Glossary

A-B

**32-Bit Program** A program compiled to run in 32-bit mode. For example, programs compiled for the PA RISC 1.x processors.

**64-Bit Program** A program compiled to run in 64-bit mode. For example, programs compiled for the PA-RISC 2.0 processor in wide mode.

**100BT** 100BASE-T is the technical term for the Fast Ethernet or IEEE802.3u standard.

**ARP** Address Resolution Protocol.

**Attach Chain** A linked list of driver attach routines (<drv>_attach). As a hardware module is being configured, this list is walked to allow each driver in the system a chance to recognize and claim the hardware module.

**BAR** Base Address Register. One of six registers in a PCI device function’s configuration space. Each BAR may specify the base and size of a memory or I/O range for the device.

**Beta Semaphore** Mutually-exclusive, blocking semaphores. When a thread acquires a beta semaphore, it is released. The owning thread may subsequently block (i.e., sleep) and still keep ownership. Threads waiting to acquire an owned beta semaphore are blocked.

**BN-CDIO** Bus Nexus CDIO, software that manages platform-dependent bus connection hardware.

**Bus Mastering** Taking control of a bus and mastering DMA transactions.

**Bus Nexus** Connection between two buses.

**C**

**Cache Coherence** Consistency of data in host memory as viewed by processor caches and I/O devices.

**CDIO** Context-Dependent I/O module. A module in GIO framework which contains all bus specific and/or driver environment specific functionality.

**Class** A logical grouping of device or hardware modules by type. For instance the ‘class tape’ would include all tape devices regardless of bus interface.

**Coherent I/O** Accesses to data in host memory by I/O devices are consistent with accesses by processor caches. Hardware in the platform maintains the consistent view of data in host memory as DMA transactions flow through...
the hardware.

**D-E**

**Data Link Provider Interface** (DLPI)

**Direct Memory Access (DMA)**
Transactions for which the device interacts directly with memory without processor intervention.

**Driver**
Software module which controls a device, interface card or bus-nexus.

**Driver Environment**
A defined set of services and entry points which allow a driver to function.

**F-I**

**General I/O System (GIO)**

**Independent Hardware Vendor (IHV)**

**ILP32**
C language data model where int, long, and pointer data types are 32 bits in size.

**Init List**
A linked list of device driver init routines (<drv>_init) which is built as the drivers configure themselves and run as the I/O system configuration is completed to perform any device driver-specific initialization.

**Instance**
A number assigned to an I/O tree node. The number is unique within a driver class.

**I/O Bus**
Interconnect bus for I/O cards and devices. PCI is an example of an I/O bus.

**I/O Node**
An element of an I/O tree which includes all relevant information needed for configuring a single hardware module.

**I/O Adapter**
Hardware to provide IOVA translation between an I/O bus and the processor/memory interconnect. Devices on the I/O bus issue bus transactions to IOVA memory targets and the I/O adapter translates IOVA memory targets to physical addresses. The I/O adapter also participates in the coherency protocol of the processor caches for platforms that are coherent or semicoherent.

**I/O Page Directory (I/O PDIR)**
Address translation table associated with an I/O adapter. The I/O PDIR is analogous to the PDIR used by processors for virtual-to-physical address translations. It is a table maintained by the kernel to provide mappings between IOVAs and physical addresses.

**I/O Tree**
Data structure for recording the I/O subsystem configuration information.

**I/O Virtual Address (IOVA)**
Address used by I/O devices to access host memory. Platforms that are semicoherent or coherent, or where the processor/memory
interconnect is greater than 32 bits wide, generally implement IOVAs.

**IP** Internet Protocol.

**ISC** Interface Select Code. Usually used as a pointer to an element of a table of isc_table_type structures (one per interface card). Each ISC entry is used by WSIO to maintain interface device driver information.

**ISR** Interrupt Service Routine. A driver-specific routine which handles interrupts from the device.

**ISV** Independent Software Vendor

**J** Local Area Network.

**LP64** C language data model where the int data type is 32 bits wide, but long and pointer data types are 64 bits wide.

