XCC Target-Dependent Behavior for XS1 Devices

IN THIS DOCUMENT

This section describes behavior of the XMOS compiler collection that is specific to the XS1 architecture.

1 Support for Clock Blocks

An XS1 device provides a single reference clock that ticks at a frequency derived from an external oscillior. XC requires the system designer to ensure that the reference clock ticks at 100MHz for correct operation of timers.

Each xCORE Tile provides a set of programmable clock blocks, which can be used to produce clock signals for ports. A clock block can use either a 1-bit port or a divided reference clock.

The `<xs1.h>` header file includes a `clock` type definition. A variable of type `clock`, not declared `extern`, must be initialized with an expression representing a clock block, for example:

```
clock c = XS1_CLKBLK_1;
```

The number of clock blocks available is given in the device datasheet. Their names are as the above declaration, numbered sequentially from 1.

In XC, the `clock` type is a *resource* type, with the following additional rules:

- A structure may declare members of type `clock`. Variables of a structure with type `clock` may be declared only as external declarations.
- A variable declaration prefixed with `on` may declare an object of type `clock`.
- Automatic variables may not be declared with type `clock`.

Publication Date: 2013/11/11 REV B

2 Support for Ports

The XC port declaration

```
port p;
```

declares a raw port. On XS1 devices, all ports used for inputting and outputting data are clocked by a 100MHz reference clock (see §1) and use a single-entry buffer, even if their declarations are not qualified with the keyword `buffered`.

The table in Figure 1 can be used to determine which I/O operations are supported on XS1 ports, depending on whether or not the corresponding XC declaration is qualified with the keyword `buffered`.

Figure 1: I/O operations supported on XS1 ports

Mode	Operation		
	Serialization	Strobing	`@ when`
Unqualified	✗	✗	✗
`buffered`	✓	✓	✓

The compiler detects and issues errors in the following cases:

- Serialization: A port not qualified with `buffered` is declared with a transfer width different from the port width.
- Strobing: A port not qualified with `buffered` is configured to use a ready-in or ready-out signal.
- An input uses both `@` and `when`: Both of these operators are used in an input statement with a port whose declaration is not qualified with `buffered`.

2.1 Serialization

Note that if serialization is used, the time specified by a timed input statement records the time at which the *last* bits of data are sampled. This can result in unexpected behaviour when serialization is used, since the construction

```
par {
  p @ t <: x;
  q @ t :> y;
}
```

causes the output on `p` to start at the same time as the input on `q` completes. To input and output this data in parallel, the input time should be offset in the software by an amount equal to the the transfer width divided by the port width.

2.2 Timestamping

The timestamp recorded by an input statement may come after the time when the data was sampled. This is because the XS1 provides separate instructions for

inputting data and inputting the timestamp, so the timestamp can be input after the next data is sampled. This issue also affects output statements, but does not affect inputs performed in the guards of a `select` statement. The compiler inputs the timestamp immediately after executing an input or output instruction, so in practice this behaviour is rarely seen.

2.3 Changing Direction of Buffered Ports

An attempt to change the direction of a port qualified with `buffered` results in undefined behaviour.

3 Channel Communication

On some revisions of the XS1 architecture, it is not possible to input data of size less than 32 bits from a streaming channel in the guard of a select statement.

- When compiling for the XS1-G architecture, the compiler disallows selecting on a channel input of less than a word-length in an XC streaming channel. The command line option `-fsubword-select` relaxes this restriction, but this can lead to cases with these functions not being taken even if data is available on the channel.
- When compiling for the XS1-G architecture, the `inuchar_byref`, `inct_byref` and `testct` functions may not be used in an XC `select` statement. The command line option `-fsubword-select` relaxes this restriction, but this can lead to cases with these functions not being taken even if data is available on the channel.