Introduction to Megafunction IP Cores

May 2013
UG-01056-3.0

Subscribe Feedback

Altera provides a free library of parameterizable intellectual property (IP) blocks, called megafunctions, that are optimized for efficient logic synthesis in Altera devices. Use megafunctions to access architecture-specific features within memory, DSP blocks, shift registers, and other simple and complex functions.

You can quickly define a megafunction variation in the MegaWizard Plug-In Manager GUI. Alternatively, you can define and instantiate megafunctions directly in HDL files. The Quartus II software automatically infers megafunction code whenever HDL code can be optimized by megafunction substitution.

Altera Provided Megafunctions

The Quartus II software implements your specified system or IP core parameters and generates files for synthesis and simulation in the Quartus II software and other EDA tools. The Quartus II software includes the following megafunctions for use without additional license.

Arithmetic Megafunctions

The Quartus II software includes these arithmetic megafunctions.

Megafunction Name	Function
LPM_ABS	Absolute value
LPM_ADD_SUB	Adder/Subtractor
LPM_COMPARE	Comparator
LPM_COUNTER	Counter
LPM_DIVIDE	Divider
LPM_MULT	Multiplier
ALTACCUMULATE	Accumulator
ALTECC	ECC Encoder/Decoder
ALTMEMMULT	Memory Constant Coefficient Multiplier
ALTMULT_ACCUM	Multiply-Accumulator
ALTMULT_ADD	Multiply-Adder

Megafunction Name	Function
ALTMULT_COMPLEX	Complex Multiplier
ALTSQRT	Integer Square-Root
PARALLEL_ADD	Parallel Adder

Floating-Point Megafunctions

The Quartus II software includes these floating-point megafunctions.

Megafunction Name	Function
ALTFP_ADD_SUB	Adder/Subtractor
ALTFP_DIV	Divider
ALTFP_MULT	Multiplier
ALTFP_EXP	Exponential
ALTFP_INV	Inverse
ALTFP_INV_SQRT	Inverse Square Root
ALTFP_LOG	Natural Logarithm
ALTFP_ATAN	Arctangent
ALTFP_SINCOS	Trigonometric Sine/Cosine
ALTFP_ABS	Absolute value
ALTFP_COMPARE	Comparator

Gate Megafunctions

The Quartus II software includes these gate megafunctions.

Megafunction Name	Function
LPM_CLSHIFT	Combinatorial logic shifter
LPM_CONSTANT	Constant generator
LPM_DECODE	Decoder
LPM_MUX	Multiplexer

I/O Megafunctions

The Quartus II software includes these I/O megafunctions.

Megafunction Name	Function
ALTLVDS	LVDS transmitter and receiver
ALTPLL	Phase-Locked Loop (PLL)
ALTPLL_RECONFIG	PLL reconfiguration
ALTERA_PLL	Phase-Locked Loop for Stratix® V devices only
ALTREMOTE_UPDATE	Remote Update
ALTGXB	Gigabit transceiver block (GXB)
AL2TGXB	Gigabit transceiver block II (GXB)
ALTOCT	On-Chip Termination
ALTCLKCTRL	Clock Control
ALTDDIO	Double Data Rate I/O
ALTIOBUF	I/O Buffer
ALTTEMP_SENSE	Temperature Sensor
ALTDLL	Delay Locked Loop (DLL)
ALTDQ and ALTDQS	DQ and DQS
ALTDQ_DQS2	DQ and DQS for Stratix V
ALTASMI_PARALLEL	Active Serial Memory Interface

Memory Megafunctions

The Quartus II software includes these memory megafunctions.

Megafunction Name	Function
RAM and ROM	Internal memories
SCFIFO and DCFIFO	Single clock FIFO and Dual Clock FIFO
FIFO Partitioner	First-In First-Out Partitioner
LPM_SHIFTREG	Shift-register
ALTSHIFT_TAPS	RAM Shift Register
ALT_UFM	User Flash Memory
ALTOTP	One-Time Programmable function

JTAG Megafunctions

The Quartus II software includes these JTAG extensible megafunctions.

Megafunction Name	Function
PFL	Parallel Flash Loader
SLD_VIRTUAL_JTAG	Virtual JTAG

Related Information

Megafunctions and IP Documentation Web Page

Customizing Megafunctions

You can customize a variation of a megafunction IP core for use in your design. Using megafunctions in your design involves the following steps:

1. Identify the megafunction that best meets your design and target device requirements.
2. To define and instantiate a megafunction using the GUI, click **Tools** > **MegaWizard Plug-In Manager** and follow the wizard to define your megafunction. The Quartus II software automatically generates synthesis and optional simulation output files.
3. Use the Block Editor or Qsys to connect the megafunction to other elements in the design.
4. Compile your design in the Quartus II software. Optionally generate a netlist for other EDA tools.
5. Simulate your design in your preferred EDA simulator.

