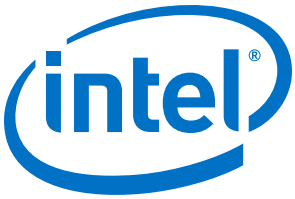


Introduction to DSP Builder for Intel FPGAs

Updated for Intel® Quartus® Prime Design Suite: **18.1**

Subscribe
Send Feedback

HB_DSPB_INTRO | 2018.09.17
Latest document on the web: **PDF** | **HTML**

Contents

1. About DSP Design

1.1. FPGA Architecture Features for DSP Designs

You can configure FPGAs to operate in different modes corresponding to your required functionality. You can use a suitable hardware description language (HDL) such as VHDL or Verilog HDL to implement any hardware design. Thus, the same FPGA can implement a DSL router, a DSL modem, a JPEG encoder, a digital broadcast system, or a backplane switch fabric interface.

High-density FPGAs incorporate embedded silicon features that can implement complete systems inside an FPGA, creating a system on a programmable chip (SOPC) implementation. Embedded silicon features such as embedded memory, DSP blocks, and embedded processors are ideally suited for implementing DSP functions such as finite impulse response (FIR) filters, fast Fourier transforms (FFTs), correlators, equalizers, encoders, and decoders.

The embedded DSP blocks provide functionality such as addition, subtraction, and multiplication, which are common arithmetic operations in DSP functions. Generally, Intel FPGAs offer much more multiplier bandwidth than DSP processors, which only offer a limited number of multipliers.

One determining factor of the overall DSP bandwidth is the multiplier bandwidth, therefore the overall DSP bandwidth of FPGAs can be much higher using FPGAs than with DSP processors.

Many DSP applications use external memory devices to manage large amounts of data processing. The embedded memory in FPGAs meets these requirements and also eliminates the need for external memory devices in some cases.

Embedded processors in FPGAs provide versatile system integration because of flexible partitioning of the system between hardware and software. You can implement the system’s software components in the embedded processors and implement the hardware components in the FPGA's general logic resources. Intel devices provide a choice between embedded soft core processors and embedded hard core processors.

You can implement soft core processors such as the Nios® II embedded processor in FPGAs and add multiple system peripherals. The Nios II processor supports a user-determinable multimaster bus architecture that optimizes the bus bandwidth and removes potential bottlenecks found in DSP processors. You can use multimaster buses to define as many buses and as much performance as needed for a particular application. Off-the-shelf DSP processors make compromises between size and performance when they choose the number of data buses on the chip, potentially limiting performance.

Soft embedded processors in FPGAs provide access to custom instructions such as the MUL instruction in Nios II processors that can perform a multiplication operation in two clock cycles using hardware multipliers. FPGA devices provide a flexible platform to

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 9001:2015 Registered

accelerate performance-critical functions in hardware because of the configurability of the device's logic resources. DSP processors have predefined hardware accelerator blocks, but FPGAs can implement hardware accelerators for each application, allowing the best achievable performance from hardware acceleration. You can implement hardware accelerator blocks with parameterizable IP functions or from scratch using HDL.

Intel offers many IPs for DSP design on FPGAs. You can parameterize Intel DSP IP for the most efficient hardware implementation and to provide maximum flexibility. You can easily port the IP to new FPGA families, leading to higher performance and lower cost. The flexibility of programmable logic and soft IP allows you to quickly adapt your designs to new standards without waiting for long lead times usually associated with DSP processors.

1.2. DSP Design Flow in FPGAs

Traditionally, system engineers use a hardware flow based on an HDL, such as Verilog HDL or VHDL, to implement DSP systems in FPGAs. Intel tools such as DSP Builder, enable you to follow a software-based design flow while targeting FPGAs. DSP Builder for Intel® FPGAs simplifies hardware implementation of DSP functions, provides a system-level verification tool to the system engineer who is not necessarily familiar with HDL design flow, and allows the system engineer to implement DSP functions in FPGAs without learning HDL. DSP Builder for Intel FPGAs provides an interface from Simulink directly to the FPGA hardware. Additionally, you can incorporate the designs created by DSP Builder for Intel FPGAs into a Platform Designer system for a complete DSP system implementation.

