XN Specification

IN THIS DOCUMENT

1 Network Elements

xTIMEcomposer supports a single XN file that contains a single network definition. The network definition is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<Network xmlns="http://www.xmos.com"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.xmos.com http://www.xmos.com">
```

The XN hierarchy of elements is given in Figure 1

2 Declaration

A `Declaration` element provides a symbolic name for one or more xCORE Tiles. A single name or an array of names is supported with the form:

`tileref` *identifier*

`tileref` *identifier* [*constant-expression*]

An equivalent declaration is exported to the header file `<platform.h>` for use in XC programs. A `tileref` declaration is associated with physical xCORE tiles by the reference attribute of a `Tile` element (see §4.1).

Example

```
<Declaration>tileref master</Declaration>
<Declaration>tileref tile[8]</Declaration>
```

Node	Number	Description	Section
Network	1	An xCORE network	
Declarations	0+		
Declaration	1+	xCORE Tile declaration	§2
Packages	1+		
Package	1+	Device package	§3
Nodes	1		
Node	1+	Node declaration	§4
Tile	1+	An xCORE Tile	§4
Port	0+	An xCORE symbolic port name	§4.2
Boot	0 or 1	Boot method	§4.3
Source	1	Binary location	§4.4
Bootee	0+	Nodes booted	§4.5
RoutingTable	0 or 1		
Bits	1		
Bit	1+	Direction for bit	§4.6
Links	1		
Link	1+	Direction for link	§4.7
Service	0+	Service declaration	§4.8
Chanend	1+	Chanend parameter	§4.9
Links	0 or 1		
Link	1+	xCONNECT Link declaration	§5
LinkEndpoint	2	xCONNECT Link endpoint	§5.1
ExternalDevices	0 or 1		
Device	1+	External device	§6
Attribute	0+	A device attribute	§6.1
JTAGChain	0 or 1		
JTAGDevice	1+	A device in the JTAG chain	§7

Figure 1: XN hierarchy of elements

3 Package

A `Package` element refers to a package file that describes the mapping from xCORE ports and links to the pins on the package.

Attribute	Required	Type	Description
Id	Yes	String	A name for the package. All package names in the network must be unique.
Type	Yes	String	The name of the XML package. The tools search for the file *type*`.pkg` in the path specified by `XCC_DEVICE_PATH`.

Figure 2: XN `Package` element

Example

```
<Package id="L2" Type="XS1-L2A-QF124">
```

The package named `L2` is described in the file `XS1-L2A-QF124.xml`.

4 Node

A `Node` element defines a set of xCORE Tiles in a network, all of which are connected to a single switch. XMOS devices such as the G4 or L1 are both examples of nodes.

Attribute	Required	Type	Description
Id	No	String	A name for the node. All node names in the network must be unique.
Type	Yes	String	If type is `periph:XS1-SU` the node is a XS1-SU peripheral node. Otherwise the type specifies the name of an XML file that describes the node. The tools search for the file `config_`*type*`.xml` in the path specified by `XCC_DEVICE_PATH`.
Reference	Yes	String	Associates the node with a xCORE Tile indentifer specified in a `Declaration`. This attribute is only valid on nodes with type `periph:XS1-SU`.
RoutingId	No	Integer	The routing identifier on the xCONNECT Link network.
InPackageId	Yes	String	Maps the node to an element in the package file.
Oscillator	No	String	The PLL oscillator input frequency, specified as a number followed by either `MHz`, `KHz` or `Hz`.
OscillatorSrc	No	String	The name of the node which supplies the PLL oscillator input.
SystemFrequency	No	String	The system frequency, specified as a number followed by either `MHz`, `KHz` or `Hz`. Defaults to 400MHz if unset.
PllFeedbackDivMin	No	Integer	The minimum allowable PLL feedback divider. Defaults to 1 if unset.
ReferenceFrequency	No	String	A reference clock frequency, specified as a number followed by either `MHz`, `KHz` or `Hz`. Defaults to 100MHz if unset.
PllDividerStageOneReg	No	Integer	The PLL divider stage 1 register value.
PllMultiplierStageReg	No	Integer	The PLL multiplier stage register value.
PllDividerStageTwoReg	No	Integer	The PLL divider stage 2 register value.
RefDiv	No	Integer	`SystemFrequency` / `RefDiv` = `ReferenceFrequency`

Figure 3: XN `Node` element

The PLL registers can be configured automatically using the attributes System-Frequency, PllFeedbackDivMin and ReferenceFrequency, or can be configured manually using the attributes PllDividerStageOneReg, PllMultiplierStageReg, PllDividerStageTwoReg and RefDiv. If any of the first three attributes are provided, none of the last four attributes may be provided, and vice versa.

The PLL oscillator input frequency may be specifed using the Oscillator or OscillatorSrc attribute. If the Oscillator attribute is provided the OscillatorSrc attribute must not be provided, and vice versa.

If manual configuration is used, the attributes PllDividerStageOneReg, PllMultiplierStageReg, PllDividerStageTwoReg and RefDiv must be provided and the PLL oscillator input frequency must be specifed. The tools use these values to set the PLL registers and reference clock divider. Information on the PLL dividers can be found in xCORE frequency control documents for XS1-G processors (see X3221) and XS1-L processors (see X1433).

If the oscillator frequency is specifed and none of the manual PLL attributes are provided, automatic configuration is used. The tools attempt to program the PLL registers such that the target system frequency is achieved, the PLL feedback divider is greater than or equal to the minimum value and the target reference clock frequency is achieved. If any of these constraints cannot be met, the tools issue a warning and report the actual values used.

If the oscillator frequency is not specified, the tools do not attempt to configure the PLL. The PLL registers remain at their initial values as determined by the mode pins.

A network may contain either XS1-L devices or XS1-G devices, but not both.

Example

