24K FFT for 3GPP LTE RACH Detection

November 2008, version 1.0 Application Note 515

Introduction

In 3GPP Long Term Evolution (LTE), the user equipment (UE) transmits a random access channel (RACH) on the uplink to gain access to the network. One method to extract this UE RACH signal at the basestation is to perform a 24K FFT on the received data, allowing the desired bands to be extracted in the frequency domain.

This reference design demonstrates that you can perform large FFT transforms on Altera FPGAs. This application note describes how a 24K point transform may be decomposed into three 8,192 point transforms and then implemented efficiently using a multiple channel FFT architecture.

Architecture

This architecture reduces the total memory complexity of the design. The 24K FFT is decomposed into three 8K transforms that are combined together. Many different architectures are possible but because this algorithm has such a high memory requirement, it uses the architecture that uses the minimum memory.

By decomposing the transform using a decimation in time approach, it combines the three adjacent samples without the requirement for additional buffering (the additional advantage is no further latency is incurred). The following radix 3 decimation in time decomposition equation shows how to achieve the full transform by performing three separate transforms that are combined by summing and applying a multiplication by additional twiddle weights.

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

$$= \sum_{m=0}^{\frac{N}{3}-1} x(3m) \cdot W_N^{k(3m)} + \sum_{m=0}^{\frac{N}{3}-1} x(3m+1) \cdot W_N^{k(3m+1)} + \sum_{m=0}^{\frac{N}{3}-1} x(3m+2) \cdot W_N^{k(3m+2)}$$

$$= \sum_{m=0}^{\frac{N}{3}-1} x(3m) \cdot W_{\frac{N}{3}}^{km} + W_N^{k} \sum_{m=0}^{\frac{N}{3}-1} x(3m+1) \cdot W_{\frac{N}{3}}^{km} + W_N^{2k} \sum_{m=0}^{\frac{N}{3}-1} x(3m+2) \cdot W_{\frac{N}{3}}^{km}$$

$$= \mathrm{DFT}[x(3m)] + W_N^{k} \cdot \mathrm{DFT}[x(3m+1)] + W_N^{2k} \cdot \mathrm{DFT}[x(3m+2)]$$

where:

N is the number of points
n is 0 to $N-1$
k is 0 to $N-1$
W_N^{kn} is simplified notation for $e(-j2\pi kn/N)$

The design implements the decimation in time FFT using a multiple channel streaming FFT architecture, which has a memory complexity of N complex words plus small ROMs for twiddle factors. The data is in bit-reversed format, to reduce memory complexity. The design applies the digit reversal at the output of the 24K DFT, so you can consider the additional digital reversal associated with the radix 3 combination stage.

The twiddle generator is an optimized architecture that exploits the symmetry of the cosine and sine waveforms and requires a total of $N/4$ real words.

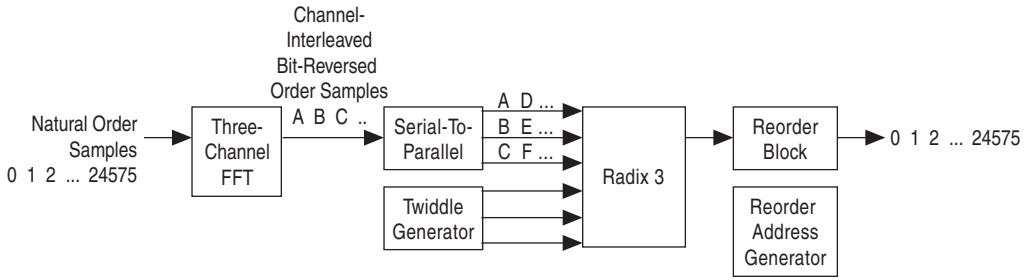
The reorder address generator applies digit reversal so that the output frame is in natural order. This operation requires a total of N complex words. 24K complex words is a significant amount, and so to allow full streaming operation, this component is not double buffered.

Although the multiple channel FFT component complies with bit-growth requirements associated with this algorithm, in practice the full natural bit growth is not required. The design constrains the output bit width of the FFT to 23 bits. The radix 3 combination stage is performed at full precision, and the design uses a convergent rounding block (with saturation) at the output to rescale the data back to 16 bits.

Additional logic must control the input and output interfaces and handle the forward and reverse flow control.

Figure 1 shows the design block diagram.

