External Request Protocols

External requests can only be issued with the System interface in slave state. An external agent asserts **ExtRqst*** to arbitrate (see External Arbitration Protocol, below) for the System interface, then waits for the processor to release the System interface to slave state by asserting **Release*** before the external agent issues an external request. If the System interface is already in slave state—that is, the processor has previously performed an uncompelled change to slave state—the external agent can begin an external request immediately.

After issuing an external request, the external agent must return the System interface to master state. If the external agent does not have any additional external requests to perform, **ExtRqst*** must be deasserted two cycles after the cycle in which **Release*** was asserted. For a string of external requests, the **ExtRqst*** signal is asserted until the last request cycle, whereupon it is deasserted two cycles after the cycle in which **Release*** was asserted.

The processor continues to handle external requests as long as **ExtRqst*** is asserted; however, the processor cannot release the System interface to slave state for a subsequent external request until it has completed the current request. As long as **ExtRqst*** is asserted, the string of external requests is not interrupted by a processor request.

This section describes the following external request protocols:

- read
- null
- write
- invalidate and update
- intervention
- snoop
- read response

External Arbitration Protocol

System interface arbitration uses the signals **ExtRqst*** and **Release*** as described above. Figure 12-28 is a timing diagram of the arbitration protocol, in which slave and master states are shown.

The arbitration cycle consists of the following steps:

1. The external agent asserts **ExtRqst*** when it wishes to submit an external request.
2. The processor waits until it is ready to handle an external request, whereupon it asserts **Release*** for one cycle.
3. The processor sets the **SysAD** and **SysCmd** buses to tri-state.
4. The external agent must wait at least two cycles after the assertion of **Release*** before it drives the **SysAD** and **SysCmd** buses.
5. The external agent deasserts **ExtRqst*** two cycles after the assertion of **Release***, unless the external agent wishes to perform an additional external request.
6. The external agent sets the **SysAD** and the **SysCmd** buses to tri-state at the completion of an external request.

The processor can start issuing a processor request one cycle after the external agent sets the bus to tri-state.

NOTE: Timings for the **SysADC** and **SysCmdP** buses are the same as those of the **SysAD** and **SysCmd** buses, respectively.

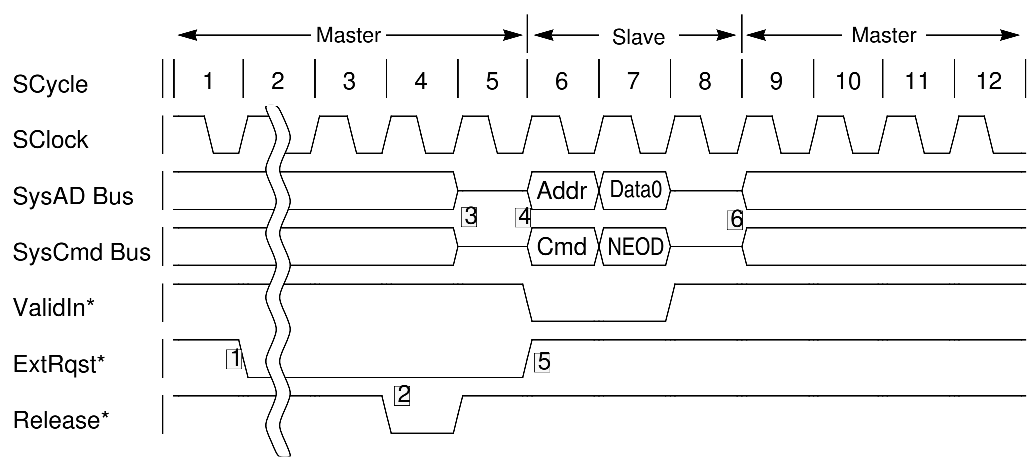

Figure 12-28 Arbitration Protocol for External Requests

External Read Request Protocol

External reads are requests for a word of data from a processor internal resource, such as a register. External read requests cannot be split; that is, no other request can occur between the external read request and its read response.

