XSIM Command-Line Manual

IN THIS DOCUMENT

- Overall Options
- Warning Options
- Tracing Options
- Loopback Plugin Options

XSIM performs a cycle-based simulation of an XMOS Executable (XE) file. The XE file contains a description of the target hardware.

1 Overall Options

xe-file Specifies an XE file to simulate.

`--max-cycles` *n*
Exits when *n* system cycles is reached.

`--plugin` *name args*
Loads a plugin DLL. The format of *args* is determined by the plugin; if *args* contains any spaces, it must be enclosed in quotes.

`--stats` On exit, prints the following:

- A breakdown of the instruction counts for each logical core.
- The number of data and control tokens sent through the switches.

`--help` Prints a description of the supported command line options.

`--version` Displays the version number and copyrights.

2 Warning Options

`--warn-resources`
Prints (on standard error) warning messages for the following:

- A timed input or output operation specifies a time in the past.
- The data in a buffered port's transfer register is overwritten before it is input by the processor.

Publication Date: 2012/10/22 Document Number: X1415D

`--warn-stack`

Turns on warnings about possible stack corruption.

XSIM prints a warning if one XC task attempts to read or write to another task's workspace. This can happen if the stack space for a task is specified using either `#pragma stackfunction` (see XM-000959-PC) or `#pragma stackcalls` (see XM-000959-PC).

`--no-warn-registers`

Don't warn when a register is read before being written.

3 Tracing Options

`--trace`
`-t`

Turns on instruction tracing for all tiles (see Figure 1).

`--trace-to` *file*

Turns on instruction tracing for all tiles. The trace is output to *file*.

`--disable-rom-tracing`

Turns off tracing for all instructions executed from ROM.

`--enable-fnop-tracing`

Turns on tracing of FNOP instructions.

Figure 1: Trace output for XS1 processors

Tile	Core State											Address		**Instruction**		**Mem**	**Cycle**
Name	I_0	I_1	I_2	$S_0S_1(T_0)$ .. $S_0S_1(T_n)$			.	M	S	K	N	PC	(sym + offset) :	name	operands	address	@val
from XN	-	*	-	-	-	*n* status pairs		-	-	-	-				val	L[adr]	
	D	P	d	a	b			m	s	k	n				r*n* (val)	S[adr]	
				A	i										res[id]		
				i	e												
				I													
				p													
				m													
				s													
				w													

I_0: -	No debug interrupt	S_1: -	Interrupts and events disabled
I_0: D	Instruction caused debug interrupt	S_1: b	Interrupts and events enabled
I_1: *	Instruction excepted	S_1: i	Interrupts enabled and events disabled
I_1: P	Instruction paused	S_1: e	Interrupts disabled and events enabled
I_2: -	Not in debug mode	M: -	MSYNC not set
I_2: d	Tile in debug mode	M: m	MSYNC set
S_0: -	Core not in use	S: -	SSYNC not set
S_0: a	Core active	S: s	SSYNC set
S_0: A	Core active (the instruction being traced belongs to this core)	K: -	INK not set
S_0: i	Core active with ININT bit set	K: k	INK set
S_0: I	Core active with ININT bit set (belongs to this core)	N: -	INENB not set
S_0: p	Core paused due to instruction fetch	N: n	INENB set
S_0: m	Core paused with MSYNC bit set	r*n* (val)	Value of register *n*
S_0: s	Core paused withSSYNC bit set	res[id]	Resource identifier
S_0: w	Core paused with WAITING bit set	L/S[adr]	Load from/Store to address

`--vcd-tracing` *args*

Enables signal tracing. The trace data is output in the standard VCD file format.

If *args* contains any spaces, it must be enclosed in quotes. Its format is:

global-options$_{opt}$⟨`-tile name` ⟨trace-options⟩*⟩*

The global options are:

`-pads` Turns on pad tracing.

`-o` *file* Places output in *file*.

The trace options are specific to the tile associated with the XN core declaration name, for example `tile[0]`.

The trace options are:

`-ports` Turns on port tracing.

`-ports-detailed` Turns on more detailed port tracing.

`-cycles` Turns on clock cycle tracing.

`-clock-blocks` Turns on clock block tracing.

`-cores` Turns on logical core tracing.

`-instructions` Turns on instruction tracing.

To output traces from different nodes, tiles or logical cores to different files, this option can be specified multiple times.

For example, the following command configures the simulator to trace the ports on tile[0] to the file trace.vcd.

```
▶ xsim a.xe --vcd-tracing "-o trace.vcd -tile tile[0] -ports"
```

Tracing by the VCD plugin can be enabled and disabled using the `_traceStart()` and `_traceStop()` syscalls, for example:

```
#include <xs1.h>
#include <syscall.h>

port p1 = XS1_PORT_1A;

int main() {
   _traceStop();

   p1 <: 1;
   p1 <: 0;

   _traceStart();
   p1 <: 1;
```

```
    p1 <: 0;
    _traceStop();

    p1 <: 1;
    p1 <: 0;

    return 0;
}
```

4 Loopback Plugin Options

The XMOS Loopback plugin configures any two ports on the target platform to be connected together. The format of the arguments to the plugin are:

`-pin` *package pin*

Specifies the pin by its name on a package datasheet. The value of *package* must match the `Id` attribute of a `Package` node (see XM-000929-PC) in the XN file used to compile the program.

`-port` *name n offset*

Specifies *n* pins that correspond to a named port.

The value of *name* must match the `Name` attribute of a `Port` node (see XM-000929-PC) in the XN file used to compile the program.

Setting *offset* to a non-zero value specifies a subset of the available pins.

`-port` *tile p n offset*

Specifies *n* pins that are connected to the port *p* on a *tile*.

The value of *tile* must match the `Reference` attribute of a `Tile` node (see XM-000929-PC) in the XN file used to compile the program.

p can be any of the port identifiers defined in `<xs1.h>`. Setting *offset* to a non-zero value specifies a subset of the available pins.

The plugin options are specified in pairs, one for each end of the connection. For example, the following command configures the simulator to loopback the pin connected to port `XS1_PORT_1A` on `tile[0]` to the pin defined by the port `UART_TX` in the program.

```
▶ xsim uart.xe --plugin LoopbackPort.dll '-port tile[0] XS1_PORT_1A 1 0 -port UART_TX
  1 0'
```

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the “Information”) and is providing it to you “AS IS” with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.