1.3. Software and Hardware DSP Design Flows in FPGAs

Intel FPGAs with embedded processors support a software-based design flow. Intel provides the Nios II EDS development tools for compiling, debugging, assembling, and linking software designs. You can then use either on-chip RAM or an external memory device to download these software designs to an FPGA.

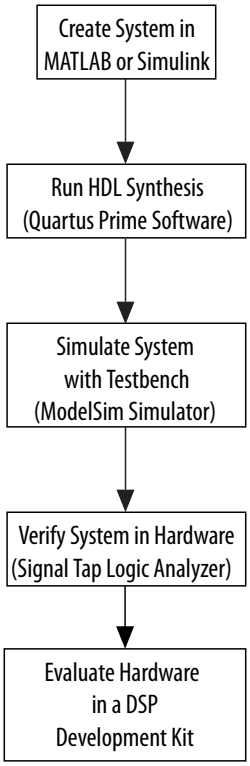
Embedded processors and hardware acceleration offer the flexibility, performance, and cost effectiveness in a development flow that is familiar to software developers. You can combine a software design flow with hardware acceleration. In this flow, you first profile C code and identify the functions that are the most performance critical. Then, you can use Intel's DSP IP or develop your own custom instructions to accelerate those tasks in the FPGA. You can run the system control code with the other nonperformance-critical DSP algorithms on a Nios II embedded processor. Intel also provides system integration tools such as Platform Designer for system-level partitioning and interconnection. You can use Platform Designer to build entire hardware systems by combining the embedded processor, such as a Nios II embedded processor, with other system peripherals and IP.

You can use an HDL-based hardware design flow to develop a pure hardware implementation of a DSP system. Intel provides a complete set of FPGA development tools including the Intel Quartus® Prime and interfaces to other EDA tools such as Synopsys, Synplify, and Precision Synthesis. These tools enable hardware design, simulation, debug, and in-system verification of the DSP system. You can also follow the DSP Builder for Intel FPGAs design flow and implement hardware-only DSP systems in FPGAs without learning HDL.

2. About DSP Builder for Intel FPGAs

DSP Builder for Intel FPGAs shortens DSP design cycles by helping you create the hardware representation of a DSP design in an algorithm-friendly development environment. DSP Builder for Intel FPGAs integrates the algorithm development, simulation, and verification capabilities of MathWorks MATLAB and Simulink system-level design tools with the Intel Quartus Prime software and third-party synthesis and simulation tools. You can combine Simulink blocks with DSP Builder for Intel FPGAs blocks to verify system level specifications and perform simulation.

Figure 1. DSP Builder for Intel FPGAs System-Level Design Flow



The DSP Builder for Intel FPGAs standard blockset is a legacy product. Intel recommends you do not use it for new designs, except as a wrapper for advanced blockset designs.

Related Information

- Volume 2: DSP Builder Standard Blockset in the DSP Builder Handbook
- Volume 3: DSP Builder Advanced Blockset in the DSP Builder Handbook

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 9001:2015 Registered

2.1. Tool Integration

DSP Builder for Intel FPGAs works with Simulink, the ModelSim simulator, and Intel Quartus Prime (including Platform Designer).

Simulink

DSP Builder for Intel FPGAs is interoperable with other Simulink blocksets. In particular, you can use the basic Simulink blockset to create interactive testbenches. The automatic testbenches allows you to compare Simulink simulation results with the output of the ModelSim simulator that simulates the HDL generated for your DSP Builder design.

ModelSim Simulator

You can run the ModelSim simulator from within DSP Builder for Intel FPGAs, if the ModelSim executable is in your path. You can use a script to integrate between the DSP Builder for Intel FPGAs advanced blockset and the ModelSim simulator. The automatic testbench flow runs a test and returns a result indicating whether or not the outputs match.