Figure 12-29 shows a timing diagram of an external read request, which consists of the following steps:

1. An external agent asserts **ExtRqst*** to arbitrate for the System interface.
2. The processor releases the System interface to slave state by asserting **Release*** for one cycle and then deasserting **Release***.
3. After **Release*** is deasserted, the **SysAD** and **SysCmd** buses are set to a tri-state for one cycle.
4. The external agent drives a read request command on the **SysCmd** bus and a read request address on the **SysAD** bus and asserts **ValidIn*** for one cycle.
5. After the address and command are sent, the external agent releases the **SysCmd** and **SysAD** buses by setting them to tri-state and allowing the processor to drive them. The processor, having accessed the data that is the target of the read, returns this data to the external agent. The processor accomplishes this by driving a data identifier on the **SysCmd** bus, the response data on the **SysAD** bus, and asserting **ValidOut*** for one cycle. The data identifier indicates that this is last-data-cycle response data.
6. The System interface is in master state. The processor continues driving the **SysCmd** and **SysAD** buses after the read response is returned.

   NOTE: Timings for the **SysADC** and **SysCmdP** buses are the same as those of the **SysAD** and **SysCmd** buses, respectively.

External read requests are only allowed to read a word of data from the processor. The processor response to external read requests for any data element other than a word is undefined.

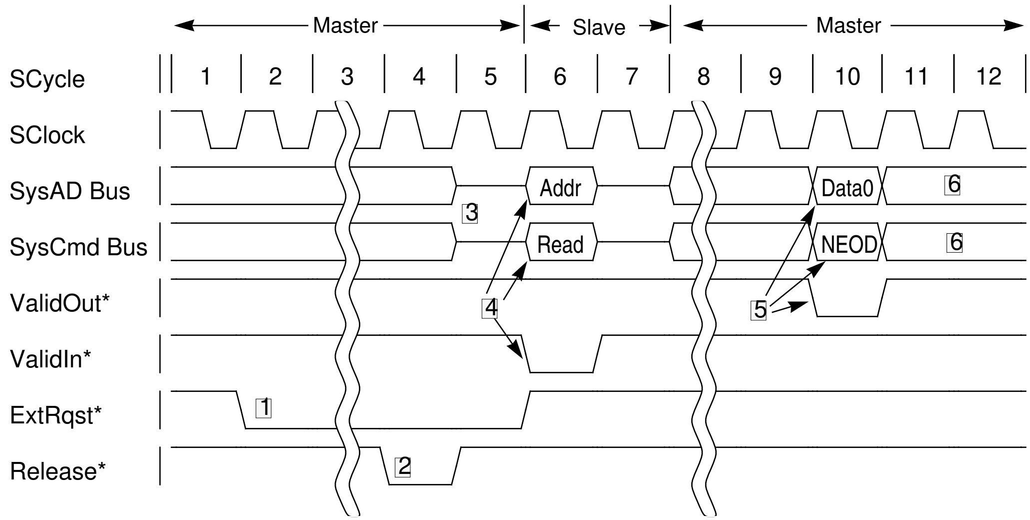

Figure 12-29 External Read Request, System Interface in Master State

NOTE: The processor does not contain any resources that are readable by an external read request; in response to an external read request the processor returns undefined data and a data identifier with its *Erroneous Data* bit, **SysCmd(5)**, set.

External Null Request Protocol

The processor supports two kinds of external null requests.

- A *secondary cache release external null request* returns ownership of the secondary cache to the processor while the System interface remains in slave state, until another external null request returns it to master state.
- A *System interface release external null request* returns the System interface to master state from slave state without otherwise affecting the processor.

Any time the processor releases the System interface to slave state to accept an external request, it also allows the external agent to use the secondary cache, in anticipation of a cache coherence request. When the external agent uses the **SysAD** bus for a transfer unrelated to the processor (for example, a DMA transfer), this ownership of the secondary cache prevents the processor from satisfying subsequent primary cache misses. To satisfy such a primary cache miss, the external agent issues a *secondary cache release external null request*, returning ownership of the secondary cache to the processor.

