



PCI-SIG ENGINEERING CHANGE NOTIFICATION

TITLE:	Expanded Resizable BARs
DATE:	April 18, 2016
AFFECTED DOCUMENT:	PCI Express Base Specification Revision 3.1
COSPONSORS:	Wilson Snyder, Cavium, Inc. Joe Cowan, Hewlett Packard Enterprise

Part I

1. Summary of the Functional Changes

The Resizable BAR capability currently allows BARs of up to 512 GB (2^{39}), which allows address bits <38:0> to be passed into an Endpoint. This proposal extends resizable BARs to up to 2^{63} bits, which supports the entire address space.

2. Benefits as a Result of the Changes

An Endpoint may have a larger than 512 GB physical address space, either because it has a large amount of attached DRAM (e.g., one cosponsor has shipping Endpoints capable of 2 TB), or because the address represents a virtualized address space, e.g., one that will pass through a system memory management unit. Virtual addresses e.g., in existing architectures presently require 48 bits (existing x86_64), 49 bits (ARM), or 52 bits (PPC), and may grow, possibly to the full x86_64 and PCI Express architectural maximums of 64 bits. Therefore, supporting the entire possible BAR space of 63 bits seems advisable for maximum forward compatibility. (Some space needs to be reserved for Upstream host addresses, therefore a BAR cannot consume all 64 bits.)

Currently, limiting the resizable BARs to 512 GB means resources are either: a) simply not allocated and left out of the system, or b) forced to report a smaller aperture in order to be allocated, but that aperture size is not optimal in all uses of the product and may cause software to need to “bank” or otherwise move the aperture at runtime.

This change allows the system to be configured with optimal resource settings.

3. Assessment of the Impact

This capability is applicable to all components that have Base Address registers and derive value from presenting two or more options for the sizes of the resources inferred by the BARs. It is thought that Functions using this capability will generally be limited to very large resources, for example, add-in cards with extremely large local memories where the existing resizable BARs are insufficiently large.

4. Analysis of the Hardware Implications

This is an optional capability that is useful for Functions that have resources requiring more address space than currently supported. Components are not required to add this capability if they do not wish to or if it does not apply. The hardware implications for a component that does implement this capability are limited to extending configuration fields and BARs, in a manner straight-forward to those that have existing resizable BAR logic.

5. Analysis of the Software Implications

There is no impact to current software. The register values associated with this extension use reserved fields and encodings, and programming a legacy encoding will still result in proper operation.

Software that desires to support this extension can have straight-forward extensions applied to handle the additional address bits as described herein.

6. Analysis of the C&I Test Implications

There is no impact to current tests, other than perhaps relaxing register field tests which checked read-as-zero on the previously reserved and now allocated fields.

Part II

Detailed Description of the change

7.22. Resizable BAR Capability

The Resizable BAR Capability is an optional capability that allows hardware to communicate resource sizes, and system software, after determining the optimal size, to communicate this optimal size back to the hardware. Hardware communicates the resource sizes that are acceptable for operation via the Resizable BAR Capability [and Control](#) registers. [Hardware must support at least one size in the range from 1 MB to 512 GB.](#)



IMPLEMENTATION NOTE

[Resizable BAR Backward Compatibility With Software](#)

[The Resizable BAR capability initially supported 20 sizes, ranging from 1 MB to 512 GB, and was later expanded with 16 larger sizes. The hardware requirement to support at least one of the initial sizes ensures backward compatibility with software that comprehends only the initial sizes.](#)

Software determines, through a proprietary mechanism, what the optimal size is for the resource, and programs that size via the BAR Size field of the Resizable BAR Control register. Hardware immediately reflects the size inference in the read-only bits of the appropriate Base Address register. Hardware must Clear any bits that change from RW to read-only, so that subsequent reads return zero. Software must clear the Memory Space Enable bit in the Command register before writing the BAR Size field. After writing the BAR Size field, the contents of the corresponding BAR are undefined. To ensure that it contains a valid address after resizing the BAR, system software must reprogram the BAR, and Set the Memory Space Enable bit (unless the resource is not allocated).

