DSP Builder Release Notes

June 2007, Version 7.1 SP1 Release Notes

These release notes for the DSP Builder software, v7.1 SP1 contain the following information:

System Requirements

To use DSP Builder, v7.1 SP1, the following system requirements exist:

- A computer running a Windows XP operating system
- The MathWorks release R14 SP3, R2006a, R2006b, or R2007a

☞ Only 32-bit versions of The MathWorks release are currently supported. Note that DSP Builder does not work with MATLAB in read only mode. If error messages are issued while creating board components during the DSP Builder installation, re-install MATLAB with the READ ONLY option unchecked.

The flow also supports:

- ModelSim® or ModelSim-Altera simulator version 6.1d to 6.1g

DSP Builder is integrated with v7.1 SP1 of the Quartus® II software which must be available on your workstation before you install DSP Builder.

For full system requirements and install instructions, refer to the *Quartus II Installation & Licensing for Windows* manual on the Altera Literature website at: **www.altera.com/literature/lit-qts.jsp**

DSP Builder provides automated design flows using Tool command language (Tcl) scripts as well as a manual design flow.

New Features & Enhancements

The following list outlines new features and enhancements in version 7.1:

- DSP Builder can now only be installed within a matching version of the Quartus II software.
- The DSP Builder blockset has been completely re-written to provide individual controls for optional ports and many new options. New blocks include:
  - Single-Port RAM block
  - True Dual-Port RAM block
  - Bus Splitter block
  - Clock and Clock_Derived blocks (replaces the ClockAltr block)
  - New Signal Compiler and TestBench blocks
  - Stratix III DSP block (supports fine-grain control of multiply add and multiply accumulate operations for Stratix III devices)
- The single-port Avalon-MM blocks have been obsoleted in favour of the multi-port Avalon master and slave blocks.
- The Multi Channel Display, Multi Channel Extract and Nios Custom Instruction blocks have been obsoleted.
- The optional enable port on the Sum of Products block is now consistent with the other blocks and enables (or disables) the entire pipeline.
- Inferred datatype selection for propagation.
- New fast functional simulation support for Video and Image Processing Suite MegaCore functions.
- New HDL generation flow.
- New testbench generation and execution interface (replacing **tbdiff**).
- Enhanced support for multiple clock domains
- Asynchronous clear ports are now implicitly wired.

Errata Fixed in This Release

The following errata are fixed in version 7.1 SP1:

- VCD Sink block was restricted to 10 inputs
- Testbench did not detect absence of ModelSim
- Simulation failed for custom library blocks
- SOPC Builder files were not added to the Quartus II project
- Could not upgrade the Subsystem Builder block
- Problem with VHDL generated for one-bit wide signed gain block

The following errata were fixed in version 7.1:

- Clock information was incorrect in VEC file for multiclock designs
- Signed fractional HIL simulation was incorrect
- VHDL for black boxes was not included in the generated scripts

For existing up-to-date errata, refer to the *DSP Builder, v7.1 Errata Sheet* on the Errata Sheets page of the Altera literature website.

Obtain & Install DSP Builder

Before you can use DSP Builder, you must obtain the files and install them on your computer. DSP Builder can be installed from the Quartus II, 7.1 SP1 DVD, or downloaded from the Altera web site.

☞ The following instructions describe downloading and installing DSP Builder. If you already have installed DSP Builder from the DVD, skip to "DSP Builder Directory Structure" on page 4.

Download DSP Builder

If you have Internet access, you can download DSP Builder from Altera's web site at **www.altera.com.** Follow the instructions below to obtain DSP Builder from the Internet.

If you do not have Internet access, contact your local Altera representative to obtain the Quartus II version 7.1 SP1 DVD.

1. Point your web browser to **www.altera.com/products/software/**

2. Click on **DSP Builder** and then the link to **Download DSP Builder**.

3. Fill out the registration form and click **Submit Request**.

4. Read the Altera license agreement. Turn on the **I have read the license agreement** check box and click **Proceed to Final Step**.

5. Follow the instructions on the download and installation page to download the executable and save it to your hard disk.

Install DSP Builder

☞ The MATLAB, Simulink and Quartus II software must be installed before you install DSP Builder.