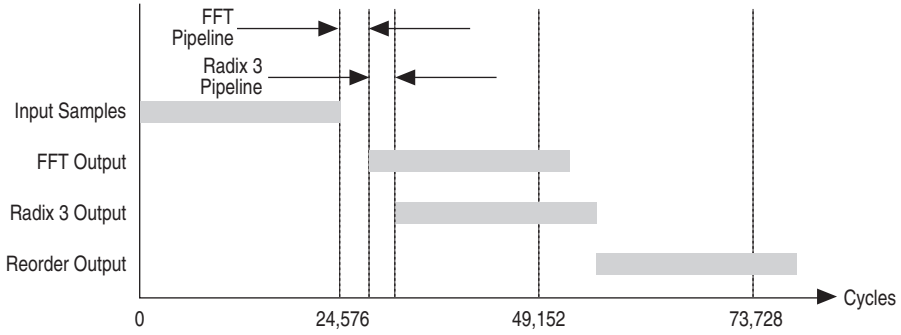
Figure 1. 24K DFT Block Diagram



Latency

Figure 1 shows the latency specifications of the design.

Figure 2. Latency Specifications



The pipeline delay associated with the FFT function is approximately 56 clock cycles. The pipeline delay associated with the radix 3 combination stage is approximately 10 clock cycles.

The multichannel FFT has an inherent algorithmic latency of N (= 24,576) clock cycles; the reordering block also has an inherent algorithmic latency of N clock cycles. As a result, the total transform latency is calculated as:

Latency = $2N$ + *FFT pipeline* + *radix 3 pipeline* ≈ 49,218 cycles
(316 µs at 156 MHz)

Total transform = $2N + N$ + *FFT pipeline* + *radix 3 pipeline* ≈ 73,789 cycles
(473 µs @ 156 MHz)

Latency is the best case value that assumes the input FFT frame is provided in a burst over 24K clock cycles, and the downstream component does not apply backpressure. Forward and reverse stalling increases the overall transform latency.

Getting Started

This section describes the following topics:

System Requirements

The reference design requires the following software:

- Quartus® II software version 7.2
- ModelSim SE version 6.3G

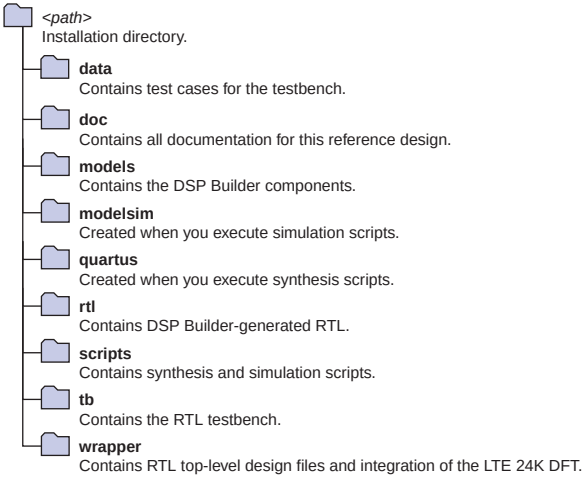
Install the Reference Design

To install the reference design, run the <*filename*> file and follow the installation instructions.

The reference design installs by default in the **c:\altera\reference_designs\lte_24k_dft** directory. You can change the default directory during the installation.

Figure 3 shows the directory structure after installation.

Figure 3. Directory Structure



Simulate the Design

To simulate the design, follow these steps:

1. In the script directory, open the <*filename*>**_msim.tcl** script, and modify the `proj_topdir` variable so that it reflects the physical location of the project on your local disk.

2. Start the ModelSim simulator. On the Tools menu click **Execute Macro** and browse to the <*filename*>**_msim.tcl** script.

A wave window opens and displays the results.

Synthesize the Design

To synthesize the design, follow these steps:

1. Start the Quartus II software.

2. On the File menu, click **Open**. Browse to the scripts directory and open the <*filename*>_quartus.tcl script.

3. Type the following command in the Tcl console to create an appropriate Quartus II project.

```
source <filename>_quartus.tcl
```

4. On the Processing menu, point to **Start** and click **Start Synthesis and Analysis** to synthesize the design.

Performance

Table 1 shows the performance in a EP2S90C4 Stratix® II device.

Table 1. Performance

Combinational ALUTs	Registers	Memory				18 ×18 Multipliers	f_{MAX} (MHz)
		Bits	M512	M4K	MRAM		
5173	6273	1,837,889	20	293	2	68	199

Interface Specification

The LTE 24K DFT design supports the Avalon® Streaming (Avalon-ST) protocol with a ready latency of zero; the same as the Altera FFT MegaCore® function.