The Resizable BAR Capability [and Control](#) registers ~~are~~ [is](#) permitted to indicate the ability to operate at 4 GB or greater only if the associated BAR is a 64-bit BAR.

This capability is applicable to Functions that have Base Address registers only. It is strongly recommended that a Function not advertise any supported BAR sizes ~~in its Resizable BAR Capability register~~ that are larger than the space it would effectively utilize if allocated.



IMPLEMENTATION NOTE

Using the Capability During Resource Allocation

System software that allocates resources can use this capability to resize the resources inferred by the Function's BAR's read-only bits. Previous versions of this software determined the resource size by writing FFFFh to the BAR, reading back the value, and determining the size by the number of bits that are Set. Following this, the base address is written to the BAR.

System software uses this capability in place of the above mentioned method of determining the resource size, and prior to assigning the base address to the BAR. Potential usable resource sizes are reported by the Function, ~~and read, from the~~ [via its](#) Resizable BAR Capability [and Control](#) registers. It is intended that the software allocate the largest of the reported sizes that it can, since allocating less address space than the largest reported size can result in lower performance. Software then writes the size to the Resizable BAR Command register for the appropriate BAR for the Function. Following this, the base address is written to the BAR.

For interoperability reasons, it is possible that hardware will set the default size of the BAR to a low size; that is, a size lower than the largest reported in the Resizable BAR Capability [and Control](#) registers. Software that does not use this capability to size resources will likely result in sub-optimal resource allocation, where the resources are smaller than desirable, or not allocatable because there is no room for them.

With the Resizable BAR capability, the amount of address space consumed by a device can change. In a resource constrained environment, the allocation of more address space to a device may result in allocation of less of the address space to other memory-mapped hardware, like system RAM. System software responsible for allocating resources in this kind of environment is recommended to distribute the limited address space appropriately.

The Resizable BAR Capability structure defines a PCI Express Extended Capability, which is located in PCI Express Extended Configuration Space, that is, above the first 256 bytes, and is shown below in Figure 7-108. This structure allows devices with this capability to be identified and controlled. A Capability and a Control register is implemented for each BAR that is resizable. Since a maximum of six BARs may be implemented by any Function, the Resizable BAR Capability structure can range from 12 bytes long (for a single BAR) to 52 bytes long (for all six BARs).

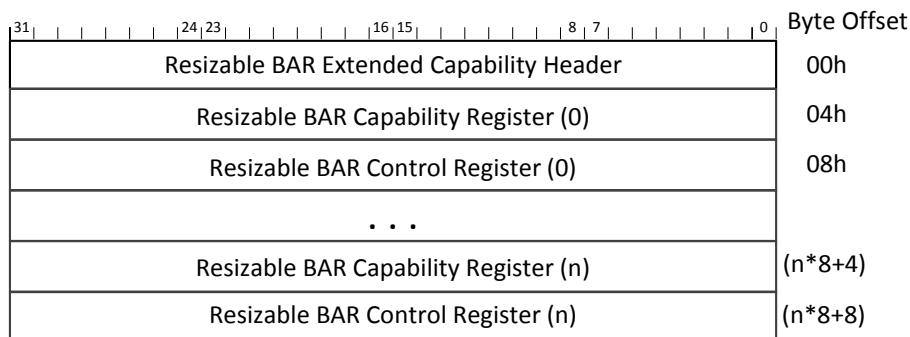


Figure 7-108: Resizable BAR Capability

Change Sections 7.22.2 & 7.22.3 as follows:

7.22.2. Resizable BAR Capability Register

For backward compatibility with software, hardware must Set at least one bit in the range from 4 to 23. See the associated Implementation Note in Section 7.22.