Follow these steps to install DSP Builder on a computer running a supported version of the Windows operating system:

1. Close the following software if it is running on your computer:

   - The Quartus II software
   - The MATLAB and Simulink tools
   - The ModelSim simulator

2. Choose **Run** (Windows Start menu).

3. Type <*path*>`\DSPBuilder-v7.1 SP1.exe`, where <*path*> is the location of the downloaded file.

4. Click **OK**. The **DSP Builder v7.1 SP1 - InstallShield Wizard** dialog box appears.

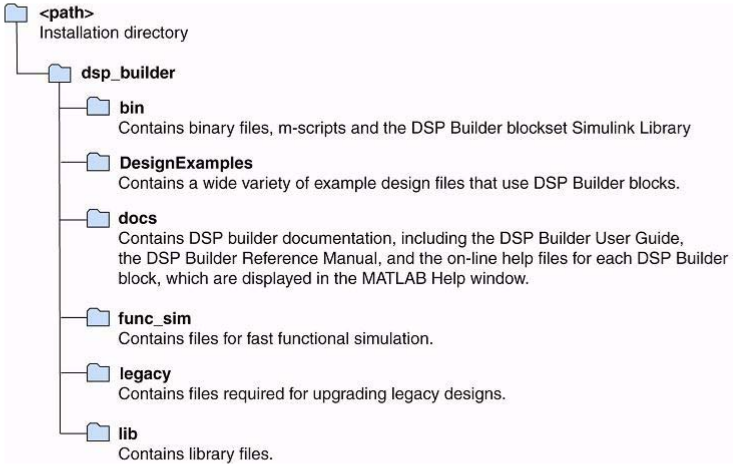
5. Follow the on-line instructions to install a new copy of the product. Note that you are prompted for the locations of the Quartus II and MATLAB software you want to use with DSP Builder.

☞ Do not remove any previous versions of DSP Builder at this stage. The old version is required to run the upgrade program that converts your old designs for use with the new version. The old version can be removed later after you have upgraded your designs.

DSP Builder Directory Structure

The DSP Builder installation program copies files into the directories shown in Figure 1 where *<path>* is the installation directory which contains the Quartus II software. The default installation directory on Windows is **c:\altera\71sp1\quartus**.

Figure 1. DSP Builder Directory Structure



<path>
Installation directory

dsp_builder

bin
Contains binary files, m-scripts and the DSP Builder blockset Simulink Library

DesignExamples
Contains a wide variety of example design files that use DSP Builder blocks.

docs
Contains DSP builder documentation, including the DSP Builder User Guide, the DSP Builder Reference Manual, and the on-line help files for each DSP Builder block, which are displayed in the MATLAB Help window.

func_sim
Contains files for fast functional simulation.

legacy
Contains files required for upgrading legacy designs.

lib
Contains library files.

Accessing the DSP Builder Libraries from within MATLAB

After installing DSP Builder, you can view DSP Builder libraries in the MATLAB software by performing the following steps:

1. Start the MATLAB software. (You can choose to automatically start MATLAB at the end of the install program.)

2. Click the Start icon in the lower left corner of the main MATLAB window. Point to Simulink and click **Library Browser**. The **Altera DSP Builder Blockset** folder is available in the Simulink Library Browser window.

Using Previous Versions of DSP Builder

The DSP Builder install program searches for any previous installations of DSP Builder before it installs a new version. If any existing installations are found, the program prompts you to update an existing installation of the product or install a new copy of the product.

If you choose to update an existing product, you can modify, repair or remove the old version.

☞ Do not attempt to modify or repair a pre-7.1 version of DSP Builder using the v7.1 install program.

A previous version of DSP Builder can co-exist with v7.1 and is required to upgrade your pre-v7.1 designs.

If an older version of DSP Builder is installed, you can register it in MATLAB by entering the following commands at the MATLAB prompt:

1. `cd` *<path to old version>*`\DSPBuilder\AltLib` ↵

2. `setup_dspbuilder` ↵

☞ Note that the old version of the DSP Builder blockset is called **Altera DSP Builder**. The new version is called **Altera DSP Builder Blockset**. Although these blocksets can co-exist while you are upgrading your designs, you cannot mix blocks from both blocksets. Errors are issued during simulation if your v7.1 design includes any pre-v7.1 blocks. The old version of DSP Builder should be uninstalled once you have upgraded your designs.

Using Multiple Versions of MATLAB

You specify the MATLAB installation that you want to use with DSP Builder during DSP Builder installation. If you have more than one MATLAB installation (for example, release R2006a and R2006b) you can register DSP Builder with another version using the following procedure:

1. Open a command prompt and change directory to the DSP Builder installation:

   ```
   cd <DSP Builder Install Path>dsp_builder
   ```

2. Run the following command to register DSP Builder with the required MATLAB installation:

   ```
   setupMatlabClassPath install <MATLAB Install Path>
   <DSP Builder Install Path>\dsp_builder
   ```

☞ You must use quotes if the DSP Builder install path or MATLAB install path include spaces.

