A

Address Decoding  Address-decoding logic produces chip-select signals for each peripheral based upon the input address and the position of the peripheral in the address map.

Address Map  A list of addresses for each of the peripherals connected to a master in a system module.

Arbitrator  Also referred to as “arbiter”. A logic block that is integrated with each slave that has multiple masters. When multiple masters select the same slave, the arbitrator selects which master gains access to the slave.

Avalon Bus Module  The Avalon Bus Module is the backbone of a system module. It is the main path of communication between peripherals components in an SOPC design. The Avalon Bus Module is the sum of all control, data and address signals, and arbitration logic that connect together the peripheral components making up the system module. The Avalon Bus Module implements a configurable bus architecture, which changes to fit the interconnection needs of the designer’s peripherals. The Avalon Bus Module is very rarely used as a discrete unit, because the SOPC Builder will almost always be used to automate system integration. The designer’s view of the Avalon Bus Module usually is (and should be) limited to the specific ports that relate to the connection of custom peripherals. Note that the Avalon Bus Module (an Avalon “Bus”) is a unit of active logic that takes the place of passive, metal bus lines on a physical PCB. In this context, the ports of the Avalon Bus Module could be thought of as the pin connections for all peripheral devices connected to a passive bus. It is natural that the Avalon Bus Module should have a large number of ports, allowing for all peripherals to connect to the “bus.” The Avalon Bus Specification defines only the ports and timing interface to the Avalon Bus Module. It does not specify any electrical or physical characteristics of a physical bus. The Avalon Bus Module provides the following services to peripherals that connect to it:

- Data-path Multiplexing
- Address Decoding
- Wait-state Generation
- Dynamic Bus Sizing
### Nios Glossary

- **Interrupt-Priority**
- **Latent Read**
- **Streaming Read and Write.**

**Avalon Peripherals**  An Avalon peripheral is a logical device—either on-chip or off-chip—which performs some system-level task, and communicates with other system components through the Avalon Bus Module. Peripherals are self-contained system components, and may be added or removed at design time, depending on the requirements of the system. Avalon peripherals can be traditional peripheral components, such as a UART, PIO, bus bridge, keyboard, mouse, and so on, as well as memories and processors. User logic can also be an Avalon peripheral, as long as it provides address, data, control, and status signal interfaces to the Avalon bus module in accordance to the Avalon Bus Specification. A peripheral connects to the ports on the Avalon Bus Module allocated for that peripheral. The peripheral may also have user-defined ports in addition to the Avalon address, data, control, and status signals. These signals connect to custom logic external to the system module.

**B**

**Bus Cycle**  A bus cycle is a basic unit of one clock period, which is defined from rising-edge to rising-edge of the Avalon clock. Bus signal timing is referenced to the bus cycle clock.

**Bus Transfer**  An Avalon bus transfer is a read or write operation of data, which may take one or more bus cycles. The transfer widths supported by Avalon include byte (8-bit), halfword (16-bit), and word (32-bit). The transfer lengths (that is, the number of bytes, halfwords, or words that are transferred during a transaction) is peripheral dependent.

**C**

**Control Signals**  Signals that are generated by master ports to access a slave port. These signals may vary with the type of peripheral connected to the master port, but they typically include an address bus and read enable and write enable signals.

**Custom Instruction**  The ability to add a custom block of logic that performs a new operation on register data as an opcode instruction in the Nios processor. These logic blocks can be combinatoric logic or multi-cycle logic and can be prefixed to provide 11-bits of sub-pcodes.
D

Data Path Multiplexing  Multiplexers in the Avalon Bus Module transfer data from the selected slave peripheral to the appropriate master peripheral. This connection between all appropriate peripherals is the fundamental service of the Avalon Bus Module.

DMA  Direct Memory Access. A method of connecting a peripheral to a memory peripheral to do a block transfer into or out of the memory peripheral without the intervention of a processor.

Dynamic Bus Sizing  Dynamic Bus Sizing hides the details of interfacing narrow peripherals to a wider Avalon bus. Dynamic Bus Sizing automatically executes multiple bus transactions as required to fetch (or store) wide data values from (or to) narrow peripherals.

I

Interrupt Priority Assignment  When one or more slave peripherals generate interrupts, the Avalon Bus Module passes the (prioritized) interrupts to appropriate master peripherals, along with the appropriate interrupt request (IRQ) number.

L

Latency Aware Peripherals  Peripherals that have ports supporting latent reads.

Latent Read  “Latent read” refers to bus transactions with slave peripherals that require a variable number of wait states to present data. The slave peripheral can stall the master for several latency cycles while it fetches the requested data.

M

Master Port  Any peripheral port that can initiate a bus transaction. Masters present control signals to initiate read and write transactions.

