FFT IP

User Guide

(UG052011, V1.4)

(24.10.2023)

Shenzhen Pango Microsystems Co., Ltd.

Revisions History

Document Revisions

Version	Date of Release	Revisions	Applicable IP and Corresponding Versions
V1.4	24.10.2023	Initial release.	V1.4

IP Revisions

IP Version	Date of Release	Revisions
V1.4	24.10.2023	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
DIF	Decimation-In-Frequency
DIT	Decimation-In-Time
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
IPC	IP Compiler
PDS	Pango Design Suite
SDF	Single-path Delay Feedback

Related Documentation

The following documentation is related to this manual:

1. ***Pango_Design_Suite_Quick_Start_Tutorial***
2. ***Pango_Design_Suite_User_Guide***
3. ***IP_Compiler_User_Guide***
4. ***Simulation_User_Guide***
5. ***User_Constraint_Editor_User_Guide***
6. ***Physical_Constraint_Editor_User_Guide***
7. ***Route_Constraint_Editor_User_Guide***

Table of Contents

Tables

Figures

Chapter 1 Preface

This chapter describes the scope, structure, and writing standards of this manual to help users quickly find the information they need.

1.1 Introduction of the Manual

This manual serves as a user guide for the FFT (Fast Fourier Transform) IP launched by Pango Microsystems, primarily including the IP user guide and related appendices. This manual helps users quickly understand the features and usage of FFT IP.

1.2 Writing Standards of the Manual

Table 1-1 Description of Writing Standards

Text	**Rules**
Attention	If users ignore the attention contents, they may suffer adverse consequences or fail to operate successfully due to incorrect actions.
Description	Instructions and tips provided for users.
Recommendation	Recommended settings and instructions for users.

Chapter 2 IP User Guide

This chapter provides a guide on the use of FFT IP, including an introduction to IP, IP block diagram, IP generation process, Example Design, IP interface description, IP register description, typical applications, instructions and considerations, and IP debugging methods. More details on the design process can be found in the following PDS help documentation.

- "***Pango_Design_Suite_Quick_Start_Tutorial***"
- "***Pango_Design_Suite_User_Guide***"
- "***IP_Compiler_User_Guide***"
- "***Simulation_User_Guide***"

2.1 IP Introduction

FFT IP is launched by Pango Microsystems to implement Cooley-Tukey FFT algorithm and highly effective calculation of DFT (Discrete Fourier Transform). Users can configure and generate the IP module using the IPC (IP Compiler) tool within the PDS (Pango Design Suite).

2.1.1 Key Features[1]

- Single channel;
- Configurable conversion length, with the range of $N=2^m$ (m=3-16);
- Configurable data input width, with the range of b_x=8-34;
- Configurable phase factor precision, with the range of b_w=8-34;
- Configurable scaling type[2]: Unscaled, Block Floating Point;

1 The actual configuration range depends on the device resources and configuration parameter combinations.

2 Full-precision fixed point without compression (Unscaled) and Block Floating Point.

- Configurable data truncation method after butterfly computation[3]: Convergent Rounding, Truncation;
- Configurable data output order[4]: Natural Order, Bit Reversed;
- Configurable implementation architecture: Pipeline, Radix-2 Burst;
- Supports FFT and IFFT, with dynamically configurable selections;
- Supports fixed-point data, represented in binary complement;
- Provides bit-accurate C code verification model.

2.1.2 Applicable Devices and Packages

Table 2-1 Applicable Devices and Packages for FFT IP

Applicable Devices	**Supported Packages**
Logos2 Family	ALL
Titan2 Family	ALL
Kosmo2 Family	ALL

3 Convergent Rounding, and Truncation.

4 Natural Order and Bit Reversed.

2.2 IP Block Diagram

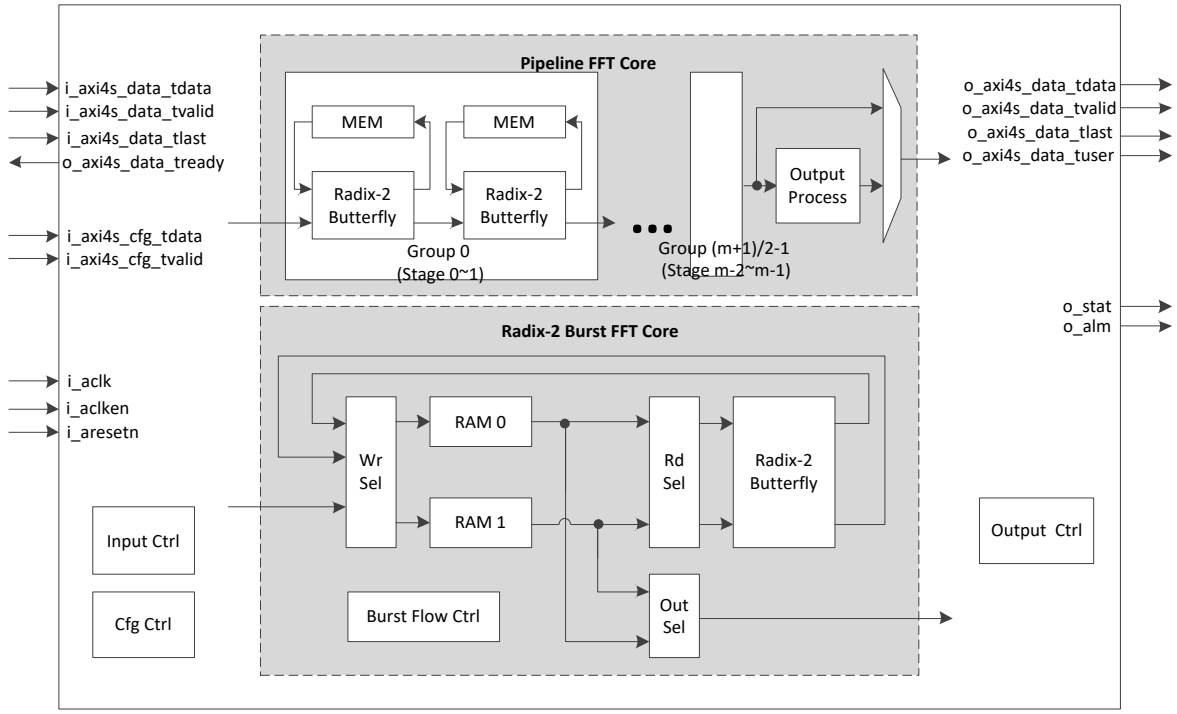


Figure 2-1 FFT IP System Block Diagram

The FFT IP system block diagram is shown in Figure 2-1, composed of Pipeline FFT Core, Radix-2 Burst Core, Input Ctrl, Cfg Ctrl, and Output Ctrl. Among them, the two FFT Cores can be selected through configuring parameters during instantiation, with data input only supporting fixed-point data, and data output supporting Natural Order or Bit Reversed.