Set Up Licensing

Before using DSP Builder, you must request a license file from the Altera web site at **www.altera.com/licensing** and install it on your computer. When you request a license file, Altera e-mails you a **license.dat** file that enables HDL file and Tcl script generation.

If you do not have a DSP Builder license file, you can create models with DSP Builder blocks but you cannot generate HDL files or Tcl scripts.

☞ Before you set up licensing for DSP Builder, you must already have the Quartus II software installed on your computer with licensing set up.

To install your license, you can either append the license to your **license.dat** file or you can specify a separate DSP Builder license file in the Quartus II software.

Appending the License to Your license.dat File

To install your license, perform the following steps:

1. Close the following software if it is running on your computer:

   - The Quartus II software
   - The LeonardoSpectrum software
   - The Synplify software
   - The MATLAB and Simulink tools

- The ModelSim simulator
- The Precision RTL synthesis software

2. Open the DSP Builder license file in a text editor. The file should contain one FEATURE line, spanning two lines.

3. Open your Quartus II **license.dat** file in a text editor.

4. Copy the FEATURE line from the DSP Builder license file and paste it into the Quartus II license file.

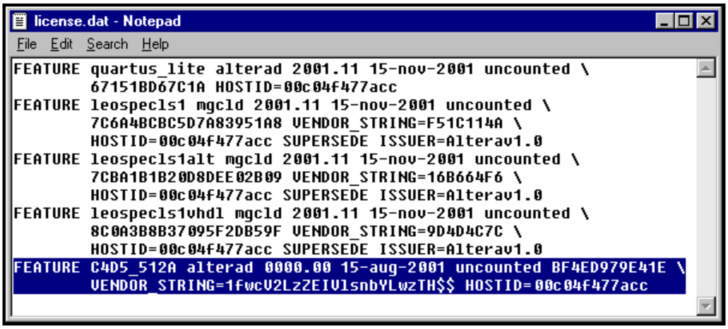
   Do not delete any FEATURE lines from the Quartus II license file.

5. Save the Quartus II license file.

   When using editors such as Microsoft Word or Notepad, ensure that the file does not have any extra extensions appended to it after you save (for example, **license.dat.txt** or **license.dat.doc**). Verify the filename at the system command prompt.

Figure 2–1 shows an example updated **license.dat** file that includes the DSP Builder FEATURE line (highlighted).

Figure 2–1. Example license.dat File



```
FEATURE quartus_lite alterad 2001.11 15-nov-2001 uncounted \
        67151BD67C1A HOSTID=00c04f477acc
FEATURE leospecls1 mgcld 2001.11 15-nov-2001 uncounted \
        7C6A4BCBC5D7A83951A8 VENDOR_STRING=F51C114A \
        HOSTID=00c04f477acc SUPERSEDE ISSUER=Alterav1.0
FEATURE leospecls1alt mgcld 2001.11 15-nov-2001 uncounted \
        7CBA1B1B20D8DEE02B09 VENDOR_STRING=16B664F6 \
        HOSTID=00c04f477acc SUPERSEDE ISSUER=Alterav1.0
FEATURE leospecls1vhdl mgcld 2001.11 15-nov-2001 uncounted \
        8C0A3B8B37095F2DB59F VENDOR_STRING=9D4D4C7C \
        HOSTID=00c04f477acc SUPERSEDE ISSUER=Alterav1.0
FEATURE C4D5_512A alterad 0000.00 15-aug-2001 uncounted BF4ED979E41E \
        VENDOR_STRING=1fwcV2LzZEIVlsnbYLwzTH$$ HOSTID=00c04f477acc
```

Specify the License File Location Using LM_LICENSE_FILE

The LM_LICENSE FILE environment variable must point to the location of the **license.dat** file. The following instructions can be used to set this environment variable for Windows 2000 or Windows XP:

1. Choose **Settings** > **Control Panel** from the Windows Start menu.

2. Double-click the **System** icon in the Control Panel window.

3. Click the **Advanced** tab in the **System Properties** dialog box.

4. Click **Environment Variables**.

5. Click the **System Variable** list to highlight it.

6. If `LM_LICENSE_FILE` already exists, select **Edit**. Otherwise, select **New** and type `LM_LICENSE_FILE`.

7. Type *<path to license file>*`\license.dat` or *<port>*`@`*<hostname>* in the **Value** box.

   If more than one application uses this environment variable, modify the existing `LM_LICENSE_FILE` variable and separate the different paths with a semicolon (;), for example:

   ```
   c:\flexlm\license.dat;1800@myserver
   ```

8. Click **OK**.

Upgrading Your Existing Models

You must have DSP Builder v7.1 and a previous version (v6.1 or v7.0) of DSP Builder registered in MATLAB before you can upgrade your pre-v7.1 design models.