```
<Node Id="MyL1" Type="XS1-L1A" Oscillator="20Mhz"
      SystemFrequency="410MHz" ReferenceFrequency="98.5Mhz">
```

The node named MyL1 is an L1 device, as described in the file config_XS1-L1A.xml.

4.1 Tile

A Tile element describes the properties of a single xCORE Tile.

Attribute	Required	Type	Description
Number	Yes	Integer	A unique number for the tile in the node. Must be a value between 0 and *n*-1 where *n* is the number of tils as defined in the node's XML file.
Reference	No	String	Associates the tile with an identifier with the form tile[n] in a Declaration. A tile may be associated with at most one identifier.

Figure 4: XN Tileref element

Example

```
<Tile Number="0" Reference="tile[0]">
```

4.2 Port

A `Port` element provides a symbolic name for a port.

Attribute	Required	Type	Description
Location	Yes	String	A port identifier defined in the standard header file `<xs1.h>`. The ports are described in the XC manual (see X1009).
Name	Yes	String	A valid C preprocessor identifier. All port names declared in the network must be unique.

Figure 5: XN `Port` element

Example

```
<Port Location="XS1_PORT_1I" Name="PORT_UART_TX"/>
<Port Location="XS1_PORT_1J" Name="PORT_UART_RX"/>
```

4.3 Boot

A `Boot` element defines the how the node is booted. It contains one `Source` element (see §4.4) and zero or more `Bootee` elements (see §4.5) that are booted over xCONNECT Links. If the source specifies an xCONNECT Link, no `Bootee` elements may be specified. In a line of XS1-L devices, bootees must be contiguous to the device booting from SPI.

The XMOS tools require a `Boot` element to be able to boot programs from flash memory (see XM-000949-PC).

4.4 Source

A `Source` element specifies the location from which the node boots. It has the following attributes.

Attribute	Required	Type	Description
Location	Yes	String	Has the form `SPI:` or `LINK`. The `device-name` must be declared in the set of `Device` elements.

Figure 6: XN `Source` element

Only XMOS XS1-L devices can be configured to boot over xCONNECT Links.

Example

```
<Source Location="SPI:bootFlash"/>
```

4.5 Bootee

A Bootee element specifies another node in the system that this node boots via an xCONNECT Link. If more than one xCONNECT Link is configured between this node and one of its bootees (see §5 and §5.1), the tools pick one to use for booting.

Figure 7: XN Bootee element

Attribute	Required	Type	Description
NodeId	Yes	String	A valid identifier for another node.

Example

```
<Bootee NodeId="Slave">
```

4.6 Bit

A Bit element specifies the direction for messages whose first mismatching bit matches the specified bit number.

Figure 8: XN Bit element

Attribute	Required	Type	Description
number	Yes	Integer	The bit number, numbered from the least significant bit.
direction	Yes	Integer	The direction to route messages.

Example

```
<Bit number="1" direction="0"/>
```

4.7 Link

When it appears within a RoutingTable element, a Link element specifies the direction of an xCONNECT Link.

Figure 9: XN Link element

Attribute	Required	Type	Description
name	Yes	String	A link identifier in the form X*n*L*m* where *n* denotes a tile number and *m* the link letter. See the corresponding package datasheet for available link pinouts.
direction	Yes	Integer	The direction of the link.

Example

