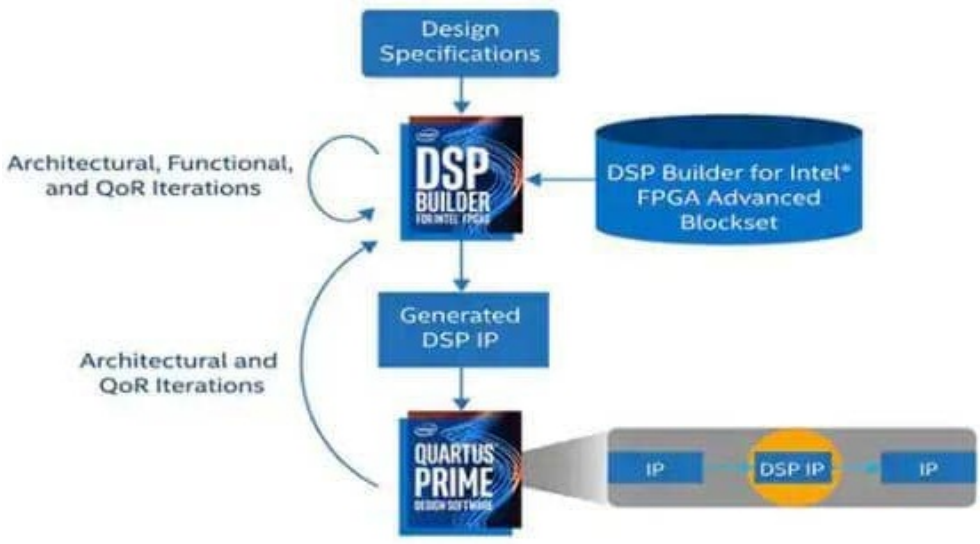
DSP Builder for Intel FPGAs User Guide

Contents

DSP Builder for Intel FPGAs



Product Information

The product is called DSP Builder for Intel FPGAs. It is a software tool that allows users to design and implement digital signal processing (DSP) algorithms on Intel FPGAs. The tool provides a graphical interface that integrates

with The MathWorks MATLAB and Simulink tool, allowing users to design DSP systems using a block diagram approach. The tool has different versions, with the latest version being 22.4. The product has gone through several revisions, with each revision introducing new features, bug fixes, and improvements. The revision history table provides a summary of the changes made in each version. The product has two blockset editions: the standard blockset and the advanced blockset. The standard blockset is available for Intel Quartus Prime Standard Edition, while the advanced blockset is available for both Intel Quartus Prime Pro Edition and Intel Quartus Prime Standard Edition. The product has system requirements that need to be met for proper installation and usage. It requires at least one version of The MathWorks MATLAB and Simulink tool, with support for 64-bit versions of MATLAB. The Intel Quartus Prime software version should match the version of DSP Builder for Intel FPGAs being used. The 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 for additional functionality.

Product Usage Instructions

1. Ensure that you have a compatible version of The MathWorks MATLAB and Simulink tool installed on your workstation. The tool only supports 64-bit versions of MATLAB.
2. Make sure you have the appropriate version of Intel Quartus Prime software installed. The version should match the version of DSP Builder for Intel FPGAs you are using.
3. Launch DSP Builder for Intel FPGAs and open the graphical interface.
4. Design your DSP system using the block diagram approach provided by the tool. Use the available blocks and features to construct your desired algorithm.
5. Take advantage of the Simulink fixed-point types for all operations in your design. Ensure you have the necessary licenses for Simulink Fixed Point.
6. If you require additional functionality, consider using the DSP System Toolbox and the Communications System Toolbox, which are recommended by Intel.
7. Once your design is complete, you can generate the necessary files for programming an Intel FPGA.

By following these usage instructions, you will be able to effectively design and implement DSP algorithms on Intel FPGAs using DSP Builder for Intel FPGAs.

