Connecting Memory and Peripheral Devices to a 32-bit Processor

Summary

This application note explores the various methods available for connection of physical memory and peripheral I/O devices to a 32-bit processor.

Physical memory is connected to the processor’s External Memory interface. Peripheral devices are connected to the processor’s Peripheral I/O interface. The following sections explore the methods of connection, from a single physical memory device, through to a range of memory and peripheral devices – accessed by single or multiple processors.

Simple Memory Connection

At the most basic level, static, dynamic, or block memory can be connected to the processor through the use of a Memory Controller peripheral device (WB_MEM_CTRL), which can be configured as either a Static RAM Controller (SRAM), Synchronous Dynamic RAM Controller (SDRAM), or a Block RAM Controller (BRAM). Figure 1 illustrates the use of such a device to interface to the SRAM located on a Daughter Board. Although not required, a Wishbone Interconnect device (WB_INTERCON) has been used to connect from the processor to the Memory Controller, due to its convenience in terms of wiring and handling of the address line mapping.

Figure 1. Interfacing to static RAM devices using an SRAM-configured Memory Controller.

For further information on the Memory Controller peripheral, refer to the WB_MEM_CTRL Configurable Wishbone Memory Controller core reference.

A Word on Connecting External Memory...

When connecting to physical memory outside of the FPGA device, the wiring involved will depend on the type of memory and where it is located.

If the memory is located on a NanoBoard or a plug-in daughter board, simply place the appropriately-configured Memory Controller, wired to the relevant port component on the one side and a Wishbone Interconnect device on the other. The latter is then connected to the processor’s External Memory interface. Figure 1 showed an example of connecting to the SRAM on a daughter board.

Note: Port components can be found in the available port-plugin integrated libraries, located in the \Library\Fpga folder of the installation. The particular library used will depend on the type of NanoBoard you are using. For a NanoBoard-NB1, the relevant components can be found in FPGA NB1 Port-Plugin.IntLib. For a Desktop NanoBoard NB2DSK01, the relevant components can be found in FPGA DB Common Port-Plugin.IntLib.
Connecting Memory and Peripheral Devices to a 32-bit Processor

If the memory device resides elsewhere – for example on a production board or a third party development board – and the memory device is of a type supported by Altium Designer's Wishbone Memory Controller (SRAM, SDRAM, BRAM) – then instead of a port component, you will need to place and wire up ports corresponding to the I/O lines to/from the memory device. Figure 2 shows an example of this.

Figure 2. Interfacing to external memory non-resident on Altium hardware.

If the memory device is not of a type supported by the Wishbone Memory Controller, you would need to write your own VHDL to effectively provide interface control between the Wishbone Interconnect and the external memory device. The connectivity would be similar to that of Figure 2, but with the WB_MEM_CTRL replaced by a sheet symbol referencing the entity in your VHDL code.
Connecting Multiple Memory Devices

The nature of your design may warrant the use of several memory devices, possibly of differing type, each of which requires connection to the processor's single Wishbone External Memory interface. This can be readily achieved through the use of a Wishbone Interconnect device. This device connects directly to the processor's External Memory interface and, through its configurable nature, provides multiplexed access to multiple slave memory devices.

Figure 3 illustrates the use of a Wishbone Interconnect device to connect to SRAM devices on a Daughter Board, one of the SRAM devices on the NanoBoard and a dedicated single-port block of RAM within the design. In each case, the respective Memory Controller device (configured as either SRAM or BRAM Controller) sits between the Wishbone Interconnect device and the physical memory device(s).

For further information on the Wishbone Interconnect peripheral, refer to the WB_INTERCON Configurable Wishbone Interconnect core reference.
Connecting Multiple Peripheral Devices

Typically in a design, the processor will need to interface to multiple Wishbone-compliant peripherals (slave devices). Each of these peripherals may contain any number of internal registers with which to write to/read from. It is not possible to communicate directly, and simultaneously, with each of these slave devices. A means of multiplexing must be used, allowing the processor to talk to any number of slaves over the one interface.

In the same way that multiple memory devices are connected to the processor’s External Memory interface, this involves the use of a configurable Wishbone Interconnect device, as illustrated in the example image of Figure 4.

Figure 4. Multiplexing a 32-bit processor’s peripheral I/O interface using a Wishbone Interconnect device.

In this example circuit, the Wishbone Interconnect peripheral (WB_INTERCON) enables a single 32-bit processor (in this case a TSK3000A) to communicate with three Wishbone-compliant peripheral devices (a 1x8 Parallel Port Unit, a Serial Port Unit and a PS2 Controller).
**Dual-Mastering**

Some designs may require shared access to one or more slave memory or peripheral devices. This can be achieved by using a configurable Wishbone Dual Master device (WB_DUALMASTER). Figure 5 shows an example of using this device to connect two 32-bit processors – a PPC405CR and a TSK3000A – to the Static RAM located on a Daughter Board. Wishbone Interconnect devices have again been used for wiring/addressing convenience.

