CerebrumCore Package Definition XML Document Specifications

PSU Engineering (Cerebrum) Team

Created: 11 February 2011

Updated: 11 February 2011

CHANGE LOG

Version	Date	Name	Description
1.0	11 Feb 2011	M. Cotter	Created the document

Table of Contents

1 <CerebrumCore> Definition Structure

The top-level element of the document, encompasses all attributes and parameters needed to utilize a CerebrumCore package.

1.1 <General> General Core Attributes

This section defines basic attributes for cataloging, accessing, and identifying the cores in both the framework and a project design.

1.1.1 Visible (Boolean)

By default this value is True. This flag indicates whether this core should be displayed in the Library toolbox of the Cerebrum Design tool. If this value is set to false, the core will be loaded, but will not be accessible via the design library toolbox.

1.1.2 Name

This value specifies the friendly name of the core. Spaces and special characters are allowed, but the raw text in the XML file must conform to XML specifications to be properly parsed without error.

1.1.3 Type

This value specifies the type-name of the core. Only alphanumeric characters and the underscore ("_") are permitted to be used in this field.

1.1.4 Version

This value indicates the simple version number using the "#.##.X" format. (i.e. "1.02.d")

1.1.5 Description

This value is a short to average length description of the functionality provided by the core. This information will be displayed on tooltips in the Cerebrum design tool.

1.1.6 Owner

This value specifies the name or organization which created and therefore "owns" this core.

1.1.7 CoreServer (deprecated)

This value indicates whether a core-server application should be compiled for this core, if Embedded Linux is being compiled for any processors on its FPGA.

1.1.8 InstancePrefix

This value indicates the string that should be used as the prefix for new instances of this core when created in the design framework. For example, an InstancePrefix of "mycore" would result in instances such as "mycore_0", "mycore_1", etc.

1.1.9 Keywords (Not Yet Supported)

This value specifies a semi-colon delimited list of keywords that are relevant to this core, for use when searching for cores in the design library.

1.2 Software – Attributes Used by the Cerebrum Tool Suite

The attributes defined in this section pertain specifically to the Cerebrum Graphical Design Tool.

1.2.1 DesignDisplay

This subsection defines how the component is listed on the design library toolbox and how it is displayed when added to the design layout.

1.2.1.1 Category

This value specifies the category under which this component should be listed in the design library toolbox.

1.2.1.2 Image

This value specifies the path to an image file to be use for display on both the design library toolbox and layout. The path must be specified relative to the location of this definition file. The image file need not be in the same directory—it may be in a subdirector--but it should not be in any location above this file's directory in the file system.

1.2.1.3 DefaultSize

This is the default size of the component when added to the design layout, specified in pixels.

1.2.1.4 Ports

This subsection defines IO ports available on the component, how they are displayed on the design layout, and how they map to hardware cores within the component.

1.2.1.4.1 Port

The port attribute defines several properties of the port.

- Instance – This value specifies the internal instance name of the hardware core to which this port corresponds. If this value is not specified, this value defaults to the **FIRST** <Interface> defined in the <Hardware> section. (See Interface for more details).
- Type – The permissible values for this attribute are: INITIATOR, TARGET, INPUT, and OUTPUT.
- Name – This value specifies the friendly name assigned to the port.
- *Interface – This value specifies the interface used by the port. Only ports with the same interface may be connected to this port. (Currently working on implementation of this).*
- X, Y – These value, legal values ranging from 0 to 100, specify the horizontal (X) and vertical (Y) location of this port on the design object when displayed in the layout. These values are normalized to percentages of the component's width or height. For example, a value of 0 would be the left(top)-most coordinate, while 100 would correspond to the right(bottom)-most coordinate.

1.2.2 Properties

This section specifies the set of customizable properties that are available for setting at design time.

1.2.2.1 Property

The port attribute defines several aspects of the property and how it is displayed in the design tool.

- Name – This specifies the friendly name of the property when displayed in the property editor.