DSP Builder for Intel® FPGAs Release Notes

Related Information

- Knowledge Base
- Software Installation and Licensing

Errata

Errata are functional defects or errors, which may cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.
For full information on errata and the versions affected by errata, refer to the Knowledge Base page of the Intel® website.

Related Information
Knowledge Base

DSP Builder for Intel FPGAs Advanced Blockset Revision History

Version	Date	Description
22.4	2022.12.12	Added Matrix Multiply Engine Design Example.
22.3	2022.09.30	• Improved performance: — DSP Builder now uses the FP DSP block for FP16 and Bfloat16, correctly-rounded, **Add**, **Sub** or **AddSub** on Intel Agilex devices — Provided access to DSP heavy and DSP light architectures for exponential and natural log in the DSP Builder blockset. — improved FP FFT logic usage for two lower-precision FP formats: FP16 and FP19. • Improved integration of DSP Builder designs with other IP in Platform Designer. — DSP Builder does not unroll but keeps together vectors of (optionally) complex signals as a single conduit entity. — You can also assign a custom role to the conduit. DSP Builder automatically assigns multiple conduits with unique names by prefixing the interface with the DSP Builder model name. • Improved the default configuration of the **FFT** blocks to minimize errors when changing the FFT parameters. • Provided option to reset the internal state of the **FIR** block during a warm reset. • Added a library that contains the Simulink blocks that DSP Builder designs support.
22.2	2022.03.30	Reduced internal iteration count in **CORDIC** block to reduce resource usage and increase accuracy.

continued...

Version	Date	Description
22.1	2022.06.30	• Added latency reporting to the **GPIO** block (similar to latency reporting on the **Channel IO** blocks). • Added a hybrid back-to-back **VFFT** block, which supports continuous streaming of data when the FFT size changes without having to flush the FFT pipeline. • Added support for Intel Cyclone 10 LP, Intel MAX 10, Cyclone IV E+GX in DSP Builder Advanced Pro. You must compile the generated RTL with Intel Quartus Std edition. • Extended the read-access control mechanism to **SharedMems** block • Improved DSP block packing by converting **Add**, **Sub**, and **Mux** to a dynamic **AddSub** block
21.4	2021.12.30	Added **AXI4StreamReceiver** and **AXI4StreamTransmitter** to the **Streaming** library

21.3	2021.09.30	• Added DFT Library with **DFT**, **ReorderBlock**, and **ReorderAndRescale** blocks • Added support for Cyclone V devices • Added advisory read access (RA) controls to DSP Builder memory blocks • Added a simplified back-to-back FFT blockset • Added capability to install DSP Builder standalone without requiring a version-compatible Intel Quartus Prime installation
21.1	2021.06.30	• Added **Finite State Machine** block and design example. • Added support for MATLAB version: R2020b
20.1	2020.04.13	Removed device selector in **Device Parameters** panel.
	2019.09.01	Added support for Intel Agilex® devices.
19.1	2019.04.01	• Added support for two new floating-point types float16_m7 (bfloat) and float19_m10. • Added dependent latency feature. • Added FIFO buffer fill-level reporting.
18.1	2018.09.17	• Added HDL import. • Added C++ software models.
18.0	2018.05.08	• Added support for automatic reset minimization of DSP Builder designs. Reset minimization determines the minimal set of registers in a design that require reset, while retaining the design's correct functionality. Reducing the number of registers that DSP Builder resets may give improved quality of results i.e. reduced area and increased Fmax. • Added support for bit fields to the **SharedMem** block. These fields provide analogous functionality to the existing bit field support in the **RegField** and **RegOut** blocks. • Added beta support for HDL import, which incorporates VHDL or Verilog HDL synthesizable designs into a DSP Builder design. You can then cosimulate the imported design with DSP Builder Simulink components. HDL import includes a minimal user interface, but requires some manual setup. To use this feature, you require a license for the MathWorks HDL Verifier tool.
17.1	2017.11.06	• Added super-sample NCO design example. • Added support for Intel Cyclone® 10 and Intel Stratix® 10 devices. • Removed instances of **Signals** block. • Deleted WYSIWYG option on **SynthesisInfo** block.

