



AN 464: DFT/IDFT Reference Design



Contents

About the DFT/IDFT Reference Design.....	3
Functional Description for the DFT/IDFT Reference Design.....	4
Parameters for the DFT/IDFT Reference Design.....	7
Signals for the DFT/IDFT Reference Design.....	8
Getting Started with the DFT/IDFT Reference Design.....	10
System Requirements for the DFT/IDFT Reference Design.....	10
Installing the DFT/IDFT Reference Design.....	10
Simulating the DFT/IDFT Reference Design with the Avalon-ST Testbench.....	11
Simulating the Reference design with the Command Line Regression Test.....	12
Synthesizing the DFT/IDFT Reference Design.....	12
Running the Fixed-Point Model for the DFT/IDFT Reference Design.....	14
Fixed-Point Model Parameters.....	14
Performance of DFT/IDFT Reference Design.....	16
Document Revision History for AN 464: DFT/IDFT Reference Design.....	17



About the DFT/IDFT Reference Design

The DFT/IDFT reference design performs a discrete Fourier transform (DFT) or an inverse DFT (IDFT) of a complex input sequence and produces a complex output sequence.

The reference design performs the functions for either a DFT in the uplink or an IDFT in the downlink of a typical 3G long-term evolution (LTE) physical interface (PHY) implementation

The design inputs the transform length coincident with the first validated data sample. The design can configure the transform length at runtime (on a block-by-block basis) to any one of the 53 sizes specified by *3GPP TS 36.101 version 8.29.0 Release 8..*

The reference design uses Avalon[®] Streaming (Avalon-ST) interfaces for the inputs and the outputs. The input samples are in an integer format; the output samples are in block floating point format.

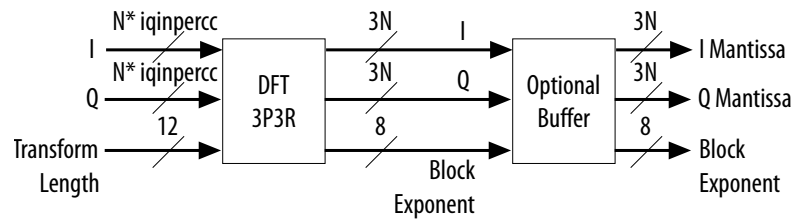
Parameters specify the transform mode (DFT or IDFT) and internal bit widths for the datapath and the twiddle value precision.

The reference design targets at Intel[®] Stratix[®] 10 devices and meets typical latency requirements while minimizing power.

Functional Description for the DFT/IDFT Reference Design

The input is an Avalon-ST sink interface with a ready latency of 1. The design does not use the eop signal at the input as it expects packets of exactly length. The transform commences within the design after you provide all length samples.

Figure 1. Block Diagram



The DFT 3P3R block decomposes the input block into three sub-blocks and performs three parallel DFT operations. The DFT results feed a parallel radix-3 pipelined engine for the final radix-3 pass that computes the final DFT result. The DFT radix-3 engine output 3 IQ samples per clock cycle.



Transform Size	Block-to-Block Latency (cycles)		Input-to-Output Latency	
	1 IQ input sample/ clock cycle	3 IQ input samples/ clock cycle	(cycles) 1 IQ input sample/clock cycle	(μ s) 1 IQ input sample/clock cycle
72	266	218	334	0.679
96	323	259	407	0.828
108	351	279	443	0.901
120	380	300	480	0.976
144	436	340	552	1.123
180	521	401	661	1.344
192	549	421	697	1.418
216	702	558	866	1.762
240	662	502	842	1.713
288	895	703	1107	2.252
300	803	603	1023	2.081
324	991	775	1227	2.496
360	1088	848	1348	2.742
384	1152	896	1428	2.905
432	1280	992	1588	3.230
480	1409	1089	1749	3.558
540	1569	1209	1949	3.965
576	1665	1281	2069	4.209
600	1730	1330	2150	4.374
648	2098	1666	2550	5.188
720	2050	1570	2550	5.188
768	2178	1666	2710	5.513
864	2747	2171	3343	6.801
900	2531	1931	3151	6.410
960	2691	2051	3351	6.817
972	3071	2423	3739	7.607
1080	3396	2676	4136	8.414
1152	3612	2844	4400	8.951
1200	3332	2532	4152	8.447
1296	4044	3180	4928	10.026
1440	4477	3517	5457	11.102
1500	4133	3133	5153	10.483
1536	4765	3741	5809	11.818
1620	5017	3937	6117	12.445
<i>continued...</i>				



