ECR #: 30
Title: Addressing beyond 4G
Release Date: May 12, 1997
Impact: Change
Spec Version: A.G.P. 1.0

Summary: This ECR describes how an A.G.P. compliant device should handle addresses greater than 32 bits. This includes both the use of **PIPE#** on the **AD** bus and SBA port. This ECR also contains some clarifications about the existing text when using the SBA port to enqueue A.G.P. requests.

Background: The A.G.P. interface specification alludes to support of addresses greater than 32 bits in the SBA port operation. However, this is limited to 36 bits of addressing. As systems grow and larger address spaces are supported, A.G.P. needs to allow addressing beyond the current definition for both the SBA port and **PIPE#** on the **AD** bus.

Change Current Specification as shown:

Modify Table 3-1 page 12 as follows:

CCCC	A.G.P. Operation
0000	Read
0001	Read (hi-priority)
0010	reserved
0011	reserved
0100	Write
0101	Write (hi-priority)
0110	reserved
0111	reserved
1000	Long Read
1001	Long Read (hi-priority)
1010	Flush
1011	reserved
1100	Fence
1101	Dual Address Cycle
1110	reserved
1111	reserved

Add the following description after the table after the Fence command and before the Reserved description.

> *Dual Address Cycle:* is used by the A.G.P. compliant master to transfer a 64 bit address to the A.G.P. compliant target when using the AD bus. When using the SBA Port to enqueue requests, the DAC command is not valid and must be implemented as a reserved command. The master is required to use two clock periods to transfer the entire address using **AD[31::00]** and **C/BE[3::0]#**. During the first clock the master provides the lower address bits (A31-A03) and the

length encoding on (A2-A0), just like a 32 bit request, but provides the DAC command (1101) encoding on **C/BE[3::0]#** instead of the actual command. The second clock of the request contains the upper address bits (A63-A32) on **AD[31::00]** and the actual command on **C/BE[3::0]#**.

A new section needs to be added after section 3.5.1.2 (Section 3.5.1.3. 64 bit addressing).

64-bit Addressing

The A.G.P. interface supports the accessing of data above the 4 gigabyte boundary by providing an addressing mechanism that provides more than 32 bits of addressing. The A.G.P. compliant master indicates that it desires to use addresses greater than 32 bits by setting but 5 in its A.G.P. status register (See section 6.1.8.) The master is only allowed to initiate a request that addresses a location above the 4 gigabyte boundary when bit 5 of its A.G.P. command register is set (See section 6.1.9.) The following two sections will explain how a master initiates a request that addresses a location above the 4 gigabyte boundary using when using the **AD** bus and when using the SBA port.

AD Bus

The A.G.P. compliant master can enqueue a request using the **AD** bus by asserting **PIPE#** when initiating the transaction. When the master initiates a 32-bit request, it enqueues a new request on each clock in which **PIPE#** is asserted. The last request is enqueued when **REQ#** is deasserted and **PIPE#** is asserted. When bit 5 of the master's A.G.P. command register is set, it is enabled to make 64-bit requests. When the A.G.P. master enqueues a request that uses a 64 bit address, it is required to use two clocks to transfer the request. During the first clock the master provide the lower address bits (A31-A2) and the length encoding (A2-A0) when the command is DAC (**C/BE[3::0]#** is 1101). The following clock the master provides the upper address bits (A63-A32) and the actual command on **C/BE[3::0]#**. **REQ#** is deasserted during the final clock of the last request to be enqueued. The arbiter is not required to decode the command of the request but can simply sample **PIPE#** and **REQ#** to determine when the enqueuing of requests is complete. Note: the A.G.P. compliant master is not allowed to uses DAC to enqueue a request when the upper address bits are all zero. In other words when the master must use a single address cycle (SAC) when accessing a location that resides within the lower 4 gigabytes of memory and is only allowed to use a DAC when accessing addresses above that range.

The A.G.P. compliant master is allowed to mix SAC and DAC within the same transaction on the interface. An A.G.P. compliant master is allowed to enqueue both requests that access data above and below the 4 gigabyte boundary in the same bus transaction.