Attention:

The input data of the FFT Core must be a complete frame, and the length of each data frame equals the FFT conversion length.

2.2.1.1 Pipeline FFT Core

The Pipeline FFT Core uses a Radix-2^2 SDF (Single-path Delay Feedback) architecture, performing butterfly computations by Decimation-In-Frequency (DIF). Data is input in frame format in natural order, with the length of each data frame equal to the FFT conversion length. Data frames can be input and processed continuously. At the final stage of the butterfly computation, the converted data

frame is output in bit-reversed order. If the data output order is selected as Natural Order, or the scaling type is selected as Block Floating Point, additional storage resources and time are required.

2.2.1.2 Radix-2 Burst Core

The Radix-2 Burst Core uses a Radix-2 iterative architecture, performing butterfly computations by Decimation-In-Time (DIT). Similar to the Pipeline architecture, data is also input in frame format in natural order, with each data frame length equal to the FFT conversion length. Since the FFT Core has only one butterfly computation unit and each stage of iterative calculation of the FFT is completed on this butterfly computation unit, the data of the next iteration cannot be calculated until the data calculation of the previous iteration is completed, thus not supporting continuous input and processing of data frames. Therefore, the Radix-2 iterative architecture requires fewer resources than the Pipeline structure but takes longer to perform the conversion. The Radix-2 iterative architecture has 2 RAMs for storing FFT calculation data. Thus, regardless of whether the data output order is chosen as Natural Order or Bit Reversed, and whether the scaling type is chosen as Unscaled or Block Floating Point, no additional storage resources and time are required.

2.2.1.3 Input Ctrl

The Input Control Submodule receives each data frame input according to the AXI4 Stream bus protocol, parses out real and imaginary data, and sends it to the FFT Core. At the same time, it monitors external input signals, determines if they conform to the AXI4 Stream definitions, and outputs corresponding alarms.

2.2.1.4 Cfg Ctrl

The Configuration Control Submodule receives external dynamic configurations for the FFT Core.

2.2.1.5 Output Ctrl

The Output Control Submodule combines the real and imaginary parts of the calculation result of each data frame and outputs data according to the AXI4 Stream bus protocol. At the same time, it outputs data numbering and the exponent value of the block floating point.

2.3 IP Generation Process

2.3.1 Module Instantiation

Users can customise the configuration of FFT IP through the IPC tool to instantiate the required IP module For detailed instructions on using the IPC tool, please refer to "***IP_Compiler_User_Guide***".

The main steps for the instantiation of the FFT IP module are described as follows.

2.3.1.1 Selecting IP

Open IPC and click File > Update in the main window to open the Update IP dialog box, where you add the corresponding version of the IP model.

After selecting the device type, the Catalog interface displays the loaded IP models. Select the corresponding FFT under the "System" directory; the IP selection path is shown in Figure 2-2. Then set the Pathname and Instance Name on the right side of the page. The project instantiation interface is shown in Figure 2-3.

Attention:

PG2L25H, PG2L50H, PG2L100H, PG2L200H, PG2T390H: The software version must be 2022.1-SP3 or above.

PG2L100HX, PG2T390HX: The software version must be 2023.1 or above.

PG2K400: For software versions, please contact FAE.

For devices uncovered above, please consult the FAE for software version requirements.

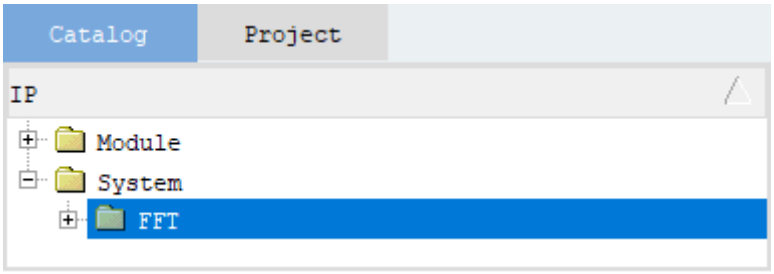


Figure 2-2 FFT IP Selection Path

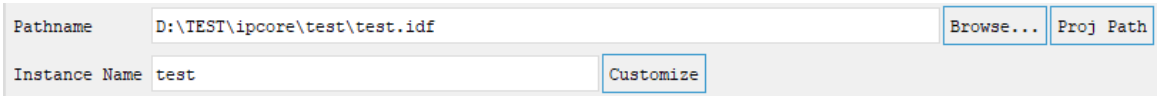


Figure 2-3 Project Instantiation Interface

2.3.1.2 IP Parameter Configuration

After selecting the IP, click <Customize> to enter the FFT IP parameter configuration interface. The left Symbol is the interface block diagram, as shown in Figure 2-4; the Parameter Configuration window is shown on the right side, as shown in Figure 2-5. For configuration parameter descriptions, please refer to Table 2-2.