Intel Quartus Prime

The advanced blockset allows you to build high-speed, high-performance DSP datapaths. In most production designs there is an RTL layer surrounding this datapath to perform interfacing to processors, high speed I/O, memories, and so on. To complete the design, use Platform Designer or RTL to assign board level components. Intel Quartus Prime can then complete the synthesis and place-and-route process. You can automatically load a design into Intel Quartus Prime by clicking on the **Run Quartus Prime** block in the top-level model.

Platform Designer

DSP Builder for Intel FPGAs creates a conduit interface and `hw.tcl` file for each advanced blockset design. It creates a memory-mapped interface only if the design contains interface blocks or external memory blocks. It can also create an Avalon® Streaming interface. The `hw.tcl` file can expose the processor bus for connection in Platform Designer. A DSP Builder for Intel FPGAs advanced blockset subsystem is available from the **System Contents** tab in Platform Designer after you add the path to the `hw.tcl` file to the Platform Designer IP search path

3. Installing and Licensing DSP Builder for Intel FPGAs

3.1. System Requirements

DSP Builder for Intel FPGAs integrates with MathWorks MATLAB and Simulink tools and with the Intel Quartus Prime software.

Ensure at least one version of The MathWorks MATLAB and Simulink tool is available on your workstation before you install DSP Builder for Intel FPGAs. You should use the same version of the Intel Quartus Prime software and DSP Builder for Intel FPGAs. DSP Builder for Intel FPGAs only supports 64-bit versions of MATLAB.

From v18.0, DSP Builder for Intel FPGAs advanced blockset is available for Intel Quartus Prime Pro Edition and Intel Quartus Prime Standard Edition. DSP Builder for Intel FPGAs standard blockset is only available for Intel Quartus Prime Standard Edition.

Table 1. DSP Builder for Intel FPGAs MATLAB Dependencies

Version	MATLAB Supported Versions		
	DSP Builder Standard Blockset	DSP Builder Advanced Blockset	
	Intel Quartus Prime Standard Edition		Intel Quartus Prime Pro Edition
18.1	R2013a	R2013a	R2018a R2017b R2017a R2016b
18.0	R2013a	R2013a	R2017b R2017a R2016b R2016a R2015b
17.1	R2013a	R2013a	R2016a R2015b R2015a R2014b R2014a R2013b

Note: The DSP Builder for Intel FPGAs advanced blockset uses Simulink fixed-point types for all operations and requires licensed versions of Simulink Fixed Point. Intel also recommends the DSP System Toolbox and the Communications System Toolbox, which some design examples use.

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 9001:2015 Registered

Related Information

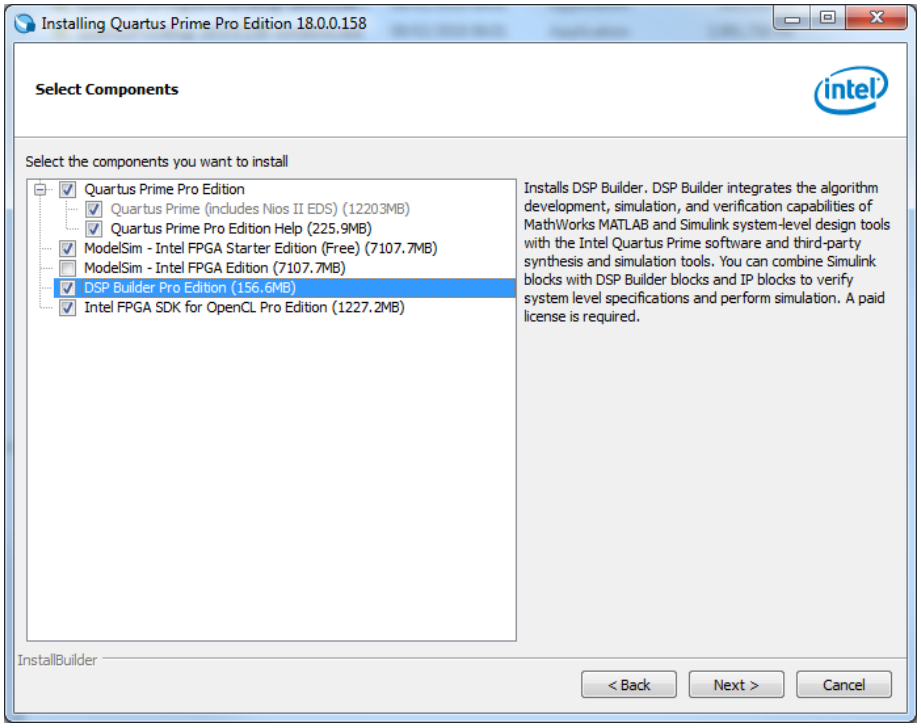
Intel Software Installation and Licensing.