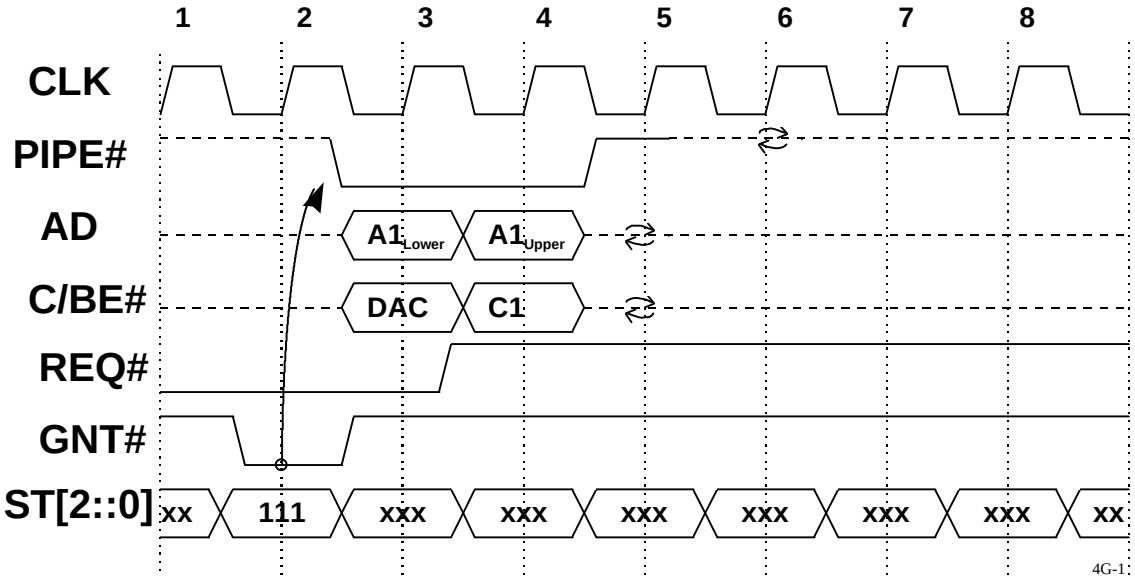

Figure 4G-1 illustrates a single request being enqueued that uses a 64 bit address or DAC command. The transaction is the same as 8-1 (ECR-19) except the DAC command is used during the first clock in which **PIPE#** is asserted. In this case, **PIPE#** is asserted for 2 clocks even though only a single address is enqueued. Notice that **REQ#** is deasserted during the second address phase of the transaction and not during the initial clock of the request. In this way the arbiter knows that it can assert a new **GNT#** on clock 6 and knows that no contention will occur on the shared signals.

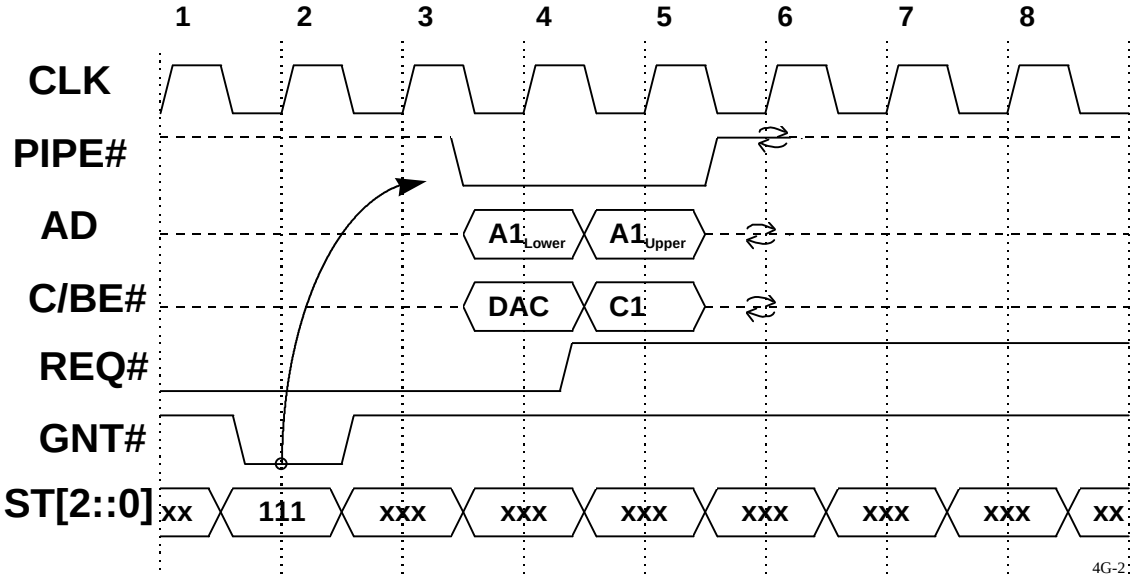

Figure 4G-2 is the same as 4G-1 except the master delays the assertion of **PIPE#** one clock.