Transform Size	Block-to-Block Latency (cycles)		Input-to-Output Latency	
	1 IQ input sample/ clock cycle	3 IQ input samples/ clock cycle	(cycles) 1 IQ input sample/clock cycle	(μ s) 1 IQ input sample/clock cycle
1728	5341	4189	6513	13.250
1800	5558	4358	6778	13.789
1920	5918	4638	7218	14.685
1944	6662	5366	7978	16.231
2160	6638	5198	8098	16.475
2304	7070	5534	8626	17.549
2400	7359	5759	8979	18.267
2592	8823	7095	10571	21.506
2700	8259	6459	10079	20.505
2880	8799	6879	10739	21.848
2916	9903	7959	11867	24.143
3000	9160	7160	11180	22.745
3072	9375	7327	11443	23.280
3240	10984	8824	13164	26.782

If you turn on the output buffer, the ready input controls the data flow out of the buffer. The buffer always fills in bursts as the DFT engine completes individual transforms so you must ensure the buffer depth is adequate for your system to avoid overflow. The design behaves unpredictably if the buffer overflows.

The design provides the `sop` and `eop` signals at the output, which you can optionally use.

To perform an IDFT, set the `idft_mode` parameter to 1.

The design does not perform the $1/\text{length}$ scaling for an IDFT.

Related Information

[Avalon Interface Specification](#)

Parameters for the DFT/IDFT Reference Design

Table 2. Parameters

Name	Values	Description
<code>idft_mode</code>	0 or 1	0 = DFT, 1 = inverse DFT.
<code>iqinpercc</code>	1 or 3	Input. IQ samples per clock cycle.
<code>datawidth</code>	16 .. 24	Internal datapath and input widths in bits.
<code>twidwidth</code>	14 .. 24	Twiddle factor and butterfly weight coefficient widths in bits (internal to the design).
<code>use_output_buffer</code>	0 or 1	0 = no output buffer is instantiated and the output interface does not have a ready input (<code>source_ready_inis</code> inactive).

continued...



		1= an output buffer is instantiated, which allows the reading of data from the design to be paused. The ready latency is 0.
BUFFER_DEPTH	4 .. 2048	Ifuse_output_buffer=1, this parameter defines the buffer depth. Each buffer location corresponds to one pair of complex (I and Q) samples.
BUFFER_DEPTH_N	2 .. 11	Set so that 2BUFFER_DEPTH_N>= BUFFER_DEPTH For example, if BUFFER_DEPTH= 600, set BUFFER_DEPTH_Nto 10.