External null requests require no action from the processor other than to return the System interface to master state, or to regain ownership of the secondary cache.

Figures 12-30 and 12-31 show timing diagrams of the two external null request cycles, which consist of the following steps:

1. The external agent asserts **ExtRqst*** to arbitrate for the System interface.
2. The processor releases the System interface to slave state by asserting **Release***.
3. The external agent drives a secondary cache release external null request command on the **SysCmd** bus, and asserts **ValidIn*** for one cycle to return the secondary cache interface ownership to the processor.
4. The **SysAD** bus is unused (does not contain valid data) during the address cycle associated with an external null request.
5. After the address cycle is issued, the null request is complete.

For a *secondary cache release external null request*, the System interface remains in slave state.

For a *System interface release external null request*, the external agent releases the **SysCmd** and **SysAD** buses, and expects the System interface to return to master state.

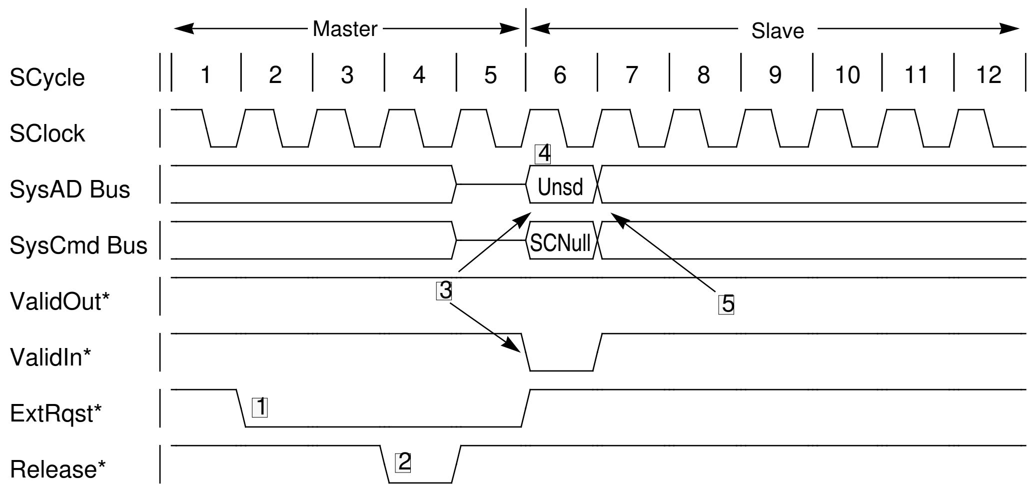

Figure 12-30 Secondary Cache Release External Null Request

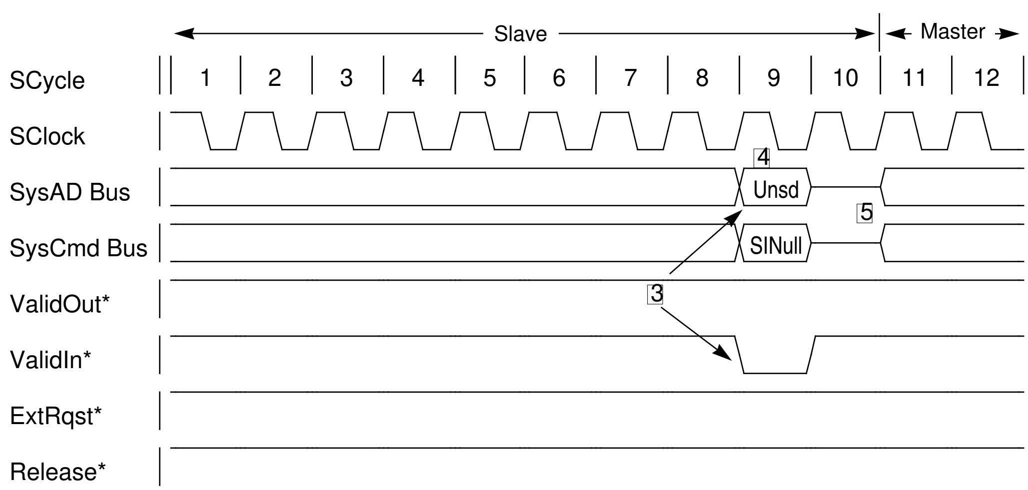

Figure 12-31 System Interface Release External Null Request

External Write Request Protocol

External write requests use a protocol identical to the processor single word write protocol except the **ValidIn*** signal is asserted instead of **ValidOut***. Figure 12-32 shows a timing diagram of an external write request, which consists of the following steps:

1. The external agent asserts **ExtRqst*** to arbitrate for the System interface.