- Default – This specifies the default value of this property. When a value list is specified via the Values property, this value should indicate the RAW value, rather than the display value.
- Category – This specifies the category/property tab under which this property should appear.
- Description – This specifies the tooltip to be displayed when hovering over the property in the property editor dialog.
- Type – This specifies the type of the property. (Currently unused – All Properties are treated as strings)
- Values – This specifies a comma-delimited list of Name/Value pairs to be used for populating a list of fixed allowable values. For example, "A=B, C=D" would result in the values "A" and "C" appearing in the property editor. Choosing "A" would assign a value of "B" to the property. The name of each value may contain spaces, but may NOT contain commas.
- Core – This value specifies the internal core instance that this property is related to. If this value is not specified, the property is treated as a "General" property of the entire component rather than any individual core within the component.

1.3 Hardware – Attributes Used by the Cerebrum Tool Suite

This subsection defines the internal hardware cores of the component, and their interfaces to communication infrastructure of the Falcon Framework.

1.3.1 Interface

Any number of Interface Elements may be specified under the Hardware Element. The inner text of each Interface element specifies the internal instance core representing the interface. Each interface element also must specify a Type attribute. Legal values of this attribute are "SAP" or "SOP", depending on the interface exposed by the internal core.

1.3.2 PCores

This defines all of the Xilinx-style pcores that are to be included as part of this component.

1.3.2.1 PCore

The PCore attribute specifies several properties used to locate and configure each pcore within the component.

- Type – The type-name of the pcore.
- Version – The version of the pcore.
- Instance – The internally-used instance name of the pcore.
- Source – If specified, indicates the path, relative to this file's directory, to locate the pcore's definition, HDL, and configuration.
- Valid – String that is evaluated, within the context of the component's properties, to determine whether the pcore should be included with others in the component.
- UCF – Specifies a path, relative to this file's directory, to a customized UCF-specification file to be included for this core.

1.3.3 SupportedArchitectures

This section specifies the list of FPGA architectures that this component can be instantiated on. Each supported architecture is listed as the inner text of an <Arch> element below the SupportedArchitectures element.

1.3.4 Resources

This section specifies the resources that are required by this component, including all of its included pcores. Each resource type is specified with its own Resource element. Each Resource element has 2 attributes: Name and Amount, which specify the type and quantity of resource required, respectively.

1.3.5 MHSImports

This section defines a set of MHS-inspired XML files to be used as a basis for the internal connectivity of cores within the component. Each MHS specification to be imported is listed in its own <MHSImport> tag below the MHSImports element. Each MHSImport element has a single tag, Source, whose value specifies the path to the file to be included. This path is specified as relative to this file's directory.

1.3.6 Clocks

This section defines the set of clock signals generated and required by the pcores within the component.

1.3.6.1 Clock

The Clock attribute specifies several properties used to define a clock signal, its directionality, and the pcore with which it is associated.

- Core – This specifies the internal pcore instance to which this clock signal is attached.
- Port – This specifies the port of the internal pcore to which this clock signal is attached.
- Direction – This specifies whether the clock is a required input ("IN" or "INPUT") clock, or is an output ("OUT" or "OUTPUT") clock generated by the pcore. If this attribute is not specified, the default direction is INPUT.
- Frequency – This specifies the frequency of the clock signal.
- Phase – This specifies the phase of the clock signal. Valid values range from 0 to 360. Values outside this range are modulated to fall within this range. If the Match attribute is specified, this phase adjustment is applied on top of any phase applied to the Matched clock signal.
- Group – This specifies the Clockgroup to which this signal belongs. Legal values for this are "NONE", "PLL0", and "PLL0_ADJUST"
- Buffered – This specifies whether the clock signal is buffered. Legal values are "True" and "False". The default value is "True".
- *Match* – If this value is specified, this clock signal is matched, as a ratio, to another clock signal attached to the same core. The value of this attribute indicates the Port name of the clock to match.
- *Ratio* – If the Match attribute is specified, this value specifies the ratio of this clock's frequency to the Match frequency. For example, attributes [Match="200mhzclk" Ratio="0.5"] would cause this clock's frequency to be set to half of the frequency of the clock signal attached to port "200mhzclk".