☞ The **Altera DSP Builder Blockset** must be opened in the Simulink library browser before you can upgrade a design.

To upgrade a existing design, perform the following steps:

1. Use the MATLAB browser to navigate into the design directory for the old design.

2. Open the top-level model in the design.

   ☞ If your design consists of custom library components and a separate testbench model, it is best to upgrade the testbench before any custom library components.

3. In the Edit menu, choose **Update Diagram** (or use the Ctrl+D shortcut) to ensure your design is up-to-date and correct the design if any errors or warnings are issued. For example, if there are any warning that a block's sample time is back-inherited theses should be corrected by removing the -1 setting of its sample time parameter.

4. Enter the following command at the MATLAB prompt:

```
alt_upgradeModel <model name> option1 option2 ...
```

Any number of the available options can be used in any order. Table 2 describes the available options.

Table 1. Options to the altUpgradeModel Command

Option	Description
UnlockLibraries	Causes libraries to be recursively unlocked and modified if referenced from the upgraded design. Already-unlocked libraries are always upgraded.
RemoveSubsystemAltBusBlocks	Upgrades the model and also deletes any unnecessary AltBus input blocks connected to subsystem inputs and AltBus blocks connected to subsystem outputs. In general, it is no longer necessary to have AltBus blocks connected to every subsystem input or output and removing them means that bit widths can be propagated into subsystems. However, the bit width specified on an AltBus block may be used to reshape the inputs and outputs from the system and removing these blocks may cause the behavior to change and in some cases may cause unresolvable propagation loops.
Verbose	Displays information about each block as it is upgraded or removed.

5. Save your design and close it (including any libraries that may have changed).

The design is reloaded and all libraries are correctly initialized when it is reopened.

A copy of the model is backed up to a **backup** subdirectory and all blocks in the model updated to use the new v7.1 **Altera DSP Builder Blockset**. If a backup subdirectory already exists, a new subdirectory is created by appending an integer (for example **backup1**).

☞ You cannot revert a design model to a previous version after you have updated it to v7.1. If you want to use an older version of DSP Builder, you should use the version saved in the backup subdirectory.

Limitations of the Upgrade Model Utility

The blocks in your model are upgraded to use the corresponding block in the new v7.1 blockset. However some blocks may be obsolete or require manual intervention to complete the conversion process.

Table 2 lists some of the issues which may require attention.