Related Information

Connecting Megafunctions on page 5

Using HDL Code Templates on page 7

Synthesizing Megafunctions in other EDA Tools on page 11

Customizing Megafunctions in the GUI

The MegaWizard Plug-In Manager GUI allows you to define and instantiate a custom variation of an Altera megafunction. You can edit megafunctions at any time. Megafunctions defined in your project appear in the Project Navigator. To edit a megafunction, double-click the megafunction file in the Project Navigator or Block Editor to display the MegaWizard GUI. To customize a megafunction using the MegaWizard Plug-In Manager GUI, follow these steps:

1. Launch the MegaWizard Plug-In Manager using any of the following methods:
   - In the Quartus II software, click **Tools** > **MegaWizard Plug-In Manager**.
   - In the Project Navigator, right-click a megafunction file and click **MegaWizard Plug-In Manager**.
   - In the Block Editor, click **Edit** > **Insert Symbol as Block**. In the Symbol editor, click **MegaWizard Plug-In Manager**.
   - Start the stand-alone version of the MegaWizard Plug-In Manager GUI by typing the following command at the command prompt:

   `qmegawiz` (for GUI or command-line mode)

   or

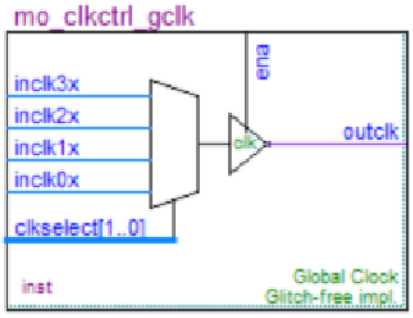
`qmegawizq` (for GUI only)

2. Specify **Create**, **Edit**, or **Copy** a megafunction.
3. In **Which device family will you be using?** select your target device family.

   Only megafunctions available for the target device are available in **Which megafunction would you like to customize?** Unsupported megafunctions are grayed out.
4. Specify the name and file format of the output file. Click **Next**
5. Parameterize the megafunction by specifying options in the wizard. Click **Next** .
6. If the wizard includes **EDA** and **Summary** tabs, follow these steps:
   - Some third-party synthesis tools can use a grey box netlist that contains the structure of an IP core without detailed logic to optimize timing and performance of the design containing it. To use this feature, turn on **Generate Netlist** to generate a netlist file for area and timing estimation instead of a wrapper file.
   - On the **Summary** tab, select the files you want to generate. A gray checkmark indicates a file that is automatically generated. All other files are optional. This step instantiates the megafunction into your HDL code and creates a wrapper file.
7. Click **Finish**. The megafunction variation is generated along with the files you specify.
8. To view the megafunction schematic, open the generated block symbol file (**.bsf**) located in your project directory. The megafunction block symbol appears in the Symbol window.

   You can edit megafunctions at any time. Megafunctions defined in your project appear in the Project Navigator. To edit a megafunction, double-click the megafunction file in the Project Navigator or Block Editor to display the MegaWizard GUI.

Figure 1: Example Parameterized Global Clock Control Module



Related Information

Creating a System with Qsys

Connecting Megafunctions

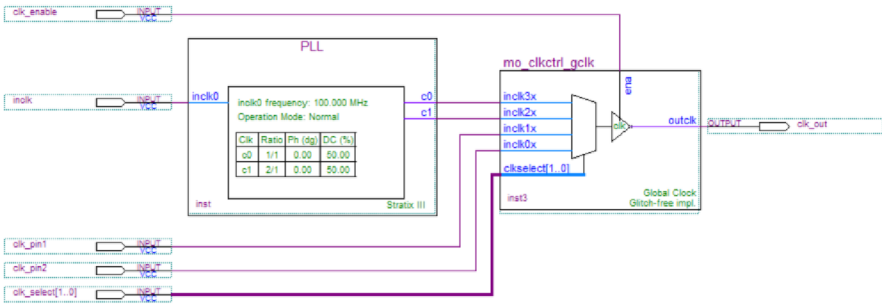
You can easily visualize and connect megafunctions to other elements of your design with the Quartus II Block Editor. You can make connections by drawing node (wire), bus, and conduit connections between blocks representing megafunctions and primitives in the schematic. If the I/O signal names in one block match those in another connected block, the Quartus II software automatically connects common I/O's

between the blocks. If the I/Os in two connected blocks are differently named, you can assign a name that matches a block I/O port to connect the port to the bus or conduit, or you can map the signal connection by name explicitly. You can also connect megafunctions included in a Qsys system in the Qsys GUI. To connect a megafunction using the block editor, follow these steps:

1. To draw a wire, bus or conduit line connecting one or more block, perform any of the following:
   - To draw a conduit, click the **Selection and Smart Drawing Tool** or the **Orthogonal Conduit Tool** button on the toolbar. The smart selection and drawing tool automatically changes to the correct node, bus, or conduit tool when you drag it from a block or symbol border.
   - To draw a bus, click the **Orthogonal Bus Tool** button on the toolbar.
   - To draw a wire, click the **Orthogonal Node Tool** button on the toolbar.

   The Quartus II software automatically connects common I/O's between the blocks
2. To map differently named I/O between two megafunctions, follow these steps:
   a. At a conduit connection, double click the mapper symbol. The **Mapper Properties** dialog box appears.
   b. On the **General** tab, specify the I/O **Type**.
   c. Click the **Mappings** tab.
   d. Under **Conduit Mappings**, select the name of your I/O on block and the corresponding **Signals in conduit**.
   e. Click **Add**. A logical connection is created between the signals.
3. In the File menu, click **Save**.

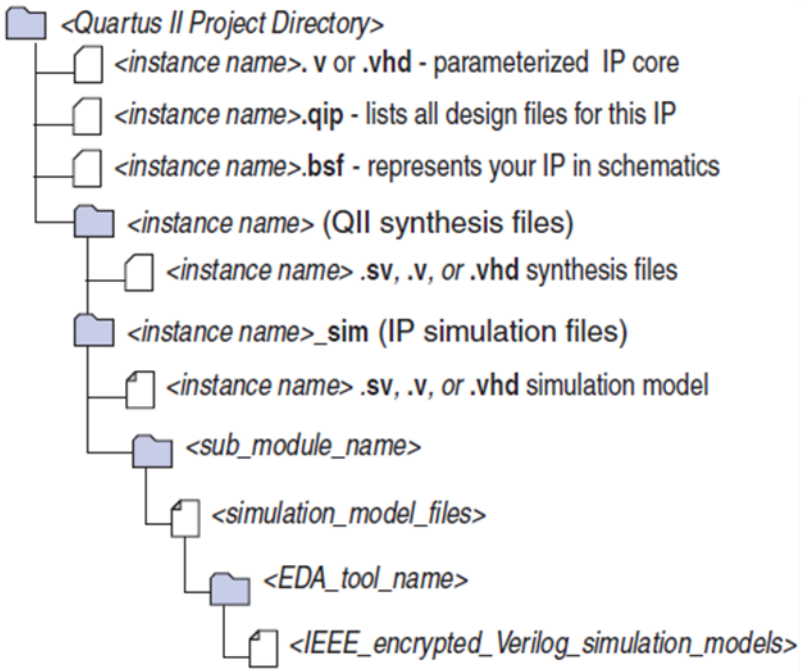
Figure 2: Example Global Clock Control Connections



Generated Megafunction Files

The MegaWizard Plug-In Manager generates one or more of the following files for synthesis and simulation of the megafunction IP core in the Quartus II software and other EDA tools.

Figure 3: IP Files Generated by MegaWizard Plug-In Manager



Related Information

Synopsys Simplify Support

Instantiating Megafunctions in HDL

You can instantiate a megafunction directly in your HDL code by calling the megafunction and setting its parameters as you would in any other module, component, or subdesign. When instantiating a megafunction in VHDL, be sure to include the correct libraries.

Using HDL Code Templates

The Quartus II software includes code examples or templates for inferred RAMs, ROMs, shift registers, arithmetic functions, and DSP functions optimized for Altera devices. To access HDL code templates to define megafunctions, follow these steps:

1. Open a file in the text editor.
2. On the **Edit** > **Insert template**.
3. In the **Insert Template** dialog box, click the + icon to expand either the **Verilog HDL** category or the **VHDL** category, depending on the HDL you prefer.
4. Under **Full Designs**, expand the navigation tree to display the type of functions you want to infer.
5. Select the function to display the code for the selected template in the Preview pane, and click **Insert**.

Note: For new DSP features optimized for Arria V, Cyclone V, and Stratix V, expand the **Arithmetic** category, and then expand the **DSP features (Stratix-V, Arria-V and Cyclone-V)** category.