Signals for the DFT/IDFT Reference Design

Table 3. Signals

Name	Direction	Description
a_reset_n	Input	Active low asynchronous reset. Deassert synchronously to clkto avoid reset-release timing violations.
clk	Input	System clock. All logic in the design is synchronous to this clock.
exponent[7:0]	Output	Avalon-ST source data. Exponent of all samples in the packet.A single value is valid throughout the packet on this output.
sink_eop	Input	Avalon-ST sink end of packet. The design does not use this signal.
sink_imag[datawidth-1:0]	Input	Avalon-ST sink data. Imaginary (Q) input sample.
sink_imag[iqinpercc * datawidth-1:0]	Input	-
sink_length[10:0]	Input	Transform length. A straight binary encoding of one of the lengths.The value on this input is sampled when sink_valid is high and sink_ready was high on the previous cycle.
sink_ready	Output	Avalon-ST Ready. The input interface has a ready latency of 1.
sink_real[datawidth-1:0]	Input	Avalon-ST sink data. Real (I) input sample.
sink_real[iqinpercc * datawidth-1:0]		
sink_sop	Input	Avalon-ST sink start of packet.
sink_valid	Input	Avalon-ST sink valid.
source_eop	Output	Avalon-ST source end of packet.
source_imag[datawidth- 1:0]	Output	Avalon-ST source data. Mantissa of Imaginary (Q) output sample.
source_imag_eng1[datawidth-1:0]	Output	Avalon-ST source data for dft engine 1.Mantissa of Imaginary (Q) output sample.
source_imag_eng2[datawidth-1:0]	Output	Avalon-ST source data for dft engine 2.Mantissa of Imaginary (Q) output sample.
source_imag_eng3[datawidth-1:0]	Output	Avalon-ST source data for dft engine 3.Mantissa of Imaginary (Q) output sample.
source_real_eng1[datawidth-1:0]	Output	Avalon-ST source data for dft engine 1.Mantissa of Real (I) output sample.
source_real_eng2[datawidth-1:0]	Output	Avalon-ST source data for dft engine 2.Mantissa of Real (I) output sample.
source_real_eng3[datawidth-1:0]	Output	Avalon-ST source data for dft engine 3.Mantissa of Real (I) output sample,

continued...



Name	Direction	Description
source_ready_in	Input	Avalon-STsource ready. This signal is only enabled if the parameter use_output_buffer is set to 1. The ready latency is0.
source_sop	Output	Avalon-ST source start of packet.
source_valid	Output	Avalon-ST source valid.



Getting Started with the DFT/IDFT Reference Design

System Requirements for the DFT/IDFT Reference Design

The reference design requires the following hardware and software:

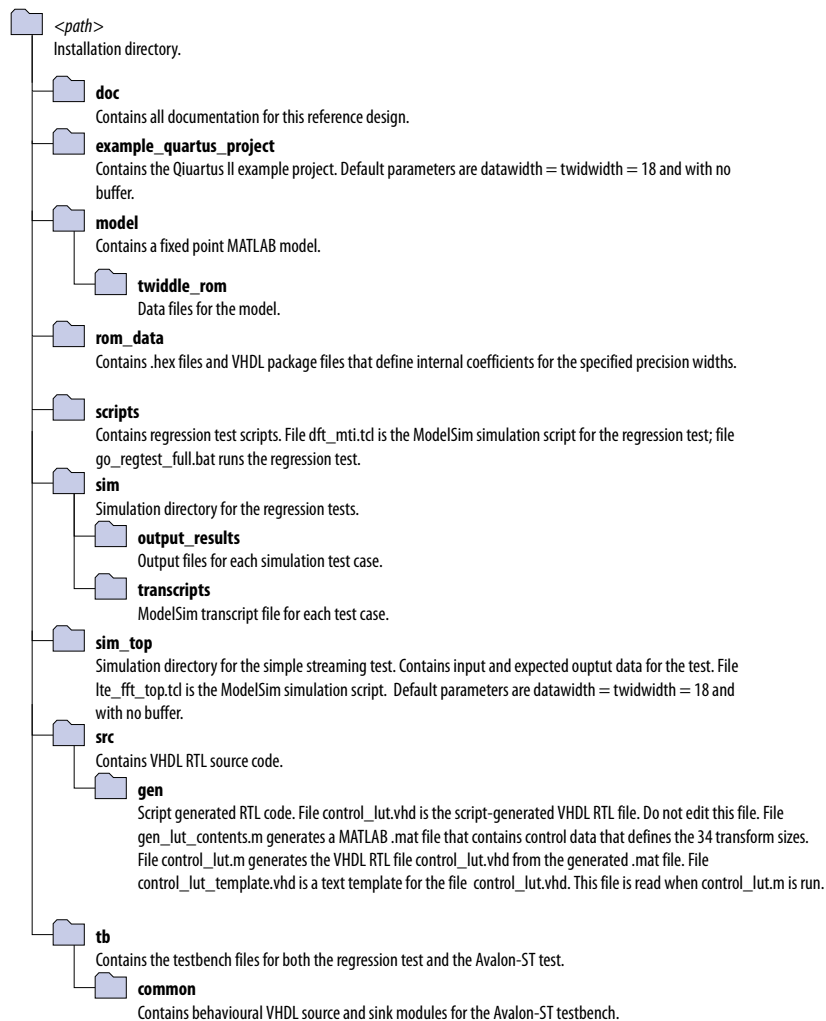
- MATLAB version R2016b
- Intel Quartus® Prime version 17.1

Installing the DFT/IDFT Reference Design

1. Run the an464.zip file.

You may need to add the /modeldirectory to the MATLAB path.