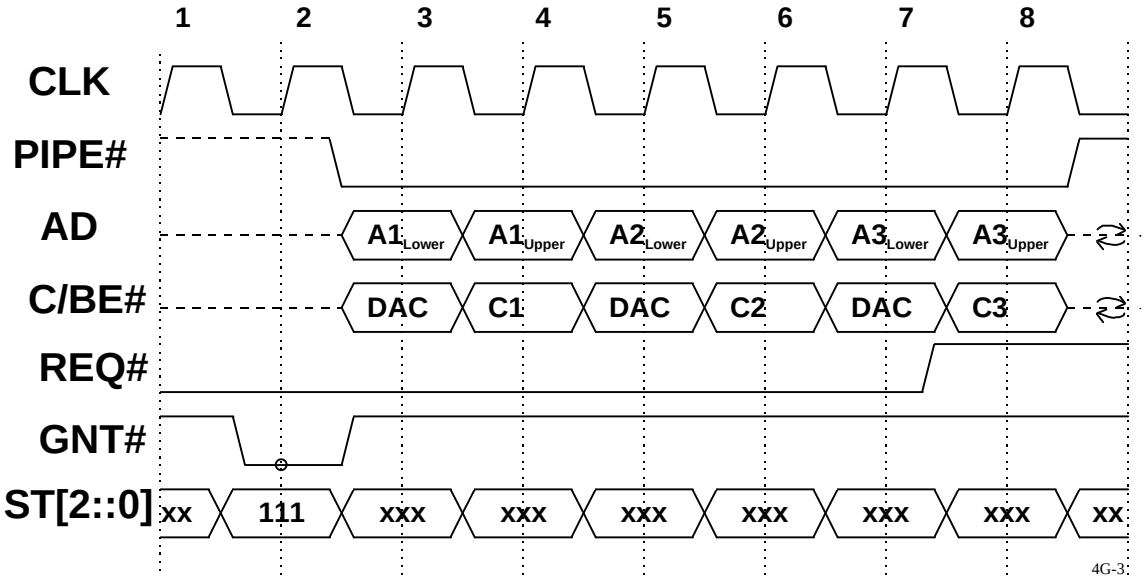

Figure 4G-3 illustrates multiple 64-bit requests being enqueued during a single request transaction. Notice that all these requests are using the DAC and require 2 clocks for each request to be transferred. The target (and arbiter) know that the last request is enqueued on clock 8 because **REQ#** is deasserted when **PIPE#** is asserted. The master always provides the lower part of the address and the length encoding on the first clock and the upper bits on the subsequent clock. The master is not allowed to insert or delay a request once **PIPE#** is asserted.