```
<Link number="XLA" direction="2"/>
```

4.8 Service

A `Service` element specifies an XC service function provided by a node.

Figure 10: XN `Service` element

Attribute	Required	Type	Description
Proto	Yes	String	The prototype for the service function, excluding the `service` keyword. This prototype is exported to the header file `<platform.h>` for use in XC programs.

Example

```
<Service Proto="service_function(chanend c1, chanend c2)">
```

4.9 Chanend

A `Chanend` element describes a channel end parameter to an XC service function.

Figure 11: XN `Service` element

Attribute	Required	Type	Description
Indentifier	Yes	String	The identifier for the chanend argument in the service function prototype.
end	Yes	Integer	The number of the channel end on the current node.
remote	Yes	Integer	The number of the remote channel end that is connected to the channel end on the current node.

Example

```
<Chanend Identifier="c" end="23" remote="5"/>
```

5 Link

xCONNECT Links are described in the system specification documents (XS1-G: X7507, XS1-L: X1151) and link performance documents (XS1-G: X7561, XS1-L: X2999).

A Link element describes the characteristics of an xCONNECT Link. It must contain exactly two `LinkEndpoint` children (see §5.1).

Attribute	Required	Type	Description
Encoding	Yes	String	Must be either `2wire` or `5wire`.
Delays	Yes	String	Of the form *x*,*y* where *x* specifies the inter delay value for the endpoint, and *y* specifies the intra delay value for the endpoint. If a value for *y* is omitted, *x*,`1` is used. If both values are omitted, `1,1` is used.
Flags	No	String	Specifies additional properties of the link. Use the value `XSCOPE` to specify a link used to send XScope trace information.

Figure 12: XN `Link` element

Example

```
<Link Encoding="2wire" Delays="4,4">
```

5.1 LinkEndpoint

A `LinkEndpoint` describes one end of an xCONNECT Link, the details of which can be found in the system specification documents (XS1-G: X7507, XS1-L: X1151). Each endpoint associates a node identifier to a physical xCONNECT Link.

Attribute	Required	Type	Description
NodeID	No	String	A valid node identifier.
Link	No	String	A link identifier in the form X*n*L*m* where *n* denotes a tile number and *m* the link letter. See the corresponding package datasheet for available link pinouts.
RoutingId	No	Integer	The routing identifier on the xCONNECT Link network.
Chanend	No	Integer	A channel end.

Figure 13: XN `LinkEndpoint` element

An endpoint is usually described as a combination of a node identifier and link identifier. For a streaming debug link, one of the endpoints must be described as a combination of a routing identifier and a channel end.

Example

```
<LinkEndpoint NodeId="0" Link="X0LD"/>
<LinkEndpoint RoutingId="0x8000" Chanend="1">
```

6 Device

A Device element describes a device attached to an xCORE Tile that is not connected directly to an xCONNECT Link.

Attribute	Required	Type	Description
Name	Yes	String	An identifier that names the device.
NodeId	Yes	String	The identifier for the node that the device is connected to.
Tile	Yes	Integer	The tile in the node that the device is connected to.
Class	Yes	String	The class of the device.
Type	No	String	The type of the device (class dependent).

Figure 14: XN Device element

xTIMEcomposer recognizes the Class SPIFlash and use the Type attribute to identify the model of the flash device.

6.1 Attribute

An Attribute element describes one aspect of a Device (see §6).

Attribute	Required	Type	Description
Name	Yes	String	Specifies an attribute of the device.
Value	Yes	String	Specifies a value associated with the attribute.

Figure 15: XN Attribute element

xTIMEcomposer supports the following attribute names for the device class SPIFlash:

PORT_SPI_MISO	SPI Master In Slave Out signal.
PORT_SPI_SS	SPI Slave Select signal.
PORT_SPI_CLK	SPI Clock signal.
PORT_SPI_MOSI	SPI Master Out Slave In signal.

Example

```
<Attribute Name="PORT_SPI_MISO" Value="PORT_SPI_MISO"/>
```

7 JTAGDevice

xTIMEcomposer loads and debugs programs on target hardware using JTAG. The JTAGChain element describes a device in the JTAG chain. The order of these elements defines their order in the JTAG chain.

Figure 16: XN `JTAGDevice` element

Attribute	Required	Type	Description
NodeID	Yes	String	A valid node identifier.

Example

```
<!-- N1 comes before N2 in the JTAG chain -->
<JTAGDevice NodeId="N1">
<JTAGDevice NodeId="N2">
```