Figure 3. Directory Structure



Simulating the DFT/IDFT Reference Design with the Avalon-ST Testbench

The Avalon-ST testbench and script uses a single fixed set of parameters and runs a sequence of transforms with the transform length changing. This testbench quickly demonstrates the integration of the reference design and the Avalon-ST interfaces.

1. Open the ModelSim simulator.
2. Change the directory to `/sim_top`.
3. Type the following command:

```
source ltefft_top.tcl
```

Ignore the address out of range warnings.

4. Compare the output file `output_data.txt` with the expected output `expected_output_data_random_16_blocks.txt`, to ensure they match except for the block length comments in the expected file.



The test input file is `input_stream_random_17_blocks.txt`. The sequence of sizes are listed in the `readme_random_17_blocks.txt` file in the `\sim_top` directory.

Simulating the Reference design with the Command Line Regression Test

The command-line regression test runs different combinations of parameters with different transform lengths. The transform results are compared to the output of a MATLAB fixed-point model fed with the same input data. This regression test takes much longer to run.

This task only exercises the kernel and does not use the Avalon-ST interfaces.

1. Open a command prompt.
2. Type the following commands, to run the 68 tests defined in file `gen_configurations/core_configurations_to_test_main_full.txt`:

```
cd <install_path>/dft/scriptsr  
go_regtest_fullr
```
3. Wait until `*** All Done. ***` is displayed.

Synthesizing the DFT/IDFT Reference Design

Intel provides an example project in the directory `/example_quartus_project`.

1. Open file `/src/new/ltefft_top.vhd` and edit the following top-level parameters as desired.

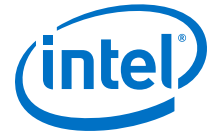
```
entity ltefft_top is  
  generic (  
    idft_mode           : integer := 0;  
    datawidth          : positive := 18;  
    twidwidth          : positive := 16;  
    iqinpercc          : positive := 3; -- input IQ samples per  
  CC  
    use_output_buffer  : boolean := false; --true; --false;  
    BUFFER_DEPTH      : integer := 1200; --8; -- 4  
  8  16  32  64  128  256  512  1024  2048  
    BUFFER_DEPTH_N    : integer := 11; --3; -- 2  3  
  4  5  6  7  8  9  10  11  
    use_90_optimised_twiddle_rom : integer := 1 -- must always be = 1  
  );
```

Note: These default settings have no effect on the simulation runs.

2. Copy all `.hex` files from the `/rom_data/bits_<twidwidth>/` directory to the `/example_quartus_proj/hex_rom` directory.
Skip this step if you are not changing the default parameters.
3. Copy the file `butterfly_coef_pkg.vhd` from the `dft/rom_data/bfly_coef_pkgs/bits_<twidwidth>` directory.
Skip this step if you are not changing the default parameters.
4. Open the Intel Quartus Prime software.



5. Open the project in the `/example_quartus_proj/` directory.
6. Select a device.
7. Compile the design.



Running the Fixed-Point Model for the DFT/IDFT Reference Design

The fixed-point MATLAB model allows fast bit-accurate simulation of the design's arithmetic behavior. The model is as a MATLAB function `dft_3p3r_model()` and is defined in the file `dft_3p3rmodel.m`. This top-level file and the sub-function MATLAB files are in the `/model` directory. Intel provides a MATLAB example that calls the model in file `simple_dft_example.m`

1. Open the MATLAB software.
2. Change the directory to `/model`.
3. Type the following command: `simple_dft_example`
4. Open the results file and examine the data.

