; however, they have published results of an example design running on a Stratix III FPGA that provides a useful basis for comparison. This design example is different than the proposed design here in that the outputs are not in normal order. Adding buffer circuitry to sort the output data would

require additional logic and add  $\sim N$  additional words of memory ( $\sim 5$  RAM blocks) to the numbers shown in Table I. (Stratix III block RAMs are 9K bits).

For comparison the proposed design was also targeted to a Stratix III FPGA of the same speed grade. The Altera implementation uses less logic, but is far slower, both in terms of the lower values of Fmax, and the increased number of cycles to complete the RB computation. Consequently, the proposed design has  $\sim 3x$  higher throughput while LUT usage is only  $\sim 47\%$  higher.

TABLE I. LTE CIRCUIT TECHNOLOGY COMPARISONS

Design	FPGA	LUT	Reg	BLK RAM	Mult 18-bit	Fmax (MHz)	RB Avg	Thrpt norm
<b>Proposed</b>	Virtex-6	2975	2853	19	72	401	16.6N	1
<b>Xilinx [1]</b>	Virtex-6	3851	4326	10	16	403	23.4N	0.71
<b>Proposed</b>	Stratix III	3816	3188	29	60	400	16.6N	1
<b>Altera [2]</b>	Stratix III	2600	N/A.	17	32	260	32.9N	0.33

#### C. Other FPGA implementations

Other published SC-FDMA implementations are compared in Table II for Virtex FPGAs. For the proposed architecture the average throughput as a function of  $N$  for all 35 transform sizes is  $2.1N$  vs.  $N$  the other two. However, these more complex architectures require far more LUT hardware, 162% and 262% for [2] and [3], respectively. Although [3] uses fewer registers, this is less meaningful because the 10:1 ratio of LUTs/registers in FPGA hardware leads to imbalances that can cause many registers to be inaccessible. Additionally, comparing “Thrpt norm” values in Table I (here based only on throughputs, since latency values weren’t supplied), they can be seen to be much slower designs.

TABLE II. LTE CIRCUIT TECHNOLOGY COMPARISONS

Design	FPGA	LUT	Reg	Blk RAM	Mult 18-bit	Fmax (MHz)	Thrpt (cycles)	Thrpt norm
<b>Chen [5]</b>	Virtex-5	7791	N/A	7	44	123	$N$	0.65
<b>Niras [6]</b>	Virtex-6	10768	786	45	41	61.3	$N$	0.32
<b>Proposed</b>	Virtex-6	2975	2853	19	72	401	2.1N	1

### V. CONCLUSION

We have shown how a new memory-based model for the FFT combines algorithm efficiency and programmability with new circuit features leading to higher throughputs, lower latencies and at the same time reduced LUT/register usage compared to other FPGA implementations.

### REFERENCES

- [1] Application Note: Xilinx DFT v3.1, DS615 Mar. 1, 2011 and Altera DFT/IDFT Reference Design, 464, May 2007.
- [2] J. Chen, J. Hu, and S. Li, “High throughput and hardware efficient FFT architecture for LTE application”, Proc. 2012 IEEE Wireless Communications and Networking Conf., pp. 826-83.
- [3] C.V. Niras and V. Thomas, “Systolic variable length architecture for discrete fourier transform in Long Term Evolution”, Int. Symp. on Electronic System Design, 2012, pp. 52-55.
- [4] J. G. Nash, “High-throughput programmable systolic array FFT architecture and FPGA implementations”, Int. Conf. on Computing, Networking and Communication, Honolulu, HI, Feb.2014, pp. 878-884.