A-0745

Figure 7-110: Resizable BAR Capability Register

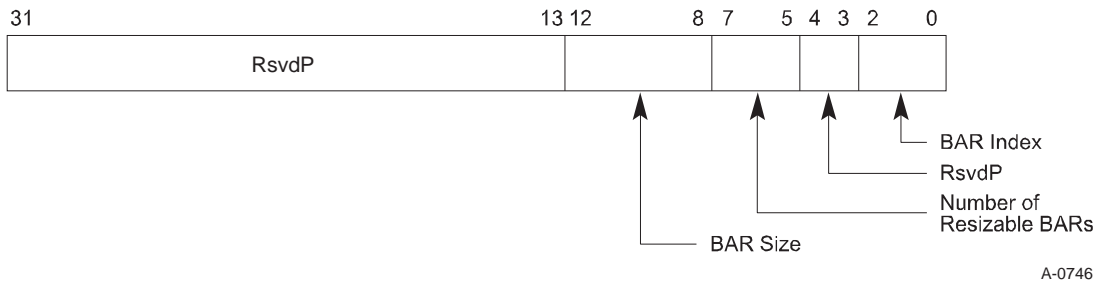
Note to editor: update the above figure to reflect the new bits added to the table below.

Table 7-96: Resizable BAR Capability Register

Bit Location	Register Description	Attributes
4	When Set, indicates that the Function supports operating with the BAR sized to 1 MB (2²⁰ bytes)	RO
5	When Set, indicates that the Function supports operating with the BAR sized to 2 MB (2²¹ bytes)	RO
6	When Set, indicates that the Function supports operating with the BAR sized to 4 MB (2²² bytes)	RO
7	When Set, indicates that the Function supports operating with the BAR sized to 8 MB (2²³ bytes)	RO
8	When Set, indicates that the Function supports operating with the BAR sized to 16 MB (2²⁴ bytes)	RO
9	When Set, indicates that the Function supports operating with the BAR sized to 32 MB (2²⁵ bytes)	RO
10	When Set, indicates that the Function supports operating with the BAR sized to 64 MB (2²⁶ bytes)	RO
11	When Set, indicates that the Function supports operating with the BAR sized to 128 MB (2²⁷ bytes)	RO
12	When Set, indicates that the Function supports operating with the BAR sized to 256 MB (2²⁸ bytes)	RO
13	When Set, indicates that the Function supports operating with the BAR sized to 512 MB (2²⁹ bytes)	RO
14	When Set, indicates that the Function supports operating with the BAR sized to 1 GB (2³⁰ bytes)	RO
15	When Set, indicates that the Function supports operating with the BAR sized to 2 GB (2³¹ bytes)	RO
16	When Set, indicates that the Function supports operating with the BAR sized to 4 GB (2³² bytes)	RO

Bit Location	Register Description	Attributes
17	When Set, indicates that the Function supports operating with the BAR sized to 8 GB (2³³ bytes)	RO
18	When Set, indicates that the Function supports operating with the BAR sized to 16 GB (2³⁴ bytes)	RO
19	When Set, indicates that the Function supports operating with the BAR sized to 32 GB (2³⁵ bytes)	RO
20	When Set, indicates that the Function supports operating with the BAR sized to 64 GB (2³⁶ bytes)	RO
21	When Set, indicates that the Function supports operating with the BAR sized to 128 GB (2³⁷ bytes)	RO
22	When Set, indicates that the Function supports operating with the BAR sized to 256 GB (2³⁸ bytes)	RO
23	When Set, indicates that the Function supports operating with the BAR sized to 512 GB (2³⁹ bytes)	RO
24	When Set, indicates that the Function supports operating with the BAR sized to 1 TB (2⁴⁰ bytes)	RO
25	When Set, indicates that the Function supports operating with the BAR sized to 2 TB (2⁴¹ bytes)	RO
26	When Set, indicates that the Function supports operating with the BAR sized to 4 TB (2⁴² bytes)	RO
27	When Set, indicates that the Function supports operating with the BAR sized to 8 TB (2⁴³ bytes)	RO
28	When Set, indicates that the Function supports operating with the BAR sized to 16 TB (2⁴⁴ bytes)	RO
29	When Set, indicates that the Function supports operating with the BAR sized to 32 TB (2⁴⁵ bytes)	RO
30	When Set, indicates that the Function supports operating with the BAR sized to 64 TB (2⁴⁶ bytes)	RO
31	When Set, indicates that the Function supports operating with the BAR sized to 128 TB (2⁴⁷ bytes)	RO