17.0	2017.05.05	• Rebranded as Intel • Deprecated **Signals** block • Added Gaussian and Random Number Generator design examples • Added variable-size supersampled FFT design example • Added **HybridVFFT** block • Added **GeneralVTwiddle** and **GeneralMultVTwiddle** blocks
16.1	2016.11.10	• Added 4-channel 2-antenna DUC and DDC for LTE reference design • Added BFU_simple block • Created Standard and Pro editions. Pro supports Arria 10 devices; Standard supports all other families. • Deprecated the **Signals** block • Added functionality for setting the Avalon-MM interface settings in the DSP Builder menu
continued...		

Version	Date	Description
16.0	2016.05.02	• Reorganized libraries • Improved folding results on MAX 10 devices • Added new design examples: — Gaussian Random Number Generator — DUC_4C4T4R and DDC_4C4T4R LTE digital-up and down-conversion • Added new FFT pruning strategy: prune_to_widths()

15.1	2015.11.11	• Deprecated **Run Quartus II** and **Run Modelsim** blocks • Added clock crossing support • Added reconfigurable FIR filters • Improved bus interfaces: — Improved error checking and reporting — Improved simulation accuracy — Improved bus slave logic implementation — Improved clock crossing • Changed some Avalon-MM interfaces • Added new blocks: — **Capture Values** — **Fanout** — **Pause** — **Vectorfanout** • Added IIR: full-rate fixed-point and IIR: full-rate floating-point demos • Added transmit and receive modem reference design
15.0	May 2015	• Added support for SystemVerilog output • Added external memories library • Added **External Memory** block • Added new **Allow write on both ports** parameter to **DualMem** block • Changed parameters on **AvalonMMSlaveSettings** block

14.1	December 2014	• Added support for Arria 10 hard-floating-point blocks • Added BusStimulus and BusStimulusFileReader blocks to memory-mapped registers design example. • Added AvalonMMSlaveSettings block and **DSP Builder > Avalon Interfaces > Avalon-MM slave** menu option • Removed bus parameters from Control and Signal blocks • Removed the following design examples: — Color Space Converter (Resource Sharing Folding) — Interpolating FIR Filter with Updating Coefficients — Primitive FIR Filter (Resource Sharing Folding) — Single-Stage IIR Filter (Resource Sharing Folding) — Three-stage IIR Filter (Resource Sharing Folding) • Added system-in-the-loop support • Added new blocks: — Floating-point classifier — Floating-point multiply accumulate — Added hypotenuse function to math block • Added design examples: — Color space converter — Complex FIR — CORDIC from Primitive Blocks — Crest factor reduction — Folding FIR — Variable Integer Rate Decimation Filter — Vector sort – sequential and iterative
continued…		

Version	Date	Description

		• Added reference designs: — Crest factor reduction — Direct RF with Synthesizable Testbench — Dynamic Decimation Filter — Reconfigurable Decimation Filter — Variable Integer Rate Decimation Filter • Removed resource sharing folder • Updated ALU folder
14.0	June 2014	• Added support for MAX 10 FPGAs. • Removed support for Cyclone III and Stratix III devices • Improved **DSP Builder Run ModelSim** option, which now allows you to run ModelSim for the top-level design or individual submodules • Changed the generation of HDL into the device level directory (under the specified target RTL directory) rather than in a hierarchy of directories • Added read signal on bus interface • Added clear port on the FIFO • Deprecated 13 FFT blocks • Added new design examples: — Avalon-ST Interface (Input and Output FIFO Buffer) with Backpressure — Avalon-ST Interface (Output FIFO Buffer) with Backpressure — Fixed-point maths functions — Fractional square root using CORDIC — Normalizer — Parallel FFT — Parallel Floating-Point FFT — Square root using CORDIC — Switchable FFT/iFFT — Variable-Size Fixed-Point FFT — Variable-Size Fixed-Point FFT without BitReverseCoreC Block — Variable-Size Fixed-Point iFFT — Variable-Size Fixed-Point iFFT without BitReverseCoreC Block — Variable-Size Floating-Point FFT — Variable-Size Floating-Point FFT without BitReverseCoreC Block — Variable-Size Floating-Point iFFT