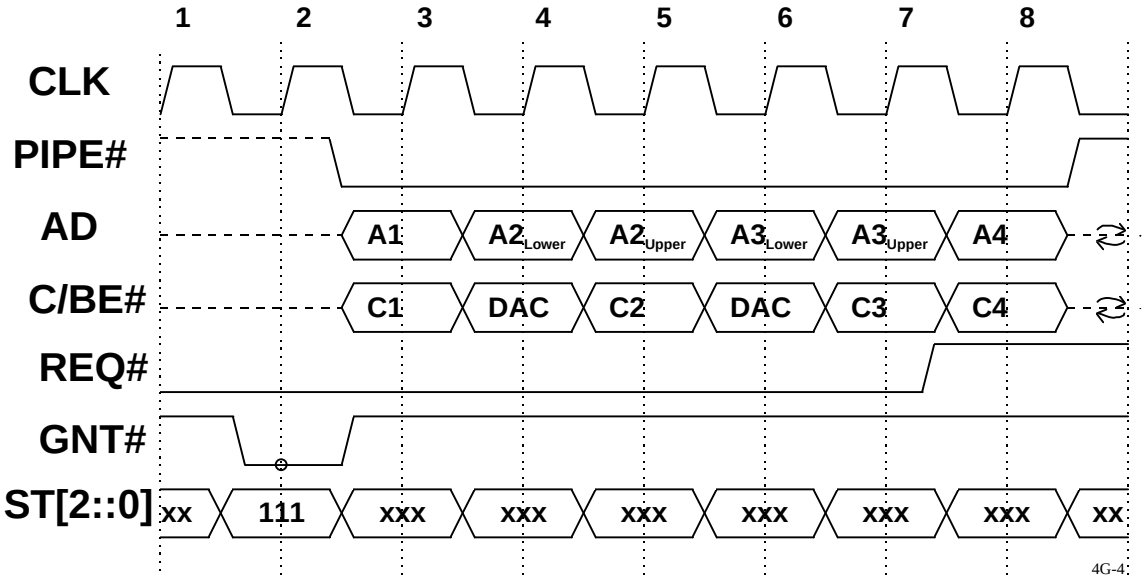

Figure 4G-4 illustrates SAC and DAC requests being enqueued during the same transaction. In this case the first and fourth request use the SAC while the second and third use the DAC command. For this transaction 2 transaction address data in the lower 4 gigabytes of the address space and the other 2 address data above the 4 gigabyte boundary. Like all requests that are enqueued using the **AD** bus, **REQ#** is deasserted during the final clock of the request transaction.

SBA Port

The A.G.P. compliant master can use the SBA port to enqueue requests instead of using the **AD** bus. In this way **AD** bus bandwidth is saved and allow requests to be enqueued while the **AD** bus is used for actual data movement. Table 3-2 defines the different messages that the master can use to transfer requests to the target. Note: that the SBA port only supports 48 bits of addressing while the AD bus supports all 64. When the SBA port is used, it is assumed by both the A.G.P. master and target that address bits 63 through 48 are zero.

Table 0-1 Sideband Address Port Encoding

Encoding	Description
S_7.. ...S_0	Shows alignment of messages on physical sideband wires.
1111 1111 **[1111 1111]**	***Bus Idle:*** used to indicate the bus is idle, also referred to as a NOP. When running at 1x transfer mode, this command is limited to a single clock tick of 8 bits (all ones) while 2x transfer mode requires the full 16 bits as shown here.
0AAA AAAA 14 08 AAAA ALLL 07 03	***Length & Lower Address Bits:*** the A.G.P. access length field (**LLL**), and lower 12 address bits (**A[14::03]**) are transferred across the sideband address port, and a memory access is initiated. The encoding is also referred to as a Type 1 sideband command. The remainder of the A.G.P. access request (**A[31::15]** and bus command) is defined by what was last transmitted using the other two sideband address port commands (Type 2 and Type 3. Note that **AD[2::0]** are assumed to be zero when using this encoding and these bits are not transferred.
10CC CCRA 15 AAAA AAAA 23 16	***Command & Mid Address Bits:*** the A.G.P. bus command (**CCCC**[1]) and mid-order 9 address bits (**A[23::15]**) are transferred across the sideband address port; no memory access is initiated. This encoding is also referred to as a Type 2 sideband command. This command, when followed by the previous command (Type 1) provides for memory access anywhere within a naturally aligned 16 MB 'page'. Note: the 'R' indicates a reserved bit that must be driven by the master as a "0" and is ignored by the target.
110R AAAA 35 32 AAAA AAAA 31 24	***Upper Address Bits:*** the upper 12 address bits (**A[35::24]**) are transferred across the sideband address port; no memory access is initiated. This encoding is also referred to as a Type 3 sideband command. This command, when followed by the two previous commands (Type 2 and Type 1) provides for memory access anywhere within a 32-bit physical address space. The master drives **A[35::32]** to zero when its bit 5 of the A.G.P. command register is cleared (0). When this bit is set (1), the master must actively drive these bits. The target ignores these bits when its corresponding bit 5 is cleared. Note: the 'R' indicates a reserved bit that must be driven as a "0" by the master and is ignored by the target.
1110 AAAA 39 36 AAAA AAAA 47 40	***Extended Address Bits:*** the extended 12 address bits (**A[47::36]**) are transferred across the sideband address port; no memory access is initiated. This encoding is also referred to as a Type 4 sideband command. This command, when followed by the three previous commands (Type 3, Type 2 and Type 1) provides for memory access anywhere within a 47-bit physical address space. The master must actively drive all bits when this command is used. Note: Bit 5 on the master's command register must be set to use this command.
1111 0*** **** ****	***reserved:*** must not be issued by an A.G.P. compliant master and maybe defined by Intel in the future.