The sink interface accepts 24K samples at a time. The sink interface supports forward flow control. After the sink interface has accepted 24K valid samples, the `sink_ready` signal goes low.

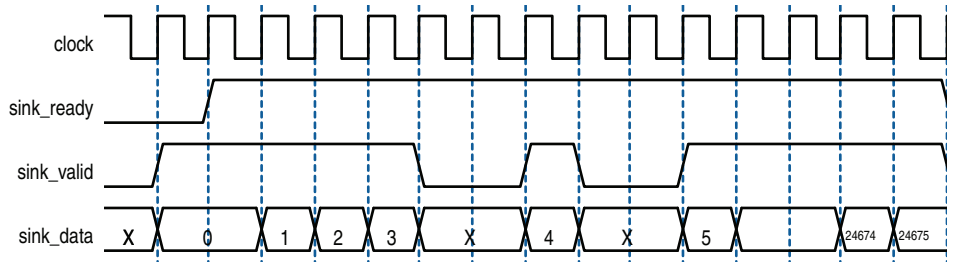
Eventually the module completes processing, and all 24K samples are available in the internal reordering output buffer. Assuming that the downstream module is ready to accept the data, this buffer presents one complex sample per clock at the output. This source interface also supports flow control (backpressure).

When all of the 24K samples have been read from the reordering memory the sink is ready to accept new data samples. The design does not need to double buffer the output reordering stage, which saves significant additional memory resources (24K complex samples).

The core architecture is not fully streaming, so the interfaces do not support start and end of packet signals.

The sink interface samples an incoming valid sample when it asserts the `sink_ready` signal. As a result, the source should provide data and assert `sink_valid` whenever it has data available. The value should be held until `sink_ready` is asserted high to acknowledge that the sample has been read. In this design when `sink_ready` is asserted, it remains asserted until the external design provides 24,576 valid samples.

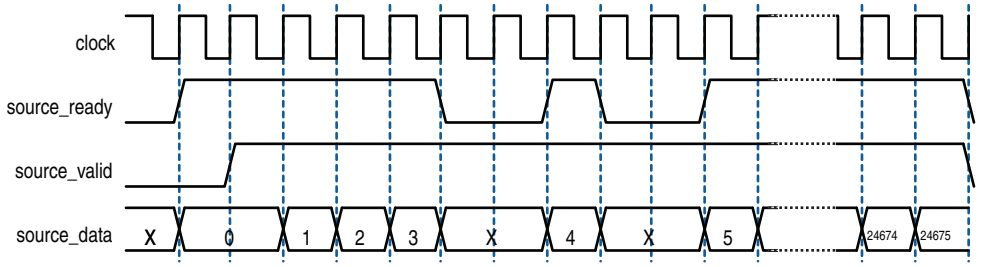
Figure 4. Sink Interface Timing Diagram



The source interface asserts the `source_valid` signal with one complex sample. The value is held until the `source_ready` signal is asserted by the downstream component. In this design when `source_valid` is asserted, all of the 24,576 samples are available and so `source_valid` remains high until the downstream block acknowledges the transfer of every sample. When all samples are transferred, the `sink_ready` signal is asserted again.

The `source_real0` and `source_imag0` ports have data on them during processing because they are connected directly to the internal dual-port memory. As the design writes values to this memory, they appear at the output. They should be ignored unless `source_valid` is asserted.

Figure 5. Source Interface Timing Diagram



The top-level entity **lte_24k_dft_toplevel** is:

```
port (
sink_ready : out std_logic_vector(0 downto 0);
sink_valid : in std_logic_vector(0 downto 0);
sink_real : in std_logic_vector(15 downto 0);
sink_imag : in std_logic_vector(15 downto 0);
```

```
 source_ready : in std_logic_vector(0 downto 0);
 source_valid : out std_logic_vector(0 downto 0);
 source_real0 : out std_logic_vector(15 downto 0);
 source_imag0 : out std_logic_vector(15 downto 0);
 clk : in std_logic;
 reset : in std_logic
 );
end;
```

Conclusion

This application note has outlined how to decompose a 24K DFT transform into three FFT transforms that are implemented on an Altera FPGA using a multi-channel FFT architecture. You can use this approach and technique to implement other non-power of two DFT transforms.

Document Revision History

Table 2 shows the revision history for this application note.

Table 2. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2008 v1.0	First release.	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support/
Literature Services:
literature@altera.com