		— Variable-Size Floating-Point iFFT without BitReverseCoreC Block • Added new blocks: — Anchored Delay — Enabled Delay Line — Enabled Feedback Delay — FFT2P, FFT4P, FFT8P, FFT16P, FFT32P, and FFT64P — FFT2X, FFT4X, FFT8X, FFT16X, FFT32X, and FFT64X — FFT2, FFT4, VFFT2, and VFFT4 — General Multitwiddle and General Twiddle (GeneralMultiTwiddle, GeneralTwiddle) — Hybrid FFT (Hybrid_FFT) — Parallel Pipelined FFT (PFFT_Pipe) — Ready
13.1	November 2013	• Removed support for the following devices: — Arria GX — Cyclone II — HardCopy II, HardCopy III, and HardCopy IV — Stratix, Stratix II, Stratix GX, and Stratix II GX • Improved ALU folding flow • Added new functions to Math block.

continued...

Version	Date	Description

		• Added Simulink fi block option to Const, DualMem, and LUT blocks • Added new design examples: — Variable-precision real-time FFT — Interpolating FIR Filter with updating coefficients — Time-delay beamformer • Added new blocks: — Anchored Delay — Polynomial — TwiddleAngle — TwiddleROM and TwiddleROMF — VariableBitReverse — VFFT
13.0	May 2013	• Updated device block with new Device Selector menu. • Added new ModelPrim blocks: — Const Mult — Divide — MinMax — Negate — Scalar Product • Added nine new FFT blocks • Added ten new FFT demonstrations

12.1	November 2012	• Added ALU folding feature • Added enhanced precision floating-point options • Added the following new ModelPrim blocks: — AddSub — AddSubFused — CmpCtrl — Math — Maximum and Minimum — MinMaxCtrl — Round — Trig • Added the following new FFT blocks: — Edge Detect (EdgeDetect) — Pulse Divider (PulseDivider) — Pulse Multiplier (PulseMultiplier) — Bit-Reverse FFT with Natural Output (FFT_BR_Natural) • Added the following new FIR design examples: — Super-sample decimating FIR filter — Super-sample fractional FIR filter • Added the position, speed, and current control for AC motors (with ALU folding) design example

Related Information
DSP Builder Advanced Blockset Handbook

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 2. 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
22.4	Not available		R2022a R2021b R2021a R2020b R2020a
22.3	Not available		R2022a R2021b R2021a R2020b R2020a
22.1	Not available		R2021b R2021a R2020b R2020a R2019b
21.3	Not available		R2021a R2020b R2020a R2019b R2019a
21.1	Not available		R2020b R2020a R2019b R2019a R2018b
20.1	Not available		R2019b R2019a R2018b R2018a R2017b R2017a
19.3	Not available		R2019a R2018b R2018a R2017b

continued…

Version	MATLAB Supported Versions		
	DSP Builder Standard Blockset	DSP Builder Advanced Blockset	
	Intel Quartus Prime Standard Edition		Intel Quartus Prime Pro Edition
			R2017a R2016b
19.1	Not supported	R2013a	R2018b R2018a R2017b R2017a R2016b
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.

Related Information
Intel Software Installation and Licensing.
DSP Builder for Intel® FPGAs Release Notes 9

Documents / Resources

	intel DSP Builder for Intel FPGAs [pdf] User Guide DSP Builder for Intel FPGAs, Builder for Intel FPGAs, Intel FPGAs, FPGAs

References

- intel 1. DSP Builder for Intel® FPGAs Release Notes
- intel 1. Answers to Top FAQs
- intel 2. Introduction to Intel® FPGA Software Installation and Licensing
- intel Intel ISO 9001:2015 Registrations
- intel FPGA Knowledge Base Articles Search