[1] The A.G.P. master when using the SBA Port must treat the DAC command as a reserved command.

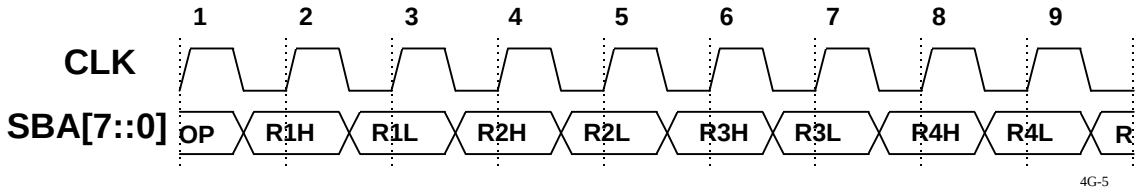

Figure 4G-5 illustrates the use of the SBA port to enqueue requests. R1, R2, R3 or R4 in the figure could use Type 1, 2 or 3 commands. A Type 4 command can only be used when enabled (see section 6.1.8 and 6.1.9 for details).

A.G.P. status register (offset CAP_PTR + 4)

Bits	Field	Description
31:24	**RQ**	The **RQ** field contains the maximum number of A.G.P. command requests this device can manage.
23:10	Reserved	Always returns 0 when read, write operations have no effect
9	**SBA**	If set, this device supports side band addressing.
8:6	Reserved	Always returns 0 when read, write operations have no effect
5	**4G**	If set, this device supports addresses greater than 4 gigabytes.
4:2	Reserved	Always returns 0 when read, write operations have no effect
1:0	**RATE**	The **RATE** field indicates the data transfer rates supported by this device. A.G.P. compliant devices must report all that apply. *<Bit 0: 1X, Bit 1: 2X >* *Note: The* **RATE** *field applies to* ***AD*** *and* ***SBA*** *buses.*

The A.G.P. status register is a *Read Only* register. Writes have no affect, and reserved or unimplemented fields return zero when read.

Change sections 6.1.8. AGP status register

A.G.P. command register - (offset CAP_PTR + 8)

Bits	Field	Description
31:24	**RQ_DEPTH**	Master: The **RQ_DEPTH** field must be programmed with the maximum number of pipelined operations the master is allowed to enqueue in the target. Value set in this field must be equal to or less than the value reported in the **RQ** field of target's status register. Target: The **RQ_DEPTH** field is reserved.
23:10	Reserved	Always returns 0 when read, write operations have no effect
9	**SBA_ENABLE**	When set, the side address mechanism is enabled in this device.
8	**AGP_ENABLE**	Master: Setting the **AGP_ENABLE** bit allows the master to initiate A.G.P. operations. When cleared, the master cannot initiate A.G.P. operations. Target: Setting the **AGP_ENABLE** bit allows the target to accept A.G.P. operations. When cleared, the target ignores incoming A.G.P. operations. *Notes: 1. The target must be enabled before the master .* *2. The* **AGP_ENABLE** *bit is cleared by* **AGP_RESET**.
7:6	Reserved	Always returns 0 when read, write operations have no effect
5	**4G**	Master: Setting the **4G** bit allows the master to initiate A.G.P. requests to addresses above the 4 gigabyte address boundary. When cleared, the master is only allowed to access addresses in the low 4 gigabytes of the address space. Target: Setting the **4G** bit enables the target to accept DAC commands when bit 9 is cleared. Setting the **4G** bit enables the target to accept a Type 4 command and to utilize **A[35::32]** for a Type 3 command when bit 9 is set.
4:2	Reserved	Always returns 0 when read, write operations have no effect
1:0	**DATA_RATE**	One *(and only one)* bit in the **DATA_RATE** field must be set to indicate the desired data transfer rate. *<Bit 0: 1X, Bit 1: 2X>*. The same bit must be set on both master and target. *Note: The* **DATA_RATE** *field applies to* **AD** *and* **SBA** *buses*