![Figure 5. Sharing a single memory device between processors using a Wishbone Dual Master device](image)

The WB_DUALMASTER allows you to define how the two Wishbone Masters contest for the slave device – either simple "Round-Robin" arbitration, giving both Masters equal access to the slave, or "Priority", whereby one Master is assigned higher priority than the other. In terms of memory access, the latter would be typically used in a design where a memory device is shared between a processor and a memory-based peripheral, such as a VGA Controller. In such a design, the VGA Controller would be given higher priority with respect to memory access, otherwise effects such as screen flicker could be prevalent.

By placing a Wishbone Interconnect device after the Wishbone Dual Master device, it is possible to connect two processor masters to a whole bank of slave memory or peripheral devices. The devices would be mapped into the respective processor address spaces at identical locations. Figure 6 shows an example of using both a Wishbone Dual Master device and a Wishbone Interconnect device, to allow two 32-bit processors (TSK3000As) to access a variety of physical slave memory devices.
Connecting Memory and Peripheral Devices to a 32-bit Processor

Figure 6. Sharing multiple memory devices between 32-bit processors using a Wishbone Dual Master device.

For further information on the Wishbone Dual Master peripheral, refer to the WB_DUALMASTER Configurable Wishbone Dual Master core reference.

Sharing Peripheral Devices

Although the Wishbone Dual Master device can be used to share peripheral devices between two processors, it cannot provide marshalling for interrupts from the peripherals through to the processors. If the peripheral devices being shared do not generate interrupts, or they are not being used, then use of a Wishbone Dual Master – in series with a Wishbone Interconnect – is fine.

Figure 7 shows an example of two 32-bit processors sharing access to three different slave peripheral devices (two parallel port units and an Ethernet Media Access Controller). The port units do not generate interrupts and the interrupt from the EMAC is not being used.
Connecting Memory and Peripheral Devices to a 32-bit Processor

Figure 7. Sharing multiple slave peripheral devices between 32-bit processors.

If you need to share peripherals between processors, then you should consider using a Wishbone Multi-Master device. It can be configured to have between two and eight masters, but more importantly, it also has support for interrupt channeling. For more information, see the next section.

Multi-Mastering

Your design might feature one or more processors and a range of memory-based peripheral devices requiring access to the same physical memory. Consider for example, the circuit fragment in Figure 8.

Figure 8. Sharing physical memory between more than two masters using two WB_DUALMASTER devices.
Connecting Memory and Peripheral Devices to a 32-bit Processor

Although not shown, the same physical SRAM is accessed by one 32-bit processor (a Nios II) and two peripheral devices (a BT656 Video Capture Controller and a VGA32_16BPP Controller). To facilitate this shared access, two Wishbone Dual Master devices have been used. The interfacing to memory can be further simplified by using a Wishbone Multi-Master device (WB_MULTIMASTER).

The WB_MULTIMASTER is similar to the WB_DUALMASTER, but with three essential differences:

- It can facilitate the connection of up to 8 Wishbone Masters.
- It allows you to specify one Wishbone Master to be granted instant access to the bus when the WB_MULTIMASTER is ‘idle’. This reduces latency as the nominated master experiences no delay in acquiring access.
- It can pass interrupts from a connected Wishbone Interconnect, through to all connected 32-bit processors. This makes it ideal for use on the peripheral side, when multiple 32-bit processors require shared access to a block of peripheral devices, and one or more of those devices generate interrupts.

Figure 9 shows the same circuit fragment as Figure 8, but with the two Wishbone Dual Master devices replaced by a single Wishbone Multi-Master device.

![Diagram of sharing physical memory between more than two masters using a single WB_MULTIMASTER device.](image)

For more information on the Wishbone Multi-Master peripheral, refer to the [WB_MULTIMASTER Configurable Wishbone Multi-Master core reference](#).

---

8 AP0148 (v2.0) February 27, 2008
Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version No.</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-Jul-2006</td>
<td>1.0</td>
<td>Initial release</td>
</tr>
<tr>
<td>01-Sep-2006</td>
<td>1.1</td>
<td>Fixed Address sizing for Intercon device in Figure 2.</td>
</tr>
<tr>
<td>27-Feb-2008</td>
<td>2.0</td>
<td>Updated for Altium Designer Summer 08</td>
</tr>
</tbody>
</table>