3.2. Installing DSP Builder for Intel FPGAs

1. Install DSP Builder for Intel FPGAs from the Intel Quartus Prime Design Suite.

   In the software installer, ensure you turn on DSP Builder for Intel FPGAs in the **Select components** window.

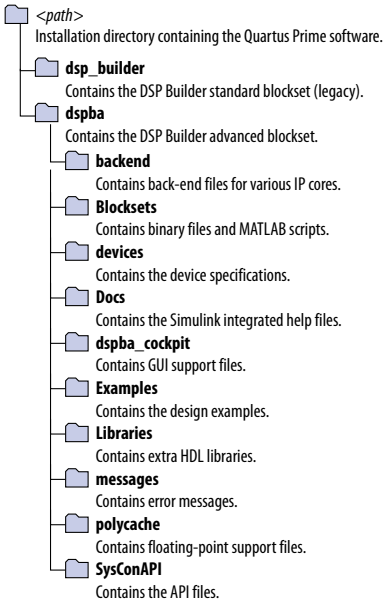
Figure 2. Select Components—DSP Builder



The default installation directory is `c:\intelfpga\<version>\quartus` on Windows or `/opt/intelfpga<version>/quartus` on Linux.

Figure 3. DSP Builder for Intel FPGAs Directory Structure

where <*path*> is the installation directory that contains the Intel Quartus Prime software



After installing DSP Builder for Intel FPGAs, the standard blockset and the advanced blockset libraries are available in the Simulink library browser in the MATLAB software.

3.3. Licensing DSP Builder for Intel FPGAs

Before you can use DSP Builder for Intel FPGAs, request a license file from the Intel website at and install it on your computer.

The Intel Quartus Prime software recommends you specify a path to an `LM_LICENSE_FILE` variable, but it also allows you to use an explicit path to a license file. However, DSP Builder for Intel FPGAs allows you to specify a path to only an `LM_LICENSE_FILE` variable.

Related Information

Intel Software Installation and Licensing.

4. Document Revision History for Introduction to DSP Builder for Intel FPGAs

Version	Software Version	Changes
2018.09.17	18.1	Updated MATLAB version support for DSP Builder v18.1.
2018.05.07	18.0	• Changed name of **FP_FFT** to **FFT_Float** • Updated MATLAB version support for v18.0. • Clarified advanced blockset for Intel Quartus Prime Pro Edition and Intel Quartus Prime Standard Edition • Removed *FPGA-Based DSP Design Flow Options* figure. • Corrected capitalization on *Directory Structure* figure.
2017.05.02	17.0	Updated MATLAB version support for DSP Builder v17.0.
2015.05.01	15.0	Updated MATLAB version support for DSP Builder v15.0.
December 2014	14.1	Updated MATLAB version support for DSP Builder v14.1.
June 2014	14.0	Updated MATLAB version support for DSP Builder v14.0.
November 2013	13.1	Updated MATLAB version support for DSP Builder v13.1.
May 2013	13.0	Updated MATLAB version support for DSP Builder v13.0.
November 2012	12.1	Updated MATLAB version support.
June 2012	12.0	• Updated MATLAB version support • Upgrading from v7.1 chapter • Updated installation instructions • Updated instructions for starting DSP Builder
November 2011	11.1	Updated MATLAB version support.
April 2011	11.0	• Updated MATLAB version support • Added support for 64-bit MATLAB • Updated installation instructions

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 9001:2015 Registered