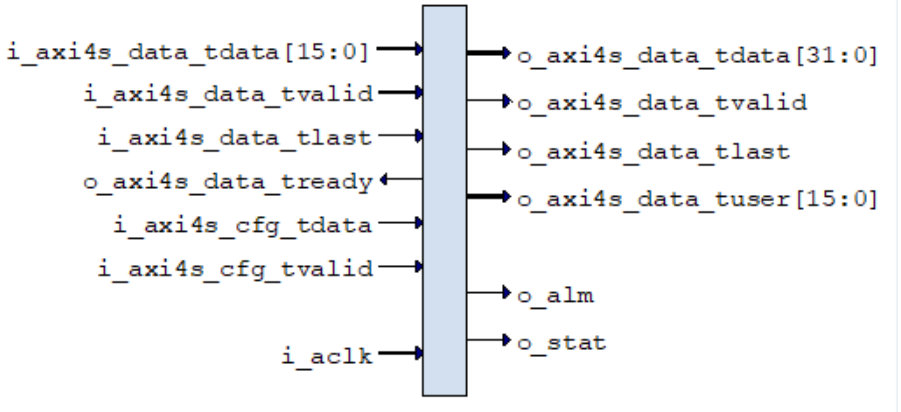


Figure 2-4 FFT IP Interface Block Diagram

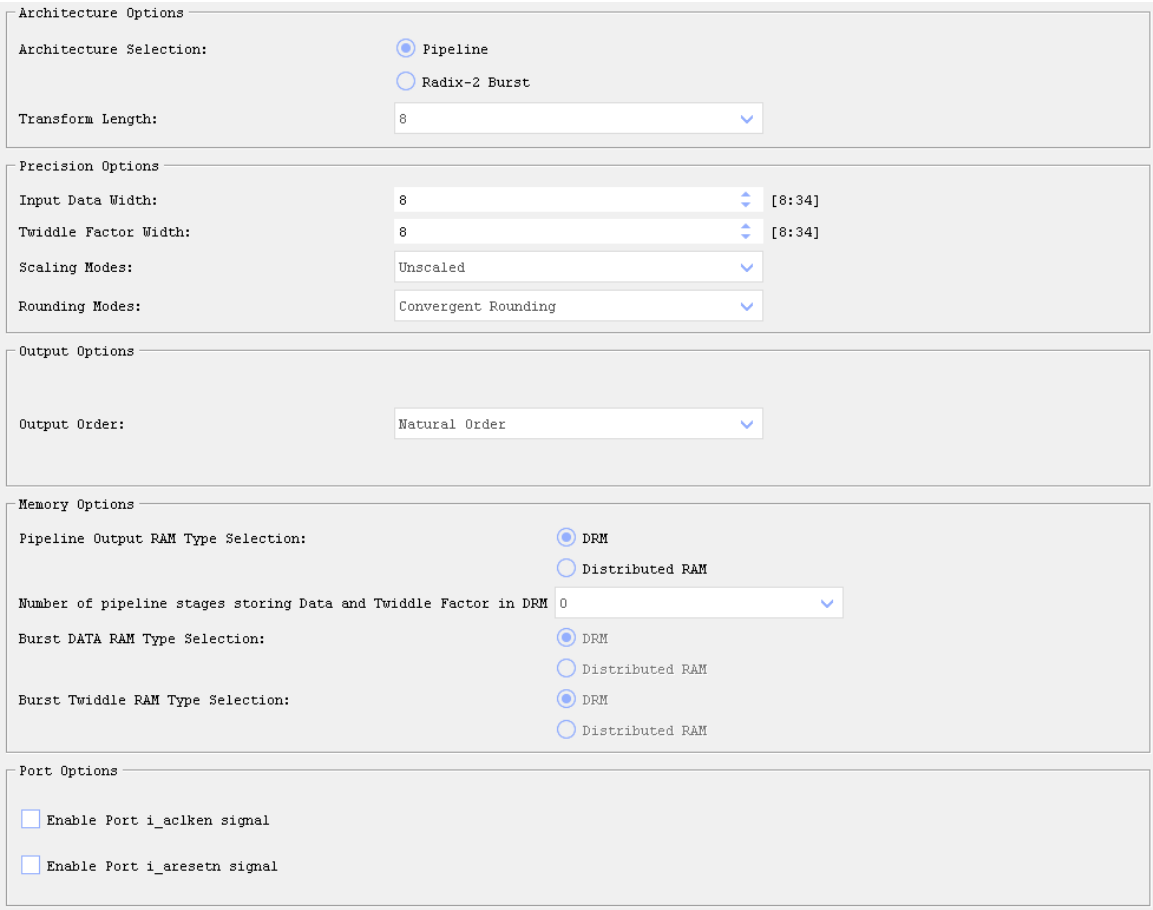


Figure 2-5 FFT IP Parameter Configuration Interface

Table 2-2 Configuration Parameter Description

Option Domain	Parameter/Configuration Options	Parameter Description	Default Value
Architecture Options	Architecture Selection	FFT Core architecture selection: Pipeline Radix-2 Burst	Pipeline
	Transform Length	FFT conversion length configuration: 8 1024 16 2048 32 4096 64 8192 128 16384 256 32768 512 65536	8

Option Domain	Parameter/Configuration Options	Parameter Description	Default Value
Precision Options	Input Data Width	FFT input data width configuration: 8-34	8
	Twiddle Factor Width	FFT twiddle factor width configuration: 8-34	8
	Scaling Modes	FFT calculation scaling type selection: Unscaled Block Floating Point	Unscaled
	Rounding Modes	FFT Truncation method selection after butterfly computation: Convergent Rounding Truncation	Convergent Rounding
Output Options	Output Order	FFT data output order selection: Natural Order Bit Reversed	Nature Order
Memory Options[5]	Pipeline Output RAM Type Selection	Pipeline Core data output memory type selection: DRM Distributed RAM	DRM
	Number of pipeline stages storing Data and Twiddle Factor in DRM	Number of Stages for Pipeline Core storing data and twiddle factor in DRM	0
	Burst DATA RAM Type Selection	Radix-2 Burst Core data memory type selection: DRM Distributed RAM	DRM
	Burst Twiddle RAM Type Selection	Radix-2 Burst Core twiddle factor memory type selection: DRM Distributed RAM	DRM
Port Options	Enable Port i_aclken signal	Configures whether to enable the clock enable signal (i_aclken). Selected: Enable; Cleared: Disable.	Cleared
	Enable Port i_aresetn signal	Configures whether to enable the asynchronous reset signal (i_aresetn). Selected: Enable; Cleared: Disable.	Cleared

2.3.1.3 Generating IP

Upon completion of parameter configuration, click the <Generate> button in the top left corner to generate the FFT IP code according to the user-specific settings. The information report interface for IP generation is shown in Figure 2-6.