Table 2. Model Upgrade Issues

Issue	Action
PLL output clocks cannot be named	In previous versions of DSP Builder, it was possible to have a PLL block and multiple ClockAltr blocks which represented PLL outputs. The PLL output clocks took the names of the clock blocks. This feature is not supported in v7.1 and cannot be automatically fixed. In v7.1, the PLL output clocks are named <pll name>_clk<output index>. All source blocks and rate change blocks referencing clock pins must be manually edited to reference these PLL clock output pins.
PLL period multiply and divide values must be integers	In previous versions, the multiply and divide values could have non-integer values and could be specified using MATLAB variables. In v7.1, you must specify the clock period ratio directly as an integer period multiplier and an integer period divider. MATLAB variables cannot be used.
The PLL output clock period is incorrect after upgrade	Occasionally, the PLL parameters are upgraded incorrectly. Open the PLL parameter dialog and enter the correct values for the period multipliers and dividers.
When upgrading a design with a PLL, extra clock blocks are created for each distinct sample time	The extra Clock and Clock_Derived blocks should be removed, and any blocks referencing them manually corrected to reference the PLL-driven clocks. Note that the numbering of these clock pins will not in general match the numbering of the PLL clocks.
The PLL input clock frequency information is lost during the upgrade process	Typically, you may want to create a new Clock block replicating this information, as the base clock pin generated by the upgrade script is unlikely to be the correct driving clock domain. For example, if the PLL specified an input clock frequency of 50 MHz, add a Clock block and configure it to a clock period of 20ns and sample time 20e-9.
Clock blocks do not support rate change divider	In previous versions, the ClockAltr blocks supported a rate change option (Addition Clock Divider) which could be used to generate a slower clock signal. This feature is not supported in v7.1. If you want to generate different frequency clocks internally in 7.1, you must add a PLL block driven from the required input clock.
Error assigning clock for Dual-Clock FIFO block	Under some circumstances - noted by the message “No clock specified for {write/read} port, ...” you may have to manually select clocks after upgrading designs containing the Dual-Clock FIFO block.
Error assigning PLL clock for Multi-Rate DFF block	When upgrading a Multi-Rate DFF block connected to a PLL clock, an error is issued of the form: “Cannot upgrade clock in block foo/Multi-Rate DFF. Original clock source: PLL CLOCK0.” The blocks must be manually corrected to reference the PLL clock.
Unnecessary clock specification for source blocks	In general, source blocks do not need to specify a clock domain, if it can be inferred from the blocks they are driving. However, the upgrade path always specifies a clock if it is not the base clock. Your multi-clock design may be easier to maintain if, after upgrading, you manually turn off **Specify Clock** for source blocks – especially constant, VCC and GND blocks – wherever possible.
Errors issued if a constant, GND or VCC block is driving a block with a different sample time	These errors can usually be fixed by turning off **Specify Clock** on the constant block. If the block is fed into several clock domains, you also need to add a Tsamp block before each one.

Table 2. Model Upgrade Issues

Issue	Action
The BP block does not support sample time mode	A warning is issued if your design includes a Bus Probe (BP) block which was set to display the sample time because this option is no longer supported.
Phase selection has been standardized across all blocks	This may result in behavioral change when upgrading blocks that use phase selection.
The Multi Channel Display and Extract blocks are not supported	These blocks are no longer supported and should be removed before running the upgrade script. You can use the Avalon-ST Packet Format Converter block directly in place of these blocks. To prevent HDL being generated, insert Output blocks followed by Non-synthesizable Input blocks on the inputs to the Avalon-ST Packet Format Converter block.
HIL designs must be recompiled	For designs with Hardware in the Loop, you must recreate the Quartus II project and recompile the HIL revision after upgrading.
Changes to rounding method used for the MATLAB arrays used to initialize the LUT and RAM blocks	The rounding method used when the data values specified by an initializing MATLAB array are not exactly expressible in the chosen data type has changed. This means for example, that if you specified the data type as Unsigned Integer and the value as 1.9 in a previous release this value was rounded up to 2; in v7.1 it is rounded down to 1. You should the check the outputs from LUT or RAM blocks if an error is issued stating that the values cannot be exactly represented in the selected data format and choose revised initialization values that can be represented exactly if the outputs are not as expected.
Black box subsystem are not upgraded	Altbus blocks used as black box inputs or outputs must be manually changed to HDL Input and HDL Output blocks and a HDL Entity block added to specify the HDL file and clock/reset ports.
VCD Sink block supports a maximum of 10 inputs	Any VCD Sink blocks with >10 inputs must be replaced by separate VCD Sink blocks with no more than 10 inputs before upgrading.
An error is issued for any block name which has a '/' character	Rename any block containing a '/' character in its name.
AltBus blocks within subsystems which function as input pins not updated correctly	Move the input pins to the top level or replace them by Input blocks. (It is better to replace these AltBus blocks before upgrading to ensure that the clock signals are set correctly.)
Device Programmer block is not supported	Remove the Device Programmer block before running the upgrade program. Use Signal Compiler or a HIL block to program the DSP development board.
The External RAM block is not upgraded	This block is outside the DSP Builder system and is not automatically updated. You must manually replace any External RAM blocks in your designs with the v7.1 version.

Update the MegaCore Functions in your Design

After you have upgraded designs containing MegaCores functions, the MegaCore blocks should be regenerated to ensure they are up-to-date. This can be done either by individually double-clicking on each block, (changing the parameters, if required) and clicking the **Generate** option, or by entering the following command at the MATLAB prompt:

```
alt_dspbuilder_refresh_megacore
```

This command automatically regenerates all the MegaCore functions in the upgraded design, using the existing parameterizations.

