Specifications for Platform.xml
Ver. 4.00.a

*PSU Engineering (**Cerebrum**) **Team***

Created: 29 June 2010

Last Updated: 26 June 2011

CHANGE LOG

Version	Date	Name	Description
1.0	18 Aug 2010	S. Nathanson	Created the document
2.0	29 Jun 2010	M. Cotter	Updated documentation for Jun 2010 release.
3.0	11 Feb 2011	M. Cotter	Added the “CHANGE LOG” section to document Updated documentation for Feb 2011 release.
4.0	26 Jun 2011	M. Cotter	Updated documentation on FPGA specification in board document.

Table of Contents

1 Introduction

The platform.xml file is an XML file which describes the physical aspects of the platform that Cerebrum works with. It is meant to only change when the platform changes, so it will remain static for a normal execution of Cerebrum.

2 Platform Hierarchy

The definition of a Cerebrum Platform is organized into a 3 part hierarchy. These parts are: Platform, Board, and FPGA. The Platform is the top-level object that defines a single, all-encompassing hardware environment. The Platform then defines 1 or more boards that comprise this hardware environment. Each board then, defines 1 or more FPGAs that comprise the board.

2.1 Directory Structure

Within the Cerebrum Platforms repository, each Platform is defined by a name and its entire definition is contained within a subdirectory with the same name. Each platform directory then contains an XML file that defines the boards within the platform, as well as additional subdirectories that correspond to each board defined in the platform.

The definition of a board follows the same hierarchical approach, and consequently the same format, as that of a platform. Each board directory contains an XML file that defines the FPGAs on the board and subdirectories that correspond to each FPGA defined on the board.

Finally, each FPGA directory contains 3 important files. The first of these is an XML file that specifies the internals of the FPGA—ports, I/O Interfaces, resources, etc. The second is "base_system.xml" that defines all of the cores and parameters that are minimally required for Cerebrum to support the FPGA. The third file is "system.ucf", which is a Xilinx User Constraint File that corresponds to the cores and interfaces defined in "base_system.xml".

3 Platform Definition

The top-level platform is defined by a single XML file that identifies all boards in the platform along with a pointer to a board definition file for each board. An example file is shown in the example in section 6.2.

4 Board Definition

Each board file follows a very similar definition as the platform. Each FPGA is identified on the board with an ID and a pointer to an FPGA definition file. In addition to defining the FPGAs assigned to each board in the platform, this tier of the hierarchy also defines and parameterizes any Cerebrum cores that are ***required*** to be instantiated on each FPGA of the board. Any custom bitfile generation scripts may also be specified at this level. An example file is shown in the example in section 6.2.

4.1 FPGA

4.2 BitGenScript

This node element specifies any bitgen scripts that should be used for synthesis of the FPGA.

4.3 RequiredCore

A RequiredCore node specifies a Cerebrum Core package to be included on the FPGA any time the platform is utilized.

4.3.1 Source

This attribute indicates the name of the folder where the component's package and source files may be found.

4.3.2 Type

This attribute indicates the type-name of the component to be included.

4.3.3 Version

This attribute indicates the version number of the component to be included.

4.3.4 DesignVisible

This Boolean value indicates whether this core should be shown within the design UI environment, or hidden from the user and added only during component mapping.

4.3.5 Parameter Instance

This node must have a single attribute, named instance, the value of which indicates the instance name of the core to be utilized within the design.

4.3.6 Parameter/Port/CerebrumProperty

These nodes specify any parameter, port, or CerebrumProperty to be associated with the required component/core. Each node element should specified 3 attributes: name—the name of the entry, value—the value of the entry, and core—the core within the component the entry is associated with, if any.

5 FPGA Definition

The FPGA definition file is the heart of the platform definition and correspondingly is the largest and most complex of the files in the platform definition. It is heavily influenced by and based upon Xilinx's pre-existing XBD file format. Some additional parameters and attributes are required by the Cerebrum design framework and have been added here.

There are four modules: IO_Interface, IO_Adapter, FPGA, and Global_Attributes. Under these types are three subtypes: Parameter, Port, and Attribute. Parameter is exclusive to the IO_Interface, Port can be used by the IO_Interface or the IO_Adapter, and Attribute is a subtype for all three. As a rule, anything that can't be found in the XBD should be flagged with an XML comment and all comments in the XBD should be XML comments in the file.

5.1 Modules

Three types are taken from the XBD format. Each of these should be joined by Instance = <instance name>.
For example

```
<FPGA Instance = “FPGA_0”>
   Subtypes…
</FPGA>
```

If you are familiar with the format, then you can skip ahead to the Global_Attributes section where there are new additions.

5.1.1 IO_Interface

This covers physical modules on the board. Each block specifies some physical module on the board other than an FPGA. Each one also holds an IP which can be placed on the FPGA to interface with the physical module.

5.1.2 IO_Adapter

This refers to some glue logic that could be used to connect an IO_Interface’s pins with the IP mentioned above.

5.1.3 FPGA

This represents an FPGA on the board. The properties are covered in more depth on the attributes subtype

5.1.4 Global_Attributes

Though the XBD format does not specify a section for global attributes, this format does. All of these attributes refer to the board itself, rather than a specific module.

Global Attributes

Vendor	Name of the vendor
Name	String containing the name and package of a board
Revision	A unique number which applies to one board revision and only one board revision
Contact_Info_URL	“http://www.xilinx.com/support/techsup/tappinfo.htm” for completeness
Spec_URL	“http://www.xilinx.com” for completeness with the XBD
Desc	Short description of the board
Long_Desc	Long description of the board. Single quotes are needed if it contains commas.
Resource	Name of a resource followed by its amount. When used as a global attribute, it refers to global memory.

