

Taken from “PCI Express Technology Comprehensive Guide to Generations 1.x, 2.x, 3.0 MINDSHARE, INC.
Mike Jackson Ravi Budruk”

Single Root Enumeration Example

Now that we’ve discussed the basic elements involved in the enumeration process, let’s walk through an example of the process. Figure 3-13 on page 113 illustrates an example system after the buses and devices have been enumerated. The discussion that follows assumes that the configuration software uses either of the two configuration access mechanisms defined in this chapter to achieve this result. At startup time, the configuration software executing on the processor performs enumeration as described below.

1. Software updates the Host/PCI bridge Secondary Bus Number to zero and the Subordinate Bus Number to 255. Setting this to the max value means that it won’t have to be changed again until all the bus numbers downstream have been identified. For the moment, buses 0 through 255 are identified as being downstream.
2. Starting with Device 0 (bridge A), the enumeration software attempts to read the Vendor ID from Function 0 in each of the 32 possible devices on bus 0. If a valid Vendor ID is returned from Bus 0, Device 0, Function 0, the device exists and contains at least one Function. If not, go on to probe bus 0, device 1, Function 0.
3. The Header Type field in this example (Figure 3-12 on page 108) contains the value one (01h) indicating this is a PCI-to-PCI bridge. The Multifunction bit (bit 7) in the Header Type register is 0, indicating that Function 0 is the only Function in this bridge. *The spec doesn’t preclude implementing multiple Functions within this Device and each of these Functions, in turn, could represent other virtual PCI-to-PCI bridges or even non-bridge functions.*
4. Now that software has found a bridge, performs a series of configuration writes to set the bridge’s bus number registers as follows:
 - Primary Bus Number Register = 0
 - Secondary Bus Number Register = 1
 - Subordinate Bus Number Register = 255

The bridge is now aware that the number of the bus directly attached downstream is 1 (Secondary Bus Number = 1) and that the largest bus number downstream of it is 255 (Subordinate Bus Number = 255).

5. Enumeration software must perform a depth-first search. Before proceeding to discover additional Devices/Functions on bus 0, it must proceed to search bus 1.
6. Software reads the Vendor ID of Bus 1, Device 0, Function 0, which targets bridge C in our example. A valid Vendor ID is returned, indicating that Device 0, Function 0 exists on Bus 1.
7. The Header Type field in the Header register contains the value one (0000001b) indicating another PCI-to-PCI bridge. As before, bit 7 is a 0, indicating that bridge C is a single-function device.
8. Software now performs a series of configuration writes to set bridge C’s bus number registers as follows:

- Primary Bus Number Register = 1
 - Secondary Bus Number Register = 2
 - Subordinate Bus Number Register = 255
9. Continuing the depth-first search, a read is performed from bus 2, device 0, Function 0's Vendor ID register. The example assumes that bridge D is Device 0, Function 0 on Bus 2.
 10. A valid Vendor ID is returned, indicating bus 2, device 0, Function 0 exists.
 11. The Header Type field in the Header register contains the value one (0000001b) indicating that this is a PCI-to-PCI bridge, and bit 7 is a 0, indicating that bridge D is a single-function device.
 12. Software now performs a series of configuration writes to set bridge D's bus number registers as follows:
 - Primary Bus Number Register = 2
 - Secondary Bus Number Register = 3
 - Subordinate Bus Number Register = 255
 13. Continuing the depth-first search, a read is performed from bus 3, device 0, Function 0's Vendor ID register.
 14. A valid Vendor ID is returned, indicating bus 3, device 0, Function 0 exists.
 15. The Header Type field in the Header register contains the value zero (0000000b) indicating that this is an Endpoint function. Since this is an endpoint and not a bridge, it has a Type 0 header and there are no PCI-compatible buses beneath it. This time, bit 7 is a 1, indicating that this is a multifunction device.
 16. Enumeration software performs accesses to the Vendor ID of all 8 possible functions in bus 3, device 0 and determines that only Function 1 exists in addition to Function 0. Function 1 is also an Endpoint (Type 0 header), so there are no additional buses beneath this device.
 17. Enumeration software continues scanning across on bus 3 to look for valid functions on devices 1 - 31 but does not find any additional functions.
 18. Having found every function there was to find downstream of bridge D, enumeration software updates bridge D, with the real Subordinate Bus Number of 3. Then it backs up one level (to bus 2) and continues scanning across on that bus looking for valid functions. The example assumes that bridge E is device 1, Function 0 on bus 2.
 19. A valid Vendor ID is returned, indicating that this Function exists.
 20. The Header Type field in bridge E's Header register contains the value one (0000001b) indicating that this is a PCI-to-PCI bridge, and bit 7 is a 0, indicating a single-function device.
 21. Software now performs a series of configuration writes to set bridge E's bus number registers as follows:
 - Primary Bus Number Register = 2
 - Secondary Bus Number Register = 4
 - Subordinate Bus Number Register = 255
 22. Continuing the depth-first search, a read is performed from bus 4, device 0, Function 0's Vendor ID register.
 23. A valid Vendor ID is returned, indicating that this Function exists.
 24. The Header Type field in the Header register contains the value zero (0000000b) indicating that this is an Endpoint device, and bit 7 is a 0, indicating that this is a single-function device.
 25. Enumeration software scans bus 4 to look for valid functions on devices 1 - 31 but does not find any additional functions.