5 Refer to the parameter configuration descriptions specified in "2.8.3 Data and Twiddle Factor Storage".

Figure 2-6 FFT IP Generation Report Interface

Upon successful IP generation, the files indicated in Figure 2-3 will be output to the Project path specified in the table below.

Table 2-3 Output Files After FFT IP Generation

Output File[6]	**Description**
$instname.v	The top-level .v file of the generated IP.
$instname.idf	The Configuration file of the generated IP.
/rtl/*	The RTL code file of the generated IP.
/example_design/*	The Test Bench, top-level files, and some module files used in the Example Design of the generated IP.
/pnr/*	The project files and pin constraint files for the Example Design of the generated IP.
/sim/modelsim/*	Generated ModelSim simulation scripts, filelist files, and do files for the IP.
/sim/cmodel/*	Generated C model dynamic link library, header files, and reference programs for the IP.
/sim_lib/*	The directory of the encryption files for the IP.
/rev_1	The default output path for synthesis reports. (This folder is generated only after specifying the synthesis tool)
readme.txt	The readme file describes the structure of the generation directory after the IP is generated.

2.3.2 Constraint Configuration

For the specific configuration method of constraint files, please refer to the relevant help documents in the PDS installation path: "***User_Constraint_Editor_User_Guide***", "***Physical_Constraint_Editor_User_Guide***", "***Route_Constraint_Editor_User_Guide***".

6 <$instname> is the instantiation name entered by the user; "*" is a wildcard character used to replace files of the same type.

2.3.3 Simulation Runs

The simulation of FFT IP is based on the Test Bench of Example Design. For detailed information about Example Design, please refer to "2.4 Example Design".

For more details about the PDS simulation functions and third-party simulation tools, please consult the related help documents in the PDS installation path: "***Pango_Design_Suite_User Guide***", "***Simulation_User_Guide***".

In the Windows system, after IP generation, double-click the *.bat file under the <project_path>/sim/modelsim directory to run the simulation using modelsim10.1a. Upon simulation completion, the specific results will be saved in the vsim.log file, with the ModelSim simulation waveform shown as in Figure 2-7.

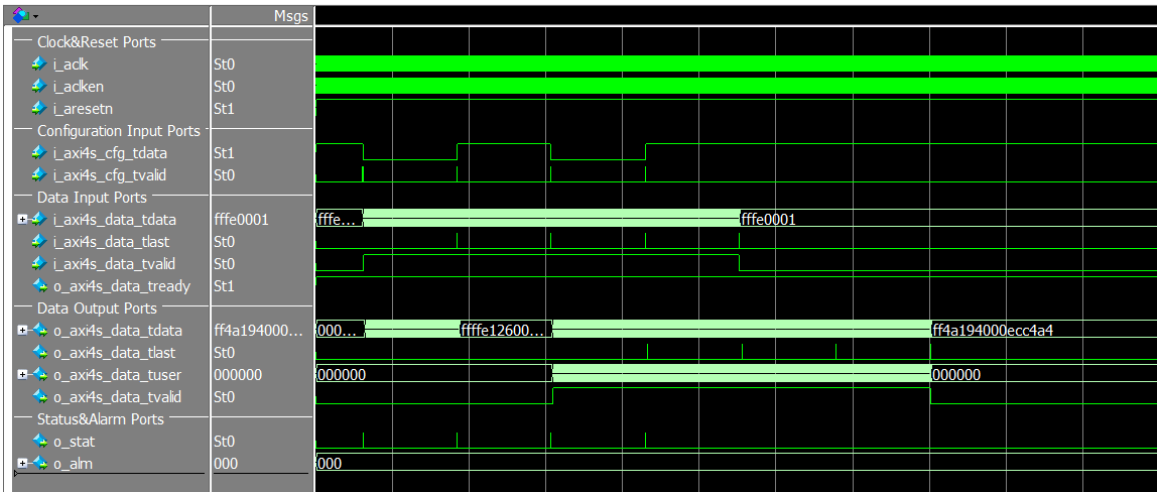


Figure 2-7 Modelsim Simulation Waveform

2.3.4 Synthesis and Placement/Routing

The specific usage of PDS synthesis tools and placement/routing tools can be found in the help documents within the PDS installation path.

Attention:

Example Design project files .pds and pin constraint files .fdc generated with the IP are located in the "/pnr/example_design" directory, and physical constraints need to be modified according to the actual devices and PCB trace routing. For details, please refer to "2.8 Descriptions and Considerations".

2.3.5 Resources Utilization

Table 2-4 Typical Resource Utilization Values for FFT IP Based on Applicable Devices

Device	Configuration Mode													Typical Resource Utilization Values				
	Architecture	Transform Length	Input Data Width	Twiddle Factor Width	Scaling Modes	Rounding Modes	Output Order	Pipeline Output RAM Type	Number of pipeline stages[7]	Burst DATA RAM Type	Burst Twiddle RAM Type	Enable Port i_aclken	Enable Port i_aresetn	LUT	FF	DRM	APM	USCM
PG2T390H/ PG2T390HX	Pipeline	3	8	8	BFP[8]	Truncation	Bit Reversed	Dist RAM[9]	0	-	-	√	×	264	636	0	6	1
	Pipeline	10	28	20	BFP	Truncation	Bit Reversed	Dist RAM	5	-	-	√	×	4432	6544	7	58	1
	Pipeline	11	33	34	Unscaled	Covergent Rounding	Nature Order	-	4	-	-	×	×	4756	10734	13	106	1
	Pipeline	12	13	13	Unscaled	Covergent Rounding	Nature Order	-	3	-	-	√	√	2657	5536	10	36	1
	Pipeline	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	6	-	-	√	×	3888	8751	34.5	92	1
	Radix-2 Burst	3	8	8	Unscaled	Truncation	Bit Reversed	-	-	DRM	DRM	√	√	222	403	1.5	5	1
	Radix-2 Burst	10	20	20	BFP	Truncation	Bit Reversed	-	-	DRM	Dist RAM	√	×	991	1298	2	8	1
	Radix-2 Burst	11	32	8	BFP	Truncation	Nature Order	-	-	-	-	×	×	1101	1720	4.5	10	1
	Radix-2 Burst	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	-	-	-	√	×	1239	1960	18.5	14	1
PG2L25H	Pipeline	12	13	13	Unscaled	Covergent Rounding	Nature Order	-	3	-	-	√	√	2691	5536	10	36	1
	Radix-2 Burst	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	-	-	-	√	×	1239	1960	18.5	14	1