The results file is: `<DFT
type>_length<length>_data<D_top>bit_tw<T_top>bit_rand<seed>.txt`

The `simple_dft_example.m` file defines the parameters.

Fixed-Point Model Parameters

The `simple_dft_example.m` file defines the parameters for the DFT/IDFT Reference Design fixed-point model.

The following pairs of columns are from left to right:

1. Input data
2. Model block Floating Point output.
3. Model integer output.
4. MATLAB output.
5. Absolute Difference between the Model and MATLAB outputs
6. Percentage difference between the Model and MATLAB outputs.

In each pair of columns the left hand column is the real data and the right hand column is the imaginary data. All data is in normal order with time 0 and bin 0 on the first row.

The model and the reference design do not perform $1/\text{length}$ scaling when in IDFT mode, so the $1/\text{length}$ scaling is removed from the MATLAB results by multiplying all samples by the length.

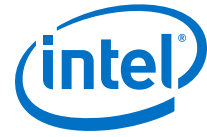


Table 4. Model Output Parameters

Variable Name	Type	Description
vec_out	Array of complex integers	Complex output array in normal order (provided disable_reordering = 0) Array length is the same as input array 'vec'
blk_exponent	Integer (-127 to +127)	Exponent value common to all values in output array vec_out. If you use the model for system modelling without reference to the RTL, set invert_sign_of_exponent to 0 so that the output sample values are: $vec_out \times 2^{blk_exponent}$

Table 5. Model Input Parameters

Variable Name	Type	Default Value ⁽¹⁾	Description
vec	Array of complex integers	-	Input array to be transformed. The model performs a transform of length equal to the input array length. The array length must be one of the 34 lengths.
model_is_fixed_point	0 or 1	1	0 = double precision model; 1 = fixed point model.
idft_mode	0 or 1	-	0 = DFT; 1 = IDFT.
B_top	Integer 14 to 24	T_top ⁽²⁾	Butterfly coefficient precision in bits.
D_top	Integer 14 to 24	-	RAM data path width in bits. This value is also the input data width and the mantissa width of the output samples.
T_top	Integer 14 to 24	-	Twiddle coefficient precision in bits.
bfly_mult_convergent_mode	0 or 1	1	Rounding mode at outputs of butterfly complex multipliers: 0 = symmetrical round-up 1 = convergent rounding
tw_mult_convergent_mode	0 or 1	1	Rounding mode at outputs of twiddle complex multipliers: 0 = symmetrical round up 1 = convergent rounding ⁽³⁾
limit_bfp_mux_size	0 or 1	1	0 = BFP scaling is done to the full precision of width D_top width. 1 = BFP scaling is limited to max_num_of_bfp_msb_shift_bits down from the MSB. ⁽³⁾
max_num_of_bfp_msb_shift_bits	2 to (D_top- 1)	3	BFP scaler size. ⁽³⁾
skip_bfp_on_last_pass	0 or 1	1	0 = BFP scaling is performed on all reads from data RAM; 1= BFP scaling is omitted for the last read from data RAM. ⁽³⁾

-
- (1) Typical settings to use when running the model or default value that you should use so that the model matches the RTL behavior.
- (2) In the RTL, the value of B_top is always equal to the value set for T_top.
- (3) These parameters are only valid for `fixed_point_model = 1`.



Performance of DFT/IDFT Reference Design

Table 6. Performance

The table shows the resource utilization of the 3P3R DFT design example on a Intel Stratix 10 device. The targeted frequency is 491.52 MHz.

Version	DSP Blocks	RAM Blocks	ALUTs	Dedicated Registers	ALMs	F _{MAX} (MHz)
3P3R DFT 1SG280LN3F43E2VG	46	62 M20Ks	4,595	9,445	4,383	491.52

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

**ISO
9001:2008
Registered**



Document Revision History for AN 464: DFT/IDFT Reference Design

Version	Changes
2018.05.30	Added support for Intel Stratix 10 devices.
2007.06.01	Initial release.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2008
Registered