**MAC** Medium Access Control.

**Map PCI -> port handle**

Mapping a PCI I/O space address to a port handle is the act which allows a driver to access the I/O space using pci_read_port_uintNN_isc() and pci_write_port_uintNN_isc(), passing in the port handle as an argument. The mapping is done through a call to pci_get_port_handle_isc().

**Map PCI -> virtual address**

Mapping a PCI memory space address to a virtual address is the act that allows a driver to access PCI space using READ_REG_UINTNN_ISC() or WRITE_REG_UINTNN_ISC() with that virtual address. The mapping is done through a call to map_mem_to_host().

**MP** Multi-Processor

**MP Safe** Describes a module which is protected in an MP environment through the use of various spinlocks and semaphores. Note that MP-safeness does not imply any performance considerations due to the granularity of the semaphores (e.g., use of a single I/O Empire semaphore or separate semaphores for each instance all imply MP-safeness).

**MP Scalable** Describes an MP module which may add components without causing more drain on other MP modules. An MP-scalable driver will provide a separate spinlock for each instance of the driver. Non MP-scalable drivers may still be MP-safe but perhaps only provide a single semaphore and spinlock for all instances of the driver. Adding more instances of a non
MP-scalable driver will therefore cause additional taxing of those resources for each instance added to the system.

**NIC** Network Interface Card.

**Noncoherent I/O** Accesses to data in host memory by I/O devices are not made consistent with processor caches by hardware. Software must explicitly flush the processor caches prior to starting a DMA transaction by an I/O device; and, in the case of data read from an I/O device, purge the processor caches after the DMA transaction completes.

**PA** Precision Architecture. When referring to busses, these are busses which conform to the Precision I/O Architecture.

**PCI** Peripheral Component Interconnect. An industry standard bus used mainly by current generations of HP platforms as a means of providing expansion I/O.

**PCI Address** An address in the PCI memory or I/O space. This is the type of address found in a PCI memory or I/O base address register. It is NOT a virtual address or an I/O port handle, which a driver could use to access a card.

**PCI I/O Space** The 'space' that is addressed by an I/O cycle on the PCI bus. This is a less often used way to access card registers on cards who choose to respond to PCI I/O accesses. Most cards have registers that are in PCI memory space instead of I/O space (i.e., they respond to PCI memory cycles, not PCI I/O cycles). PCI memory space The space that is addressed by a memory cycle on the PCI bus. It is called memory space to indicate that it is memory-mapped input/output, as opposed to a special I/O style of input/output. Typical cards map their registers into PCI memory space, meaning they can only be accessed by PCI memory cycles.

**Physical Address** Real address by which host memory or an I/O device register is accessed.

**Port Handle** The kernel resource associated with a mapped range of PCI I/O space. This handle is used to access the I/O space addresses by calling 

```c
pci_read_port_uintNN_isc()
```

and

```c
pci_write_port_uintNN_isc()
```

**PPA** Physical Point of Attachment

**R-S** Service Attach Point

**SCSI** Small Computer System Interface. An industry standard
external I/O bus available on all HP9000 systems.

**Semia coherent I/O** Similar to coherent I/O. However, for the case of data read from an I/O device, software must synchronize the data that have been read into host memory after the DMA transaction completes.

**Series700** HP9000/7XX family of PA-RISC workstations.

**Series800** HP9000/8XX family of PA-RISC business servers.

**SIO** Server I/O; I/O environment for port-server drivers with origins in S/800 systems.

**SNAP** Sub-Network Access Point

**Spinlock** Basic locking primitive used by the kernel for short-term locks. When a thread acquires a spinlock, the thread’s current processor becomes the effective owner until the spinlock is released. Threads (processors) waiting to acquire an owned spinlock will spin while waiting; they do not block. For the duration that a processor owns a spinlock, external interrupts to the processor are disabled.

**T-Z**

**Virtual Address** Address used by processors, when executing in virtual mode, to access host memory. Address translation hardware converts a virtual address to a physical address before host memory is accessed. Virtual addresses may also be used to map and access I/O device registers.

**WSIO** Workstation and Server I/O; I/O environment for reentrant drivers with origins in S/700 systems and converged with S/800 systems.