7 Number of pipeline stages: Number of pipeline stages storing Data and Twiddle Factor in DRM

8 BFP: Block Floating Point

9 Dist RAM: Distributed RAM

Device	Configuration Mode													Typical Resource Utilization Values				
	Architecture	Transform Length	Input Data Width	Twiddle Factor Width	Scaling Modes	Rounding Modes	Output Order	Pipeline Output RAM Type	Number of pipeline stages[7]	Burst DATA RAM Type	Burst Twiddle RAM Type	Enable Port i_aclken	Enable Port i_aresetn	LUT	FF	DRM	APM	USCM
PG2L50H	Pipeline	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	6	-	-	√	×	3885	8751	34.5	92	1
	Radix-2 Burst	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	-	-	-	√	×	1237	1960	18.5	14	1
PG2L100H/ PG2L100HX	Pipeline	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	6	-	-	√	×	3885	8751	34.5	92	1
	Radix-2 Burst	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	-	-	-	√	×	1239	1960	18.5	14	1
PG2L200H	Pipeline	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	6	-	-	√	×	3892	8751	34.5	92	1
	Radix-2 Burst	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	-	-	-	√	×	1240	1960	18.5	14	1
PG2K400	Pipeline	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	6	-	-	√	×	3891	8751	34.5	92	1
	Radix-2 Burst	13	24	24	Unscaled	Covergent Rounding	Nature Order	-	-	-	-	√	×	1237	1960	18.5	14	1

Note: "-" indicates that the parameter is not configurable.

Note:

For devices not listed in the above table, please refer to the typical resource utilization values in the table of the devices with similar configurations. Whether this configuration is supported depends on the overall resources of the device.

2.4 Example Design

This section mainly introduces the Example Design scheme based on FFT IP. This scheme generates random data frames, computes them via the FFT IP module, and compares the results with the expected values. The correctness of the computation is judged based on the output error indication.

2.4.1 Design Block Diagram

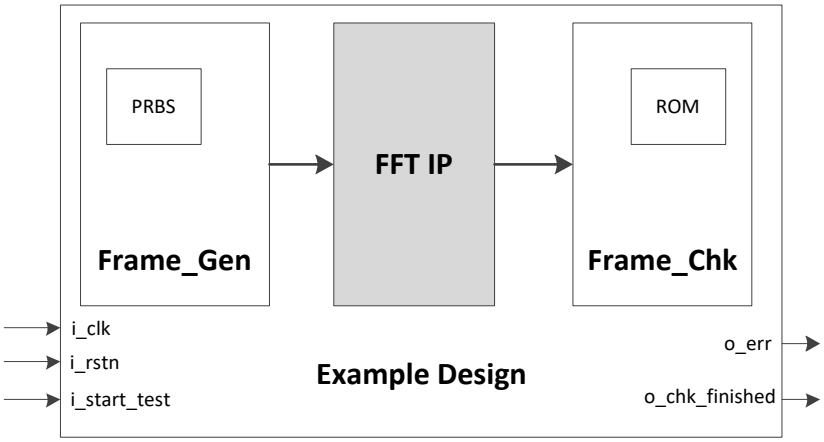


Figure 2-8 FFT IP Example Design Block Diagram

The Example Design integrates FFT IP, Frame_Gen, and Frame_Chk modules at the top level, with the system block diagram shown as in Figure 2-8. The on-board scheme uses the P05I330RD05_A0_PCB single board; users need to modify constraints based on actual conditions during use.

2.4.2 Descriptions of Ports

Table 2-5 Example Design Interface List

Port	I/O	Bit width	Description	Pin constraints
i_clk	I	1	External input single-ended clock, with a maximum frequency of 150MHz	AB27
i_rstn	I	1	External input system reset (active-low)	AB30
i_start_test	I	1	Start external input test (active on the rising edge)	AD28

Port	I/O	Bit width	Description	Pin constraints
o_err	O	1	Test result error indication: 1: Error in test results 0: No error in test results The initial value after reset is 1, facilitating comparison with the first correct computation indication of 0	Y26
o_chk_finished	O	1	Test completion indication: 1: Test completed 0: Test not completed The initial value after reset is 1	Y25

2.4.3 Module Description

2.4.3.1 Frame_Gen

The data frame generation module, which upon initiation of the i_start_test signal, defaults to sending 4 frames of random data, with each frame being identical. Before sending each frame of data, it dynamically configures the computation type of the FFT Core, alternating between FFT and IFFT frame by frame.

2.4.3.2 FFT IP

Instantiated FFT IP module.

2.4.3.3 Frame_Chk

The data frame check module which only checks the number of output frames and the data quantity per frame without checking the computed data values and also monitoring alarms on the FFT IP output. If an error in the number of output frames or data quantity per frame is detected, or an alarm on the FFT IP output is monitored, then the o_err signal will be pulled high and maintained. After the detection period ends, the o_chk_finished signal will be pulled high.

2.4.4 Test Method

After the Example Design bitstream is loaded, the user starts the FFT computation test by first pulling the control input signal i_start_test low and then pulling it high. When the o_chk_finished signal is pulled high, if the o_err signal is low, the test result is correct; otherwise, the test result is incorrect. If the test needs to be repeated, the above process should be repeated.

Attention:

When o_chk_finished is low, operations on i_start_test are invalid.

2.4.5 Instance Configuration

Example Design supports all IP configurations.