26. Having reached the bottom of this tree branch, enumeration software updates the bridge above that bus, E in this case, with the real Subordinate Bus Number of 4. It then backs up one level (to bus 2) and moves on to read the Vendor ID of the next device (device 2). The example assumes that devices 2 - 31 are not implemented on bus 2, so no additional devices are discovered on bus 2.
27. Enumeration software updates the bridge above bus 2, C in this case, with the real Subordinate Bus Number of 4 and backs up to the previous bus (bus 1) and attempts to read the Vendor ID of the next device (device 1). The example assumes that devices 1 - 31 are not implemented on bus 1, so no additional devices are discovered on bus 1.
28. Enumeration software updates the bridge above bus 1, A in this case, with the real subordinate Bus Number of 4. and backs up to the previous bus (bus 0) and moves on to read the Vendor ID of the next device (device 1). The example assumes that bridge B is device 1, function 0 on bus 0.
29. In the same manner as previously described, the enumeration software discovers bridge B and performs a series of configuration writes to set bridge B's bus number registers as follows:
 - Primary Bus Number Register = 0
 - Secondary Bus Number Register = 5
 - Subordinate Bus Number Register = 255
30. Bridge F is then discovered and a series of configuration writes are performed to set its bus number registers as follows:
 - Primary Bus Number Register = 5
 - Secondary Bus Number Register = 6
 - Subordinate Bus Number Register = 255
31. Bridge G is then discovered and a series of configuration writes are performed to set its bus number registers as follows:
 - Primary Bus Number Register = 6
 - Secondary Bus Number Register = 7
 - Subordinate Bus Number Register = 255
32. A single-function Endpoint device is discovered at bus 7, device 0, function 0, so the Subordinate Bus Number of Bridge G is updated to 7.
33. Bridge H is then discovered and a series of configuration writes are performed to set its bus number registers as follows:
 - Primary Bus Number Register = 6
 - Secondary Bus Number Register = 8
 - Subordinate Bus Number Register = 255
34. Bridge J is discovered and a series of configuration writes are performed to set bridge its bus number registers as follows:
 - Primary Bus Number Register = 8
 - Secondary Bus Number Register = 9
 - Subordinate Bus Number Register = 255
35. All devices and their respective Functions on bus 9 are discovered and none of them are bridges, so the Subordinate Bus Number of bridges H and J are updated to 9.
36. Bridge I is then discovered and a series of configuration writes are performed to set its bus number registers as follows:

- Primary Bus Number Register = 6
- Secondary Bus Number Register = 10
- Subordinate Bus Number Register = 255

37. A single-function Endpoint device is discovered at bus 10, device 0, function 0.

38. Since software has reached the bottom of this branch of the tree structure required for PCIe topologies, the Subordinate Bus Number registers for bridges B, F, and I are updated to 10, and so is the Host/PCI bridge's Subordinate Bus Number register.

The final values encoded into each bridge's Primary, Secondary and Subordinate Bus Number fields can be found in Fig.