Master-Slave Pair  A master and slave port that are connected via the SMM Avalon Bus. Arbitration assignments are made to specific master-slave pairs in the SOPC Builder.
N

Nios Configuration Wizard  The part of the SOPC Builder where you build the Nios CPU (contains three sections, Architecture, Configuration, and Custom Instructions)

Nios System Module  See System Module.

O

Opcode  The field that denotes the operation and format of a microprocessor instruction.

P

Peripheral Inside the System Module  If SOPC Builder is aware of a peripheral in a peripheral library, then it can automatically connect the peripheral to the Avalon bus module and treat the peripheral as a piece of the system module. Such a peripheral is referred to as a peripheral inside the system module. The details of connecting the address, data and control ports to the Avalon bus module are hidden from the user. Any additional ports on the peripheral are presented to the outside world as ports on the system module. These ports may connect directly to physical device pins, or may connect to the ports of other on-chip modules. SOPC Builder offers a library of fundamental peripherals, such as a UART, PIO, interval timer, etc, from which a system designer can choose.

Peripheral Outside the System Module  An Avalon Bus peripheral can exist external to the system module. For example, the peripheral may exist physically outside the PLD, or the peripheral be a separate block of logic that is connected to ports on the system module. For peripherals outside the system module, the appropriate Avalon Bus Module signals are presented as ports on the system module. Adding a “User-defined Interface” in the SOPC Builder specifies a peripheral outside the system module. The User-defined Interface lets the designer specify all settings which affect the Avalon bus interface for that peripheral, such as master/slave relationship, wait states, interrupt generation, and so on.

Ports  A port is a point of connection on an HDL module, used to propagate signals into or out of the module. Ports may connect to the ports on other on-chip modules, or may connect to the physical pins on the PLD.
Simultaneous Multi Master  The capability of having multiple masters communicating with distinct slaves concomitantly. This relies on the use of the Avalon bus module, which contains multiple point-to-point buses between masters and slaves (or their arbitrators), rather than a shared bus resource that could come under contention.

Slave Port  Any peripheral port that receives bus transactions initiated by a bus master. Slaves receive the control signals that masters transmit.

Slave Side Arbitration  This is the heart of Simultaneous Multi-Mastering. Whereas typical multi-master buses share the bus as the resource for contention, Simultaneous Multi-Mastering relies on point-to-point bus connections between masters and slave arbiters, where the slave devices are the resources under contention, that need arbitration.

SOPC  System on a Programmable Chip.

SOPC Builder  SOPC Builder is a system generation and integration tool developed by Altera. SOPC Builder generates the system module, which is the on-chip circuitry that comprises the Avalon Bus Module, a Nios processor, and peripherals. It has a graphical user interface for configuring the Nios processor and specifying the peripherals to connect to Nios via the Avalon bus module. SOPC Builder is responsible for creating and connecting HDL modules, which implement all or part of the user’s PLD design. In the most basic case, the SOPC Builder creates only an Avalon Bus Module with all data, address and control ports exposed. The user must connect these ports to peripherals components external to the system module. In most cases, the SOPC Builder is aware of the Nios processor and other peripherals, and automatically connects these components to the Avalon Bus as dictated by the user.

Status Signals  Signals that are generated by a slave port to signal their status to master ports. These signals may vary with the type of slave peripheral, but they typically include interrupt request (IRQ), and wait signals.

Streaming Peripherals  Peripherals with ports that support streaming reads and writes.

Streaming Read and Write  Streaming read and write allows multiple units of data to be transferred after a single address bus-cycle. See “Streaming Transfer”.
Streaming Transfer  A streaming transfer is defined as one or more data transactions, initiated by a bus master, which have a consistent width of transaction to an incremental region of address space. The increment step per transaction is determined by the width of transfer (byte, halfword, word). There is no minimum or maximum number of transfers per streaming transaction. The stream length is determined by the characteristics of the streaming bus master and streaming bus slave.

System Designer  Also known as Developer, used to standardize the term for the Nios user.

System Module  System Module, also referred to as “Nios System Module”, refers to the portion of an SOPC design that was automatically generated by SOPC Builder. At the minimum, the system module contains a Nios microprocessor and the entire Avalon bus module. The system module usually also contains several Avalon bus peripherals. The logic external to the system module may contain custom Avalon peripherals and other custom logic unrelated to the system module. The system module must be connected to the designer’s over-all PLD design. The ports on the system module will vary, depending on which peripherals are included in the system module and what settings were made in the SOPC Builder. These ports may include direct connections to the Avalon bus, and user-defined ports to peripherals inside the system module.

W

Wait-State Generation  Wait-state generation delays specific Avalon signals by one or more clock cycles, for the benefit of peripherals with special synchronization needs. Wait states can be generated to stall a master peripheral in cases when the target slave peripheral cannot respond in a single clock cycle. Wait states can also be generated in cases when read-enable and write-enable signals have setup or hold time requirements.