2.4.6 Instance Simulation

Under the Windows operating system, after the IP is generated, simulation can be run by double-clicking the "*.bat file[10]" in the <project_path>/sim/modelsim directory.

2.5 Descriptions of IP Interfaces

This section provides descriptions of interfaces related to FFT IP and timings.

10 For the output files after IP generation, please refer to Table 2-3.

2.5.1 Descriptions of Ports

2.5.1.1 Clock and Reset Interface

Table 2-6 List of Clock and Reset Interface Signals

Port	I/O	Bit width	Description
i_aclk	I	1	External clock input, as the sole working clock for the entire IP.
i_aclken	I	1	Optional clock enable input, active high
i_aresetn	I	1	Optional asynchronous reset input, active low, and synchronised to the i_aclk clock domain.

2.5.1.2 Data Input Interface

Table 2-7 List of Data Input Interface Signals

Port	I/O	Bit width	Description
i_axi4s_data_tdata	I	16/32/64/72	Input data. For format definition, please refer to Figure 2-13.
i_axi4s_data_tvalid	I	1	Valid data input indicator. Data transfer begins in the first clock cycle after the rising edge. It should remain at 1'b1 until the last data transfer is complete.
i_axi4s_data_tlast	I	1	Last data input indicator. 1'b1 indicates that the current input data is the last data, i.e., X_n[N-1].
o_axi4s_data_tready	O	1	Allow data input indicator. 1'b1 indicates that data input is currently allowed. For the Pipeline architecture, this signal remains 1'b1, which means external data input is always allowed.

2.5.1.3 Configuration Input Interface

Table 2-8 List of Configuration Input Interface Signals

Port	I/O	Bit width	Description
i_axi4s_cfg_tdata	I	1	Dynamic configuration information. bit0: FFT conversion type 1'b1: IP operates in FFT mode; 1'b0: IP operates in IFFT mode. When i_axi4s_cfg_tvalid is 1'b1, i_axi4s_cfg_tdata is written into the FFT IP.
i_axi4s_cfg_tvalid	I	1	Valid dynamic configuration indicator. 1'b1 indicates valid configuration.

Note:

The IP supports selecting dynamic configuration between FFT and IFFT. Regardless of whether an external reset signal is configured, the initial default value of the FFT conversion type after power-up is FFT type. If an external reset signal is configured, the FFT conversion type after reset is also FFT type. Users can dynamically configure the FFT conversion type through the configuration input interface.

2.5.1.4 Data Output Interface

Table 2-9 List of Data Output Interface Signals

Port	I/O	Bit width	Description
o_axi4s_data_tdata	O	16/32/64/72/80 /88/96/104/112	Output data. For format definition, please refer to Figure 2-14.
o_axi4s_data_tvalid	O	1	Valid data output indicator. Data transfer begins in the first clock cycle after the rising edge. It remains 1'b1 until the last data transfer is complete.
o_axi4s_data_tlast	O	1	Last data output indicator. 1'b1 indicates that the current output data is the last data, i.e., $X_k[N-1]$.
o_axi4s_data_tuser	O	16/24	Output information. For format definition, please refer to Figure 2-15.

2.5.1.5 State Alarm Output Interface

Table 2-10 List of State Alarm Output Interface Signals

Port	I/O	Bit width	Description
o_stat	O	1	bit0: datain_frame_started
o_alm	O	3	bit2: datain_channel_halt bit1: datain_tlast_unexpected bit0: datain_tlast_missing

Note: For the definition of state/alarms, please refer to Table 2-11.

2.5.2 Descriptions of Timings

2.5.2.1 Data Input Interface Timing

FFT/IFFT calculations require the input data to be a complete frame, with the data length of each frame equal to the conversion length. The FFT IP will continuously receive data regardless of the

i_axi4s_data_tvalid signal. Even if the input data is incomplete[11], it will not stop receiving data. Once the IP starts receiving a new frame of data, the internal frame data counter starts counting from 0.

2.5.2.2 Pipeline Architecture

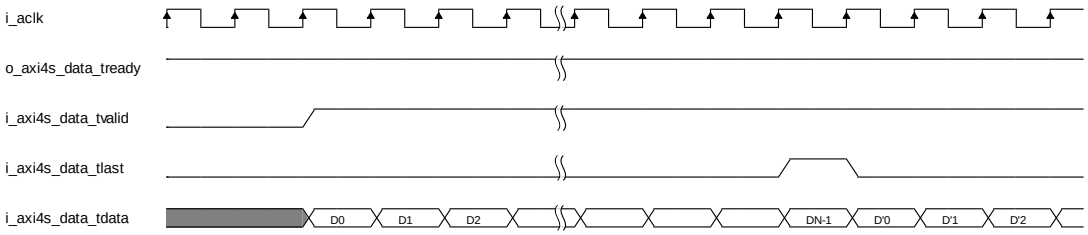


Figure 2-9 Pipeline Architecture Data Input Interface Timing Diagram

2.5.2.3 Radix-2 Burst Architecture

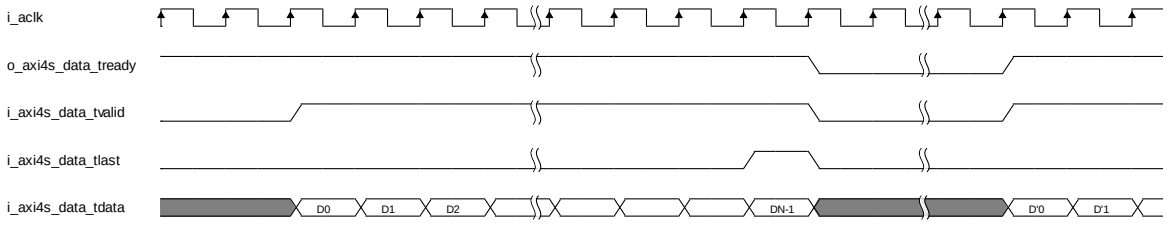


Figure 2-10 Radix-2 Burst Architecture Data Input Interface Timing Diagram

11 For example, if the i_axi4s_data_tvalid signal is pulled down before the current frame data transfer is incomplete, or if it is not maintained at 1'b1 during data transfer.