1.3.7 Resets (Not Yet Implemented)

This section defines the set of reset signals generated and required by the pcores within the component.

2 MHSImport File Structure

The MHSImport file defines internal connectivity of pcores within a component. All instance names within the file, and any signals containing those instance names will be transformed during XPS project generation to conform to the new instances assigned to each pcore.

2.1 <MHSImport>

This is the top-level element of the file. All entries are specified below this.

2.2 <Ports>

This section specifies any top-level global ports to be exposed by the system.

2.2.1 <Port_Pin>

Each Port_Pin element supports 2 attributes:

- Value – This is the literal string to be included as a PORT definition in the MHS file.
- Valid – This is a string, evaluated within the context of the component's properties, which specifies whether this element should be included in the resulting file.

2.3 <Cores>

This section specifies the pcores, parameters, and signal connections as defaults in the component.

2.3.1 <Core>

Each Core element requires 3 attributes, and supports a 4th optional attribute:

- Name – This is the type name of the pcore.
- Instance – This is the internal instance of the pcore.
- HW_Ver – This is the version of the pcore.
- Valid – This is a string, evaluated within the context of the component's properties, which specifies whether this core should be included in the resulting file.

2.3.1.1 <Parameter>, <Port>, and <Bus_Interface> Elements

Each of these elements, specified under the <Core> element corresponds to a Parameter, Port, or Bus_Interface that is defined and connected internally within the component. Each of these elements supports a single attribute. The name of the attribute is the case-sensitive name of the Parameter, Port, or Bus_Interface, and the value of the attribute is the value or signal name assigned to the Parameter, Port, or Bus_Interface.

3 CustomUCF File Structure

The CustomUCF file defines UCF constraints that are applicable and/or required for the pcore which includes the file from the component definition.

3.1 <UCF>

This is the top-level element of the file. All entries are specified below this.

3.2 *<ucf_entry>*

The format of these entries is generalized. The name of each element corresponds to the UCF entry (i.e. "NET", "TIMEGRP", "TNM_NET", etc). The Contents attribute of each specifies the string to be written as part of the element. Any appearance of the including pcore's internal instance name will be transformed to its true instance name when processed. Each of these highest-tier elements will be automatically terminated with a semicolon during processing and so these trailing semicolons should NOT be included. Additionally each of these elements may define sub-elements which will continue the entry prior to termination with a semicolon. Any special characters such as quotation marks may need to be XML-escaped (i.e. """ for quotation marks, "<" for less-than, etc)

3.3 Example

In this example, assume the following situation:

Component Instance: `demo_core_0`
Internal PCore Instance: `sample0` [includes sample_ucf.xml]
Actual PCore Instance: `demo_core_0_sample0`

sample_ucf.xml

```
<UCF>
   <NET Contents="&amp;quot;*sample0/sampleCLKTX&amp;quot;">
      <TNM_NET Contents="&amp;quot;TXCLK0&amp;quot;" />
   </NET>
   <TIMEGRP Contents="&amp;quot;client_clk_tx0&amp;quot; = &amp;quot;RXCLK0&amp;quot;" />
   <TIMESPEC Contents="&amp;quot;TS_client_clk_tx0&amp;quot; = PERIOD &amp;quot;
client_clk_tx0&amp;quot; 100 ps HIGH 50" />
</UCF>
```

This situation would produce the following UCF entries:

```
NET "demo_core_0_sample0/sampleCLKTX" TNM_NET "TXCLK0";
TIMEGRP "client_clk_tx0" = "TXCLK0";
TIMESPEC "TS_client_clk_tx0" = PERIOD "client_clk_tx_0" 100 ps HIGH 50;
```

4 References