Datasheet: Hardwired Floating-Point FFT (Arria/Stratix 10)

Features

- ***Floating-Point Implementation***: Uses Intel's new Arria 10 and Stratix 10 FPGA floating point primitives, reducing LUT and register usage by more than a factor of two for Centar's equivalent fixed-point IP and almost a factor of six for Intel's.
- ***Applications:*** Eliminates design costs and time required to do fixed point implementations
- ***FFT size***: User chosen power-of-two or non-power-of-two
- ***Programmability***: Easy to change functionality to meet application requirements.
- ***Dynamic Range***: single precision floating-point
- ***Scalability***: array based architecture means arbitrarily higher throughputs are obtained by increasing array size
- ***Power***: array interconnects are entirely local, reducing parasitic routing capacitance to keep power dissipation low and clock speeds high
- ***Inverse FFT:*** Run time selectable input

I/O

- IEEE754 floating point input, output and internal processing

Algorithm

The transform computation is based on a new formulation[1] of the discreet Fourier transform (DFT), different than any other FFT implementation, which decomposes it into structured sets of small, matrix-based DFTs. In particular the locality, simplicity and regularity of the processing core keeps interconnect delays lower than cell delays, leading to clock speeds that can approach the FPGA fabric limitations, e.g., "worst case" Fmax speeds determined by embedded elements. Short critical-path lengths with less delay in cells than interconnects also lower power dissipation. Additionally, a novel "base-4" algorithm reduces the number of cycles needed per FFT to less than the transform size value *N.*

Architecture

What is unique about this architecture is that the architecture can be mapped directly to the basic hardwired floating-point primitives supported by the Intel Arria and Stratix 10 FPGAs. in

The architecture consists of two small arrays of pipelined, fine-grained, locally connected, simple processing elements, all containing one of the three primitives:

- floating-point adder
- floating-point accumulator
- floating-point multiplier

[1] J. Greg Nash, "Computationally Efficient Systolic Array for Computing the Discreet Fourier Transform, IEEE Trans. Signal Processing, Vol. 53, No.12, December 2005, pp.4640-4641.

Because these exact structures are supported directly in Arria/Stratix 10 hardware, most of the FFT computation can be performed entirely within the embedded DSP blocks. This reduces LUT and register uses by a factor between 2 and 4.

Precision

"Arria® 10 FPGAs and SoCs are the industry's first FPGAs and SoCs that natively support single-precision floating-point DSP block mode as well as standard- and high-precision fixed-point computations using dedicated hardened circuitry. The single-precision floating-point DSP block mode is IEEE 754 compliant and comprises of an IEEE 754 single-precision floating-point adder and IEEE 754 single-precision floating-point multiplier as shown in Figure 1. The new Arria 10 single-precision floating-point DSP block mode allows you to implement algorithms in floating point with efficiency and power comparable to fixed-point operations. The productivity benefits with this DSP block architecture in Arria 10 FPGAs and SoCs makes them a compelling alternative to graphic processing units (GPUs) for high-performance computing applications"[2].

Stratix 10

"FPGAs enjoy a well-deserved reputation for highly parallel, high-throughput digital signal processing (DSP). This capability has been steadily increasing over past generations of FPGA devices. However, occasionally a revolutionary, rather than evolutionary, new product is introduced. Intel's new Stratix® 10 FPGA and SoC family certainly fits that description. Stratix 10 devices deliver up to 23 TMACs of fixed-point performance and up to 10 tera floating point operations per second (TFLOPS) of single-precision floating-point performance making these devices the highest performance DSP devices with a fraction of the power of alternative solutions, such as graphic processing units (GPUs) and dedicated DSP. Stratix 10 customers can expect to see up to an order of magnitude improvement in giga floating point operations per second (GFLOPS)/Watt in their actual designs compared to competitive GPU solutions"[3].

Performance and Resources

Here, example performance and resource usage data are provided for a 1024-point, streaming (normal order input and output) FFT. The goal of the implementation was to minimize the usage of FPGA register/LUT fabric, since FPGAs are available in a variety of versions containing substantial numbers of embedded elements such as memory and arithmetic.

The circuits were compiled using Intel's software tools (Quartus II v16.1) using a Arria10 10AS066H1F34E1SG FPGA (20nm technology) device. The TimeQuest static timing analyzer was used to determine maximum clock frequencies (Fmax) at 1.1V and 85C (worst case settings). The same Quartus settings were used for both the Centar and Intel designs (IP v16).