2.5.2.4 Data Output Interface Timing

2.5.2.4.1 Pipeline Architecture

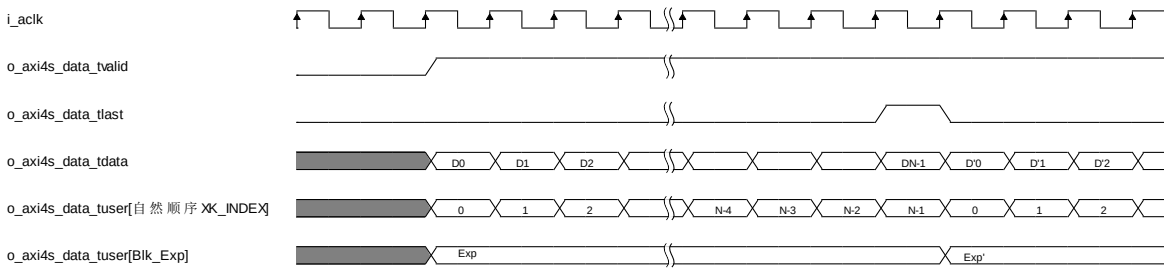


Figure 2-11 Pipeline Architecture Data Output Interface Timing Diagram

2.5.2.4.2 Radix-2 Burst Architecture

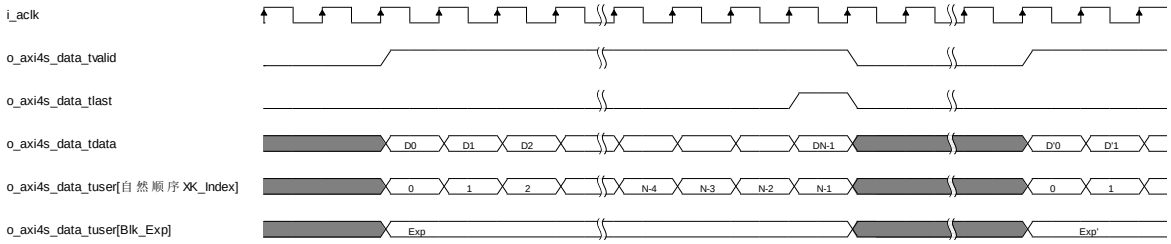


Figure 2-12 Radix-2 Burst Architecture Data Output Interface Timing Diagram

2.5.3 Data/Information Format

2.5.3.1 Input Data Format

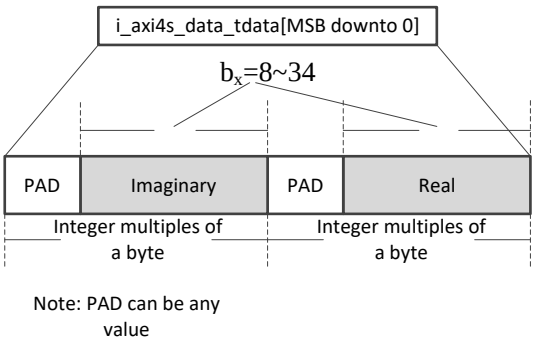
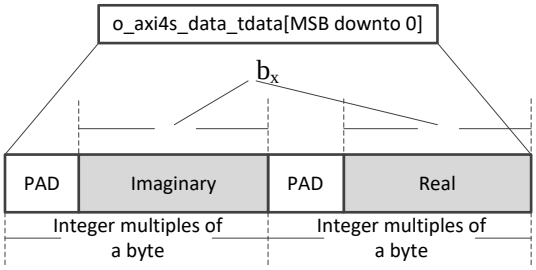


Figure 2-13 Input Data Format Definition

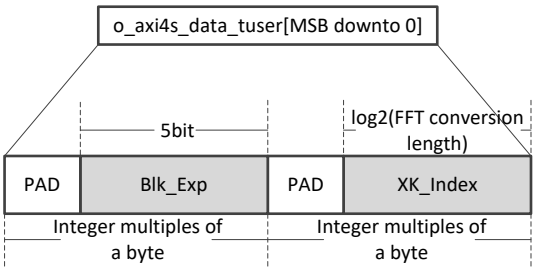
2.5.3.2 Output Data Format



Notes:
(1) PAD is a sign bit extension
(2) Unscaled: bx = input data width + log2(FFT conversion length) + 1
(3) Block Floating Point: bx = input data width

Figure 2-14 Output Data Format Definition

2.5.3.3 Output Information Format



Notes:
(1) PAD is all 0
(2) When the scaling type is not chosen as Block Floating Point, Blk_Exp is all 0

Figure 2-15 Output Information Format Definition

The IP outputs the information as shown in Figure 2-15 via the o_axi4s_data_tuser signal. Among them, Block Floating Point Exponent applies to the Block Floating Point scaling type, which can effectively increase the dynamic computation range without overflow under limited output bit width.

- XK_Index is the data sequence number used to indicate the sequence number of the current transferred data in the data frame, ranging from 0 to N-1;
- Blk_Exp is the Block Floating Point Exponent, used to The FFT IP only supports continuous data input. When this alarm the number of bits to shift the current frame of output data to the

right[12], with an effective bit count of 5. All data in a data frame share the same Block Floating Point Exponent.

2.5.4 State/Alarm Definitions

Table 2-11 State/Alarm Definitions

Item	Categories	Description
datain_frame_started	State	When the FFT Core starts calculating a new data frame, this state pulls high for 1 clock cycle. When this indicator is generated, the IP uses this indicator to latch the most recent configuration input from the configuration port. At the configuration port, users can use this state indicator to determine whether new FFT IP configuration information can be input, ensuring that the previous configuration information is not overridden before it is used.
datain_tlast_missing	Alarm	When the last frame of data is input and the i_axi4s_data_tlast signal does not pull high, this alarm pulls high for 1 clock cycle.
datain_tlast_unexpected	Alarm	When the i_axi4s_data_tlast signal pulls high but the current input frame data is not the last one, this alarm pulls high for 1 clock cycle.
datain_channel_halt	Alarm	When a frame of data is not fully transferred but the i_axi4s_data_tvalid signal pulls low, this alarm pulls high for 1 clock cycle. Note: The FFT IP only supports continuous data input. When this alarm occurs, the internal data processing of IP will not be interrupted, but the calculated output data from the IP will be incorrect.