Example Top-Level Verilog Module

Verilog HDL ALTFP_MULT in Top-Level Module with One Input Connected to Multiplexer.

```
module MF_top (a, b, sel, datab, clock, result);
    input [31:0] a, b, datab;
    input clock, sel;
    output [31:0] result;
    wire [31:0] wire_dataa;

    assign wire_dataa = (sel)? a : b;
   altfp_mult inst1 (.dataa(wire_dataa), .datab(datab), .clock(clock),
 .result(result));

    defparam
         inst1.pipeline = 11,
         inst1.width_exp = 8,
         inst1.width_man = 23,
         inst1.exception_handling = "no";
endmodule
```

Example Top-Level VHDL Module

VHDL ALTFP_MULT in Top-Level Module with One Input Connected to Multiplexer.

```
library ieee;
use ieee.std_logic_1164.all;
library altera_mf;
use altera_mf.altera_mf_components.all;

entity MF_top is
     port (clock, sel  : in  std_logic;
           a, b, datab : in  std_logic_vector(31 downto 0);
           result      : out std_logic_vector(31 downto 0));
end entity;

architecture arch_MF_top of MF_top is
signal wire_dataa : std_logic_vector(31 downto 0);
begin

wire_dataa <= a when (sel = '1') else b;

inst1 : altfp_mult
     generic map (
             pipeline => 11,
             width_exp => 8,
             width_man => 23,
             exception_handling => "no")
     port map (
             dataa => wire_dataa,
             datab => datab,
             clock => clock,
             result => result);
end arch_MF_top;
```

Example Megafunction Inference

The Quartus II software infers following the following Verilog HDL code as the LPM_MULT or ALTMULT_ADD megafunctions for an unsigned and a signed multiplier. Each example fits into one DSP block 9-bit element. In addition, when register packing occurs, no extra logic cells for registers are required.

```
module unsigned_mult (out, a, b);
   output [15:0] out;
   input [7:0] a;
   input [7:0] b;
   assign out = a * b;
endmodule


module signed_mult (out, clk, a, b);
   output [15:0] out;
   input clk;
   input signed [7:0] a;
   input signed [7:0] b;
   reg signed [7:0] a_reg;
   reg signed [7:0] b_reg;
   reg signed [15:0] out;
   wire signed [15:0] mult_out;
   assign mult_out = a_reg * b_reg;
   always @ (posedge clk)
   begin
       a_reg <= a;
       b_reg <= b;
       out <= mult_out;
   end
endmodule
```

Related Information

Recommended HDL Coding Styles

Quartus II Integrated Synthesis

Using qmegawiz Command-Line Executable

You can use qmegawiz command-line version of the MegaWizard Plug-In Manager that allows you to modify, update, or create variation files without using a GUI. The following table lists common qmegawiz syntax options and arguments To use qmegawiz, use the following syntax:

```
qmegawiz [options] [module=<module name>]| wizard=<wizard name>]
[<param>=<value>...<port>=<used>|<unused>...] <variation file name>
```

Options/Arguments	Description
`-silent`	Runs the MegaWizard Plug-In Manager in command-line mode, without displaying the GUI.
`-f:`*<parameter file>*	Specifies a **.txt** name that contains all the parameter and port values.

Options/Arguments	Description
`-p:`<*working directory*>	Specifies the default working directory that qmegawiz uses when it generates files.
`module=`<*module name*>	Specifies the module or wizard name. When there are multiple wizard names that correspond to one module name, use the wizard option to specify one wizard. When there are multiple module names that correspond to one wizard name, use the module option to specify one module.
`wizard=`<*wizard name*>	
<*param*>=<*value*>	Specifies the parameter values.
<*port*>=<*used*> | <*unused*>	Specifies whether the ports are used.
<*variation file name*>	Specifies a variation file name. Valid extensions are **.v**, **.vhd**, or **.tdf**.

For example,

```
qmegawiz -silent module=altlvds_rx wizard=altlvds
    common_rx_tx_pll=ON tx_coreclock=used lvds_sample.v
```

Related Information

Command-Line Scripting

Using IP-generate Command-Line Executable

You can use `ip-generate` to create or modify custom megafunction variations. To run the `ip-generate` command, follow these steps:

1. Type the following command at the command prompt of your operating system:
   <*ACDS installation directory*> `\quartus\sopc_builder\bin\`
2. To run the executable type `ip-generate`.
3. To instantiate the megafunction using the executable file, type the following syntax: `ip-generate --component-name=altdq_dqs2 --component-system-param=DEVICE_FAMILY="Stratix V" --file-set=QUARTUS_SYNTH --output-name[=<file_name>] --component-param[=<parameter_name>][=<parameter_value>]`

Related Information

Command-Line Scripting

IP-Generate Arguments

IP-generate accepts these common arguments.