7.22.3. Resizable BAR Control Register



A-0746

Figure 7-111: Resizable BAR Control Register

Note to editor: update the above figure to reflect the changes in the table below. Note that the BAR Size field width has changed.

Table 7-97: Resizable BAR Control Register

Bit Location	Register Description	Attributes
...
<u>42</u> <u>13</u> :8	<p>BAR Size – This is an encoded value.</p> <p>0 = 1 MB (<u>2²⁰ bytes</u>)</p> <p>1 = 2 MB (<u>2²¹ bytes</u>)</p> <p>2 = 4 MB (<u>2²² bytes</u>)</p> <p>3 = 8 MB (<u>2²³ bytes</u>)</p> <p>...</p> <p><u>19</u> = 512 GB</p> <p><u>43</u> = 8 EB (<u>2⁶³ bytes</u>)</p> <p>The default value of this field is equal to the default size of the address space that the BAR resource is requesting via the BAR’s read-only bits. <u>For backward compatibility with software, the default value must be in the range from 0 to 19.</u></p> <p>When this register field is programmed, the value is immediately reflected in the size of the resource, as encoded in the number of read-only bits in the BAR.</p> <p>Software must only write <u>supported</u>-values that correspond to those <u>reported indicated as supported</u> in the Resizable BAR Capability <u>and Control</u> registers. Writing an unsupported value will produce undefined results.</p>	RW
<u>16</u>	<u>When Set, indicates that the Function supports operating with the BAR sized to 256 TB (2⁴⁸ bytes)</u>	<u>RO</u>
<u>17</u>	<u>When Set, indicates that the Function supports operating with the BAR sized to 512 TB (2⁴⁹ bytes)</u>	<u>RO</u>
<u>18</u>	<u>When Set, indicates that the Function supports operating with the BAR sized to 1 PB (2⁵⁰ bytes)</u>	<u>RO</u>
<u>19</u>	<u>When Set, indicates that the Function supports operating with the BAR sized to 2 PB (2⁵¹ bytes)</u>	<u>RO</u>

Bit Location	Register Description	Attributes
20	When Set, indicates that the Function supports operating with the BAR sized to 4 PB (2^{52} bytes)	RO
21	When Set, indicates that the Function supports operating with the BAR sized to 8 PB (2^{53} bytes)	RO
22	When Set, indicates that the Function supports operating with the BAR sized to 16 PB (2^{54} bytes)	RO
23	When Set, indicates that the Function supports operating with the BAR sized to 32 PB (2^{55} bytes)	RO
24	When Set, indicates that the Function supports operating with the BAR sized to 64 PB (2^{56} bytes)	RO
25	When Set, indicates that the Function supports operating with the BAR sized to 128 PB (2^{57} bytes)	RO
26	When Set, indicates that the Function supports operating with the BAR sized to 256 PB (2^{58} bytes)	RO
27	When Set, indicates that the Function supports operating with the BAR sized to 512 PB (2^{59} bytes)	RO
28	When Set, indicates that the Function supports operating with the BAR sized to 1 EB (2^{60} bytes)	RO
29	When Set, indicates that the Function supports operating with the BAR sized to 2 EB (2^{61} bytes)	RO
30	When Set, indicates that the Function supports operating with the BAR sized to 4 EB (2^{62} bytes)	RO
31	When Set, indicates that the Function supports operating with the BAR sized to 8 EB (2^{63} bytes)	RO