2.6 Description of the IP Register

None.

2.7 Typical Applications

For typical applications of FFT IP, please refer to "2.4 Example Design".

12 For example, the minimum value of the Block Floating Point Exponent is 0, indicating that the current output frame data has not been right-shifted; when the Block Floating Point Exponent is 10, it indicates that the current output frame data is scaled down by 2^10, which is equivalent to a right shift of 10 bits.

2.8 Descriptions and Considerations

2.8.1 Clock Constraints

Only 1 global clock within the IP requires constraints. Take the Example Design as an example:

create_clock -name {i_clk} [get_ports {i_clk}] -period {6.666} -waveform {0.000 3.333}

2.8.2 IFFT Calculation

For IFFT calculation, the IP does not implement $\frac{1}{N}$, and therefore, the only difference between FFT and IFFT calculations is that the twiddle factors are conjugates of each other.

2.8.3 Data and Twiddle Factor Storage

2.8.3.1 Pipeline FFT Architecture

Pipeline FFT supports configuring the stages of DRM used for storing data and twiddle factors, with "0" indicating that no DRM is used at any stage.

Table 2-12 Pipeline FFT Stage Configuration for DRM Used in Data and Twiddle Factor Storage

FFT Conversion Length	The Range of Stages of Butterfly Computation Units that Use DRM
8	0
16	0
32	0
64	0~1
128	0~2
256	0~3
512	0~4
1024	1~5
2048	2~6
4096	3~7
8192	4~8
16384	5~9
32768	6~10
65536	7~11

If the data output sequence of IP is configured as Natural Order or the truncation method is chosen as Block Floating Point as shown in Figure 2-1, the data which have been pipelined needs to enter the Output Process module for processing, then new data output is generated. The Output Process module contains a memory with a depth of FFT conversion length N. When N is less than or equal to 1024, the memory type supported can be configured as either DRM or distributed RAM; otherwise, the memory type is forced to be DRM.

2.8.3.2 Radix-2 Burst Architecture

When N is less than or equal to 1024, Radix-2 Burst supports configuring the memory type for data and phase factors as either DRM or distributed RAM; otherwise, the memory is forced to be DRM.

The memory types for data and phase factors can be independently configured.

2.8.4 Consideration of Finite Word Length

When implementing FFT conversion using FFT IP, whether choosing the Pipeline FFT Core or the Radix-2 Burst Core, the FFT is always a Radix-2 conversion. For Unscaled type, the word length is increased considering the following two aspects.

The adders and subtractors in each stage of the butterfly unit performing the addition and subtraction of two full precision values will produce a result with a data width increased by 1 bit. Therefore, the data width of the output from each stage of the butterfly unit must be increased by 1, resulting in the final output data width being $\log_2 N$ bit wider than the input data width.

For complex multiplication, the amplitude remains unchanged after calculation (equivalent to a rotation in the complex plane), but theoretically, when the input amplitude is greater than 1, the data width may also increase. For example, the amplitude of the complex number 1+i is 1.414, and if rotated by 45 degrees, the real or imaginary part value would be 1.414. Since the complex multiplication in FFT calculations always involves multiplication by rotation factors, regardless of the number of multiplications, the data width only increases once during the entire FFT processing.

Therefore, for Unscaled type, after the final stage of butterfly addition/subtraction and complex multiplication calculations, the final output data width equals the IP input data width plus $\log_2 N+1$.

2.8.5 Complex Multiplication

The complex multiplication in FFT IP uses a structure with 3 multipliers.

The method for calculating (a+bi)×(c+di) is as follows, using a total of 5 real number additions/subtractions and 3 real number multiplications.

A=(a+b)×c

B=(c+d)×b

C=(b-a)×d

(a+bi)×(c+di) = (A-B)+(B-C)i

2.8.6 C Model

FFT IP provides a bit-accurate C code model, which is provided as a dynamic link library. Additionally, C reference code is provided as an example to help users understand how to call the C model.

Attention:

The data type of the C model is double-precision floating point, and the maximum mantissa width may be less than the requirements for the calculation by FFT IP in certain configurations; therefore, there may be discrepancies between the C model calculations and RTL calculations.

Table 2-13 C Model Files

File	Description
ipsxe_fft_def_vx_x.h	C model header file.
lib_ipsxe_fft_vx_x.so	Dynamic link library used for compiling in Linux system.
lib_ipsxe_fft_vx_x.lib	Dynamic link library used for compiling in Windows system.
ipsxe_fft_wrapper_tb.c	Top-level file for C model simulation.

Note: "vx_x" indicates the version number of the file, such as v1_1.

The usage is as follows:

- On the Linux platform, run the compilation instruction to generate the executable program ipsxe_fft_wrapper_tb.

  gcc -o ipsxe_fft_wrapper_tb ipsxe_fft_wrapper_tb.c ./lib_ipsxe_fft_v1_1.so -lm
- On the Windows platform, run the compilation instruction to generate the executable program ipsxe_fft_wrapper_tb.exe.

  gcc -o ipsxe_fft_wrapper_tb ipsxe_fft_wrapper_tb.c ./lib_ipsxe_fft_v1_1.lib -lm

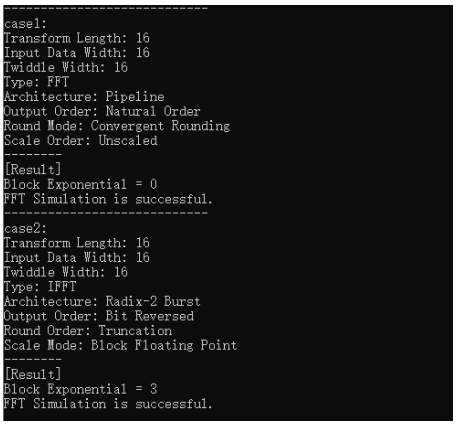


Figure 2-16 C Model Simulation Results

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage license, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its family's products at any time without prior notice.