Options/Arguments	Description
--component-name=<*variant name*>	Specifies the megafunction variant name.
`--file-set=QUARTUS_SYNTH --output-name[=<file_name>]`	A parameter that is used by the ip-generator to specify an output file. Valid extensions are **.v**, **.sv**, **.vhd**, or **.tdf**.
`--component-system-param=DEVICE_FAMILY ="<device family name>"`	Specifies the target device family.
`--component-param[=<parameter_name>] [=<parameter_value>]`	Specifies the target device family.

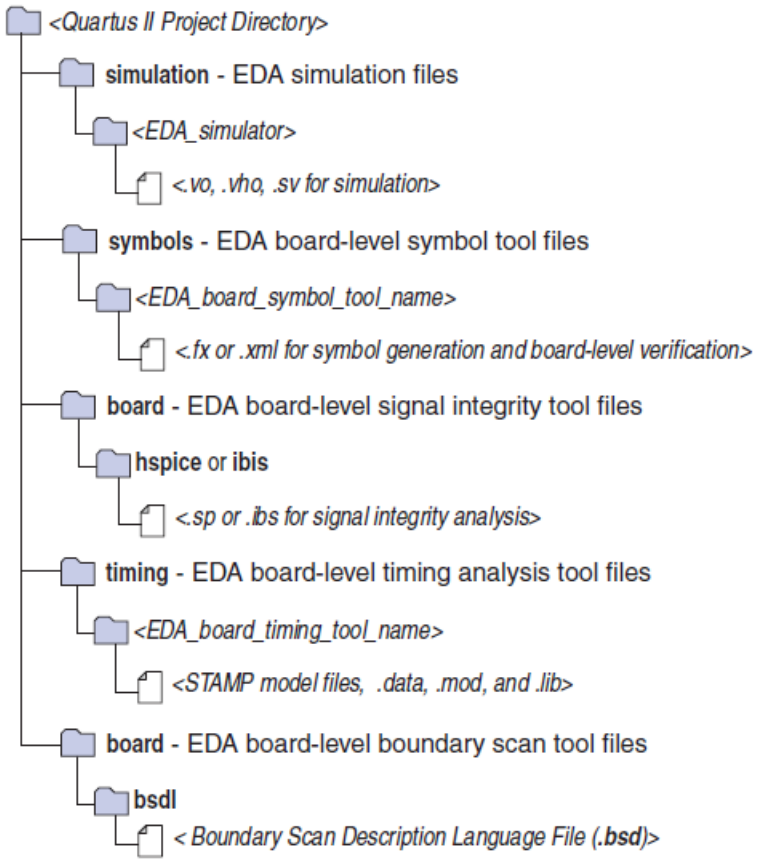
Synthesizing Megafunctions in other EDA Tools

You can use supported EDA tools to synthesize a design that includes megafunctions. When you generate megafunction synthesis files for use with third-party EDA synthesis tools, you can optionally create an area and timing estimation netlist.

area and timing estimation netlist describe the megafunction connectivity and architecture, but not details about true functionality (grey box). This information enables certain third-party synthesis tools to better report area and timing estimates. In addition, synthesis tools can use the timing information to achieve timing-driven optimizations and improve the quality of results.

The netlist file is called <*variant name*>_**syn.v** file. The Quartus II software generates this file in Verilog HDL format regardless of the output file format you specify. If you use this netlist for synthesis, you must include the megafunction wrapper file <*variant name*>**.v** or <*variant name*>**.vhd** in your Quartus II project for placement and routing.

Figure 4: Quartus II Generated Files for Other EDA Tools



Related Information

Quartus II Integrated Synthesis

Simulating Megafunctions

Simulation verifies design behavior before device programming. The Quartus II software supports RTL and gate level design simulation of megafunction IP cores in other EDA simulators. Simulation involves setting up your simulator working environment, compiling simulation model libraries, and running your simulation. Altera provides various tools to help you quickly setup and run simulation. You can use the Quartus II NativeLink feature to automatically generate simulation files and scripts. NativeLink launches your preferred simulator a from within the Quartus II software.

Use a custom flow for more control over all aspects of simulation file generation. Alternatively, the Simulation Library Compiler automatically compiles and stores the correct simulation model libraries for functional and gate-level timing simulation of your design.

Related Information

Simulating Altera Designs