5.2 Subtypes

Each subtypes has a set of subproperties, each of which also has its own value. The first entry is the name for parameter and port and the instance for attributes. All entries after the first are subproperties and should be formatted like so:

```
…
```

```
<Parameter num_bits = “11” IO_Is = “num_bits”/>
…
```

5.2.1 Parameter

There is no upper bound on how many parameters a module can have, but each parameter must have a name and some value associated with it which is the first entry in the paramter’s line.
The subproperties are as follows:

IO_Is	Pairs IO_Adapter with its corresponding port on the FPGA
Memory_Type	Currently can only be FLASH
Range	Comma separated integers, intended to be the clock frequencies allowed on the clock module
Value_Note	A string which describes the parameter and its value

5.2.2 Port

Dir	Specifies the direction the port on the FPGA is going
Interrupt_Priority	Sets the level of the priority for interrupts on the port
IO_Is	Pairs the port with a peripheral
Sensitivity	Determines what signal will cause an interrupt (ex. Low or high)
SigIs	Designates the port as an interrupt port
UCF_Net_Strings	Constraints with the NET associated with the port. Takes a string which is composed of comma separated, single-quote strings.
InitialVal	The value the port is driven to if there is no corresponding core

5.2.3 Local_Attributes

Attributes differ from parameter and port in that they occur on the same level as the module name. Below is an example of an FPGA with the attribute Instance = FPGA_0

```
<FPGA Instance = “FPGA_0”>
   Subtypes…
</FPGA>
```

With the exception of instance, the attributes each refer to one specific type. They are detailed below as follows:

Universal Attributes

Instance	Differentiates modules

IO_Adapter Attributes

Corename	Identifies the core used by IO_Adapter
Version	Version of the pcore.

IO_Interface Attributes

IOType	Describes the IO_Interface block.
Exclusive	Names a group of mutually exclusive IO_Interfaces. No two IO_Interfaces in the same group can be used simultaneously.

FPGA Attributes

JTAG_Position	Determines the FPGA's position in the JTAG chain
Speed_Grade	The speed grade of the FPGA
Neighbors	FPGA's which it has a direct link to and the link speed
Resource	The amount and type of resources available on the FPGA

6 Platform Structure/Hierarchy Example

Below is a list of example full paths and file names for a Platform consisting of 2 ML510 boards. In this example, the platform name is "`Twin_ML510`" and the Platforms repository is located in "`C:\Cerebrum\Platforms`". The corresponding contents of the Platform XML and Board XML files is also shown below the list.

Note that the Board **directory name** corresponds to the Board XML filename, and the FPGA **directory name** corresponds to the FPGA XML filename.

6.1 File/Directory List

```
C:\Cerebrum\Platforms\Twin_ML510\Twin_ML510.xml

C:\Cerebrum\Platforms\Twin_ML510\ML510_0\ML510_0.xml
C:\Cerebrum\Platforms\Twin_ML510\ML510_0\fpga0\fpga0.xml
C:\Cerebrum\Platforms\Twin_ML510\ML510_0\fpga0\base_system.xml
C:\Cerebrum\Platforms\Twin_ML510\ML510_0\fpga0\system.ucf
C:\Cerebrum\Platforms\Twin_ML510\ML510_0\fpga0\bitgen_config.ut

C:\Cerebrum\Platforms\Twin_ML510\ML510_1\ML510_1.xml
C:\Cerebrum\Platforms\Twin_ML510\ML510_1\fpga0\fpga0.xml
C:\Cerebrum\Platforms\Twin_ML510\ML510_1\fpga0\base_system.xml
C:\Cerebrum\Platforms\Twin_ML510\ML510_1\fpga0\system.ucf
C:\Cerebrum\Platforms\Twin_ML510\ML510_1\fpga0\bitgen_config.ut
```

6.2 Platform/Board XML Contents

C:\Cerebrum\Platforms\Twin_ML510\Twin_ML510.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Platform>
       <Board ID="ML510_A" File="ML510_0.xml" />
       <Board ID="ML510_B" File="ML510_1.xml" />
</Platform>
```

C:\Cerebrum\Platforms\Twin_ML510\ML510_0\ML510_0.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Board>
   <FPGA ID="fpgaX" File="fpga0.xml">
       <BitGenScript
          Name="JTAG"
          File="bitgen_config.ut"
          Default="True" />
       <RequiredCore
          Source="io_unit_v1_00c"
          Type="io_unit"
          Version="1.00.c"
          DesignVisible="True">
       <!-- True Instance name will be FPGAID_Instance -->
       <Parameter
          Instance=" io_unit _0" />
       <CerebrumProperty
          Core="io_unit _0"
          Name="HardwareInterfaces"
          Value="IO_IN;IO_OUT;CLOCKS" />
       <Parameter
          Core="input_component_0"
          Name="C_WIDTH"
          Value="32" />
       ...
       </RequiredCore>
   </FPGA>
</Board>
```

C:\Cerebrum\Platforms\Twin_ML510\ML510_1\ML510_1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Board>
       <FPGA ID="fpgaY" File="fpga0.xml" />
</Board>
```

6.3 Cerebrum FPGA ID Generation

The IDs of the two FPGAs defined in the above platform would be:

```
ML510_A.fpgaX
ML510_B.fpgaY
```

Each of these corresponds to the Board IDs and FPGA IDs defined in the Platform and Board XML files respectively.