Describe a target platform

Hardware platforms are described using XN. An XN file provides information to the XMOS compiler toolchain about the target hardware, including XMOS devices, ports, flash memories and oscillators.

The XMOS tools use the XN data to generate a platform-specific header file `<platform.h>`, and to compile, boot and debug multi-node programs.

1 Supported network topologies

To route messages across the xCONNECT Link network, the routing ID and routing table of each node on the network must configured. The tools use the information in the XN file to setup the routing for the network before running the application.

If the routing configuration is explicitly specified in the XN file, the tools use this configuration. If the routing configuration is omitted from the XN file the tools choose a suitable set routing IDs and routing tables based on the network topology. The tools can automatically compute routing configurations for the the following network topologies.

Figure 1: Topologies that can be automatically routed

Network Topology	Supported Configurations
Line	Not supported on XS1-G devices
Hypercube	Degree-2 (pair of nodes) Degree-3 (ring of 4 nodes) Degree-3 (cube of 8 nodes) Degree-4 (canonical cube of 16 nodes)
Hypercube with lines attached	Not supported on XS1-G devices

2 A board with two packages

Figure 2 illustrates a board containing two XMOS L8-64 devices arranged in a line. A suitable XN description is described below.

An XN file starts with an XML declaration.

```
<?xml version="1.0" encoding="UTF-8"?>
```

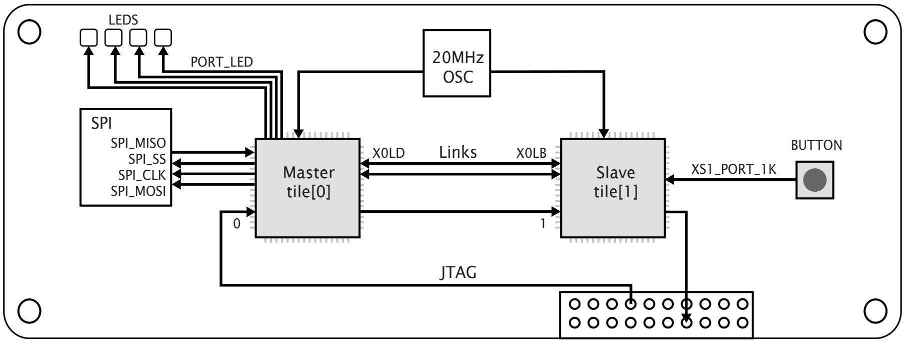

Figure 2: Example hardware platform

The following code provides the start of the network.

```
<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 following code declares two xCORE Tiles. The declaration “`tileref tile[2];`” is exported to the header file `<platform.h>`.

```
<Declarations>
  <Declaration>tileref tile[2]</Declaration>
</Declarations>
```

The following code declares a package named `P1`, which contains a single node named `Master`.

```
<Packages>
  <Package Id="P1" Type="XS1-LnA-64-TQ128">
    <Nodes>
      <Node Id="Master" Type="XS1-L8A-64" InPackageId="0"
            Oscillator="20MHz" SystemFrequency="400MHz">
      <Boot>
        <Source Location="SPI:bootFlash"/>
        <Bootee NodeId="Slave" Tile="0"/>
      </Boot>
      <Tile Number="0" Reference="tile[0]">
        <Port Location="XS1_PORT_1A" Name="PORT_SPI_MISO"/>
        <Port Location="XS1_PORT_1B" Name="PORT_SPI_SS"/>
        <Port Location="XS1_PORT_1C" Name="PORT_SPI_CLK"/>
        <Port Location="XS1_PORT_1D" Name="PORT_SPI_MOSI"/>
        <Port Location="XS1_PORT_4A" Name="PORT_LED"/>
      </Tile>
    </Node>
  </Nodes>
</Package>
```

The node `Master` is a 400MHz XS1-L8A-64 device in a TQ128 package, clocked by a 20MHz oscillator. It is booted from an SPI device named “bootFlash” which has the class “SPIFlash”.

The declaration of tile “0” is associated with `tile[0]` and the ports 1A, 1B, 1C, 1D and 4A are given symbolic names. These declarations are exported to the header file `<platform.h>`.

The following code declares a package named `P2`, which contains a single node named `Slave`.

```
  <Package Id="P2" Type="XS1-LnA-64-TQ128">
    <Nodes>
      <Node Id="Slave" Type="XS1-L8A-64" InPackageId="0"
            Oscillator="20Mhz" SystemFrequency="400MHz">
        <Boot>
          <Source Location="LINK"/>
        </Boot>
        <Tile Number="0" Reference="tile[1]">
          <Port Location="XS1_PORT_1K" Name="PORT_BUTTON"/>
        </Tile>
      </Node>
    </Nodes>
  </Package>
</Packages>
```

The node `Slave` is a 400MHz XS1-L8A-64 device in a TQ128 package, clocked by a 20MHz oscillator. It is booted from node `Master` over an xCONNECT Link.

The following code defines a 2-wire xCONNECT Link with, which connects the node `Master` on link X0LD to the node `Slave` on link `X0LB`.

```
<Links>
  <Link Encoding="2wire" Delays="4,4">
    <LinkEndpoint NodeId="Master" Link="X0LD"/>
    <LinkEndpoint NodeId="Slave" Link="X0LB"/>
  </Link>
</Links>
```

The links have intra-symbol and inter-symbol delays of 4 clock periods.

The following code specifies a list of components on the board that are connected to XMOS devices.

```
<ExternalDevices>
  <Device NodeId="Master" Tile="0" Name="bootFlash"
          Class="SPIFlash" Type="AT25FS010">
     <Attribute Name="PORT_SPI_MISO" Value="PORT_SPI_MISO"/>
     <Attribute Name="PORT_SPI_SS"   Value="PORT_SPI_SS"/>
     <Attribute Name="PORT_SPI_CLK"  Value="PORT_SPI_CLK"/>
     <Attribute Name="PORT_SPI_MOSI" Value="PORT_SPI_MOSI"/>
   </Device>
 </ExternalDevices>
```

A device named `bootFlash` is connected to xCORE Tile 0 on Node `Master`, and is given attributes that associate the four SPI pins on the device with ports. (The class `SPIFlash` is recognized by XFLASH.)

The following code describes the JTAG scan chain.

```
  <JTAGChain>
    <JTAGDevice NodeId="Master" Position="0"/>
    <JTAGDevice NodeId="Slave" Position="1"/>
  </JTAGChain>

</Network>
```