

Efficient Split Radix FFTs in FPGAs

Tom Dillon
Dillon Engineering, Inc.

This presentation outlines methods for split radix FFTs implemented in FPGAs. Analysis of various algorithms with regards to performance, cost and power consumption are presented.

FPGAs are rapidly finding their way into high performance DSP applications, specifically real time signal processing applications. Large FPGAs offer a significant cost, size, and power advantage over other alternatives for many front end real time processing operations. FPGAs offer the advantage of short and flexible design cycles, high performance and relatively low NRE.

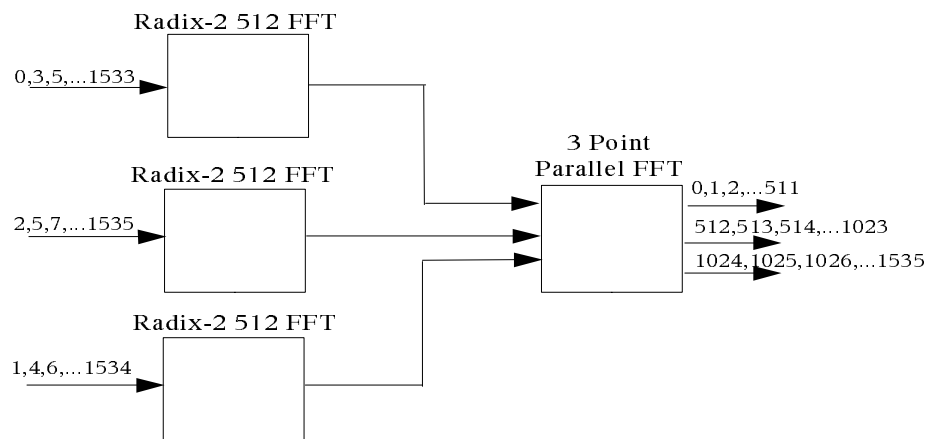
The FFT is at the heart of many real time signal processing applications, including radar, communication, and image processing. Logic for high speed FFTs can account for up to 90% of the cost and power of a given application, making efficient resource usage critical.

Architectures for radix-2 FFTs are well known and have been in use in FPGAs for some time with excellent results. Many applications require bin spacing that can't be achieved with radix-2 FFTs since one is limited to a power of 2 length. In order to attain more useful bin spacing, many times a split radix FFT architecture is used. For example, using a radix-2 with radix-3 architecture can produce 384 (128 x 3), 768 (256 x 3), 1536 (512 x 3) and so on. This architecture is not limited to radix-2 and radix-3, as with proper data routing structures any split radix combination can be implemented.

Two common methods for implementing split radix FFT architectures are Kolba-Parks and Cooley-Tukey. The advantages of both architectures pertaining to FPGA implementations will be detailed.

The basic structure of a 1536 point FFT architecture is given in the following diagram.

Figure 1: High Performance Parallel Radix-2 Times 3 Pt. Architecture
1536 Implementation



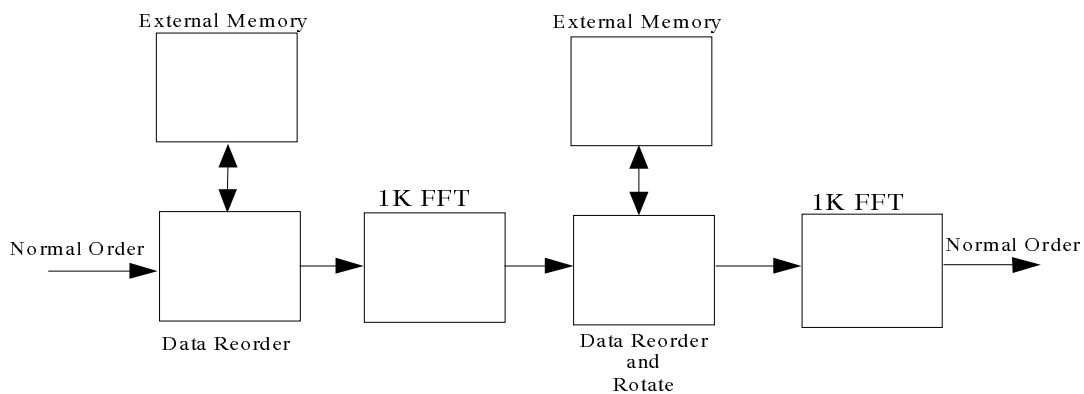
The size, cost, and power consumption of the FPGA are directly proportional to the throughput

of this FFT engine since logic is added as required to keep up with processing requirements. Some real examples of FPGA device usage, power consumption and cost will be presented. The architecture as shown accepts 3 inputs and produces 3 outputs per clock cycle, but can easily be extended for higher or lower performance applications. Clock rates in excess of 200MHz can be used with today's FPGA technology.

Another simple extension to this architecture is for ultra long FFTs. A similar architecture has been used to produce 512K and even 1M point FFTs in FPGAs with the simple addition of external memory to store intermediate results. Longer lengths only require more external memory.

The basic structure is shown below and will be presented with device usage, power and cost examples.

Figure 2: Ultra Long FFT Architecture
1M Continuous Data



A strong case is made for using FPGA technology for FFT processing in real time DSP applications.