

## XS1 Data Types

---

The size and alignment of C and XC's data types are not specified by the language. This allows the size of `int` to be set to the natural word size of the target device, ensuring the fastest possible performance for many computations. It also allows the alignment to be set wide enough to enable efficient memory loads and stores. Figure 1 represents the size and alignment of the data types specified by the xCORE Application Binary Interface (see [XM-000967-PC](#)), which provides a standard interface for linking objects compiled from both C and XC.

Data Type	Size (bits)	Align (bits)	Supported		Meaning
			XC	C	
<code>char</code>	8	8	✓	✓	Character type
<code>short</code>	16	16	✓	✓	Short integer
<code>int</code>	32	32	✓	✓	Native integer
<code>long</code>	32	32	✓	✓	Long integer
<code>long long</code>	64	32	✗	✓	Long long integer
<code>float</code>	32	32	✗	✓	32-bit IEEE float
<code>double</code>	64	32	✗	✓	64-bit IEEE float
<code>long double</code>	64	32	✗	✓	64-bit IEEE float
<code>void *</code>	32	32	✗	✓	Data pointer
<code>port</code>	32	32	✓	✗	Port
<code>timer</code>	32	32	✓	✗	Timer
<code>chanend</code>	32	32	✓	✗	Channel end

---

**Figure 1:**  
Size and  
alignment of  
data types on  
XS1 devices

---

In addition:

- ▶ The `char` type is by default unsigned.
- ▶ The types `char`, `short` and `int` may be specified in a bit-field's declaration.
- ▶ A zero-width bit-field forces padding until the next bit-offset aligned with the bit-field's declared type.
- ▶ The notional transfer type of a port is `unsigned int` (32 bits).
- ▶ The notional counter type of a port is `unsigned short` (16 bits).
- ▶ The notional counter type of a timer is `unsigned int` (32 bits).