[2] Altera white paper WP-01267-1.0, "Enabling High-Performance Floating-Point Designs" 2016.
[3] "Stratix 10: The Most Powerful, Most Efficient FPGA for Signal Processing", https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/backgrounder/stratix10-floating-point-backgrounder.pdf

	Intel	Centar	
Transform Size		V1	V2
ALMs	4852	2241	4106
ALUTs	6058	3531	6795
Registers	10844	4969	6121
M20Ks	20	62	30
MLAB Memory Bits	4136	4776	70,312
DSP Blocks	64	96	96
Fmax (data rate, MHz)	432	585	572

In the table above the adaptive logic module (ALM) is the basic unit of a Arria FPGA (two 4-input adaptive LUTs, four registers plus other logic.

To illustrate the flexibility possible in choosing resources, two versions of the circuit are shown: "V1" uses M20K memories for the internal FFT engine working memory and "V2" uses MLABs for this. Thus, "V1" shows that the Intel version uses approximately twice the number of register and LUT fabric resources as that from Centar while using a third the number of M20Ks. "V2" trades off embedded memory (M20Ks) for LUTs, but still uses 15% fewer ALMs.

In both cases in the table the Centar circuit provides about 36% higher throughputs.

Device Family Support

Intel Device Families Supported

- Stratix 10
- Arria 10

Xilinx Device Families Supported

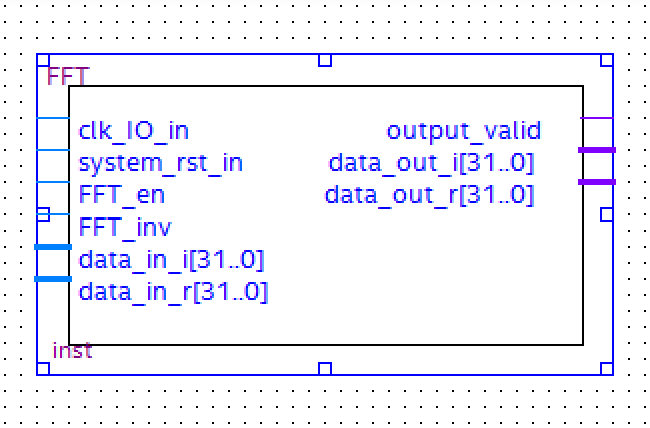
- None

Deliverables

- Netlist (e.g., for Intel FPGAs a *.qxp file for synthesis or a *.vo or lib file for simulation)
- Synthesis constraints (e.g., for Intel FPGA's an *.sdc file)
- Modelsim Testbench (*.vo file for DFT circuit plus verilog testbench for control). Matlab verification utilities also available.
- Intel Stratix III FPGA board development kit testbench
- Matlab behavioral bit-accurate model (p-code)
- Documentation

Pin-outs

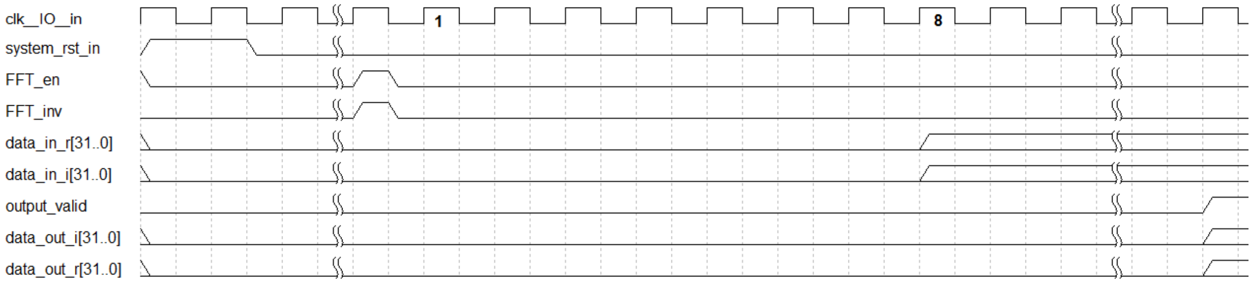
A symbol list corresponding to the pin-outs shown below are provided in the accompanying table. Depending upon the desired interfaces, some signals could change.



Name	Signal	Description
clk_IO_in	clock	Can be either a low frequency board oscillator clock output in which case circuit clock is derived from a PLL or actual data I/O clock
system_rst_in	control	Resets circuit; active high
FFT_en	control	Registers FFT_inv signal; active high
FFT_inv	control	High->forward; low->inverse
data_in_r/i	32-bit IEEE 754	Real and imaginary inputs
output_en	control	High during data output
data_out_r/i	32-bit IEEE 754	Real and imaginary outputs

Timing Diagram

A variety of timing possibilities exist depending upon the desired interface. That shown here is applicable to a streaming, normal order input/output scheme.



At any time after system_rst_in goes low, FFT_ en is used to latch the direction of the transform (FFT_inv=0/1 for forward/inverse). Following this, the circuit expects to see continuous data appearing on the 8th subsequent cycle. Valid data out occurs when output_valid goes high. For streaming operation output is continuous from this point on.