Non-synthesizable Simulink components

In pre-v7.1 releases, it was possible to connect non-synthesizable blocks within a DSP builder system which would be automatically ignored by Signal Compiler. For example, a design may have included Simulink assertion blocks that halted the simulation if the output on a particular pin changed. It is now not possible to connect to blocks that accept doubles only; therefore it may be necessary to add non-synthesizable output blocks before such blocks.

Using a Simulink Library Forwarding Table

You can use a Simulink library forwarding table to update models with changes in the names or locations of the library blocks that they reference. Using this feature, all of the links in the upgraded version of the old file can be quickly and easily changed to point to a different library.

For more information, refer to *Making Backward-Compatible Changes to Libraries* in the MATLAB Simulink help.

How to Contact Altera

Although every effort has been made to ensure that this version of the DSP Builder software works correctly, if any problems occur, please use the following contact information to communicate your issues to the appropriate Altera representative.

Table 3. Contacting Altera (Part 1 of 2)

Information Type	Contact *Note (1)*
Technical support	www.altera.com/mysupport/
Product literature	www.altera.com/literature/
Altera literature services	literature@altera.com

Table 3. Contacting Altera (Part 2 of 2)

Information Type	Contact *Note (1)*
FTP site	ftp.altera.com

Note to Table 3

(1) You can also contact your local Altera sales office or sales representative.

Revision History

Table 4 shows the revision history for DSP Builder.

Table 4. DSP Builder Revision History

Version	Date	Revision
7.1 SP1	June 2007	Several defect fixes plus various documentation enhancements and corrections.
7.1	May 2007	New blockset includes: Single-Port RAM, True Dual-Port RAM, Bus Splitter, new Signal Compiler and TestBench, Clock and Clock_Derived (replaces ClockAltr), Stratix III DSP block (supports fine-grain control of multiply add and multiply accumulate operations for Stratix III devices) Single-port Avalon-MM blocks obsoleted in favour of the multi-port Avalon master and slave blocks. New fast functional simulation support for Video and Image Processing Suite MegaCore functions. New HDL generation flow. Support for multiple-clock domains Individual controls for optional ports and implicitly wired asynchronous clear. Added support for Arria™ GX devices.
7.0	March 2007	Added support for Cyclone® III devices.
6.1	December 2006	Added support for Stratix® III devices. Added support for the Cyclone II EP2C70 DSP development board and Santa-Cruz connectors added to the Cyclone II EP2C35 board DSPBUILDER_ROOTDIR is now a system environment variable New **All Blocks** library provides direct access to all blocks. New **Simulation** library containing External RAM block **SOPC Builder Links** library renamed as **Interfaces** library contains renamed Avalon® Memory-Mapped interface blocks and new Avalon Streaming adapter and interface blocks. New *tbdiff* comparison utility and updated *dspbuilder_sh* utility
6.0 SP1	June 2006	Updated Quartus II Global Project Assignment block and various errata fixes.
6.0	April 2006	New DSPBUILDER_ROOTDIR environment variable. Added support for Stratix II GX devices. Further enhancements to SOPC Builder integration. Extended parameter support using MATLAB workspace or masked subsystem variables. The help system now supports the MATLAB Index and Search tabs.
5.1 SP1	January 2006	Various errata fixes and documentation enhancements.

Table 4. DSP Builder Revision History

Version	Date	Revision
5.1	October 2005	New HDL Import block in Altlab library. Improved SOPC Builder integration including new Avalon blockset in SOPC Builder links library. Automatic propagation of signal names. Parameterized signal widths supported using MATLAB workspace variables. Improved error messaging. Improved documentation integrated with the MATLAB help system.
5.0.1	August 2005	Added support for the Stratix II EP2S180 DSP development board.
5.0.0	April 2005	Updated version from 3.0.0 to 5.0.0. Added support for the Cyclone II EP2C35 DSP development board.
3.0.0	January 2005	Added support for Hardware in the Loop (HIL). Added additional blocks and design examples.
2.2.0	August 2004	Added support for MegaCore functions. Added support for Cyclone II and Stratix II devices.
2.1.3	July 2003	Split documentation into separate reference manual and user guides.
2.1.2	April 2003	Added support for the Stratix EP1S80 DSP development board.
2.1.1	February 2003	Added information on using DSP Builder modules in external RTL designs. Added information on creating custom library blocks.
2.1.0	December 2002	Added support for Stratix GX and Cyclone devices. Added PLL and state machine support.
2.0.0	June 2002	New arithmetic, storage, DSP board, complex signals and SOPC Builder blocks.
1.0	October 2001	First release of DSP Builder.

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