2. The processor releases the System interface to slave state by asserting **Release***.
3. The external agent drives a write command on the **SysCmd** bus, a write address on the **SysAD** bus, and asserts **ValidIn***.
4. The external agent drives a data identifier on the **SysCmd** bus, data on the **SysAD** bus, and asserts **ValidIn***.
5. The data identifier associated with the data cycle must contain a coherent or noncoherent last data cycle indication.
6. After the data cycle is issued, the write request is complete and the external agent sets the **SysCmd** and **SysAD** buses to a tri-state, allowing the System interface to return to master state. Timings for the **SysADC** and **SysCmdP** buses are the same as those of the **SysAD** and **SysCmd** buses, respectively.

External write requests are only allowed to write a word of data to the processor. Processor behavior in response to an external write request for any data element other than a word is undefined.

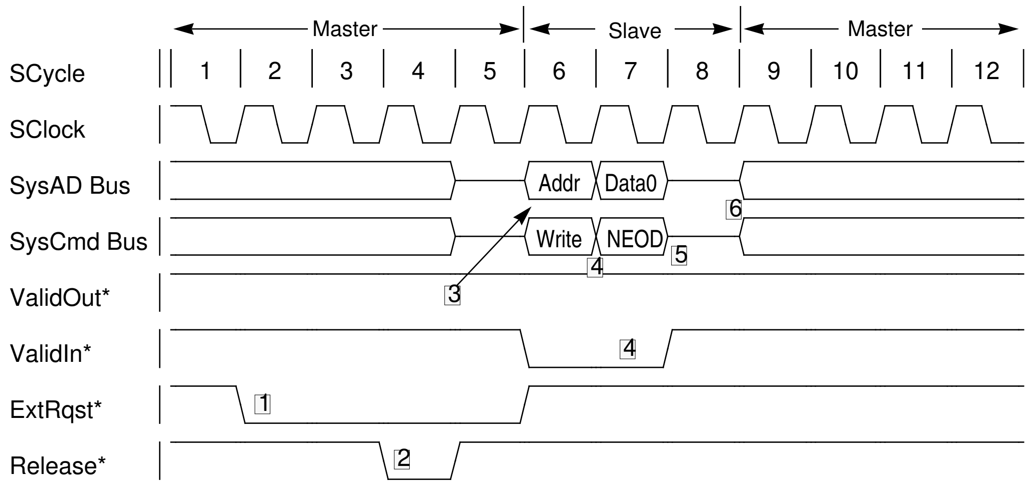

Figure 12-32 External Write Request, with System Interface initially a Bus Master

External Invalidate and Update Request Protocols

External invalidate and update request protocols are the same as the external write request protocol. The data element provided with an update or invalidate request can be a doubleword, partial doubleword, word, or partial word. The single data cycle transfer is not used (it does not contain valid data) for an invalidate request.

Figure 12-33 illustrates an external invalidate request following an uncompelled change to slave state.

> **NOTE:** Timings for the **SysADC** and **SysCmdP** buses are the same as those of the **SysAD** and **SysCmd** buses, respectively.

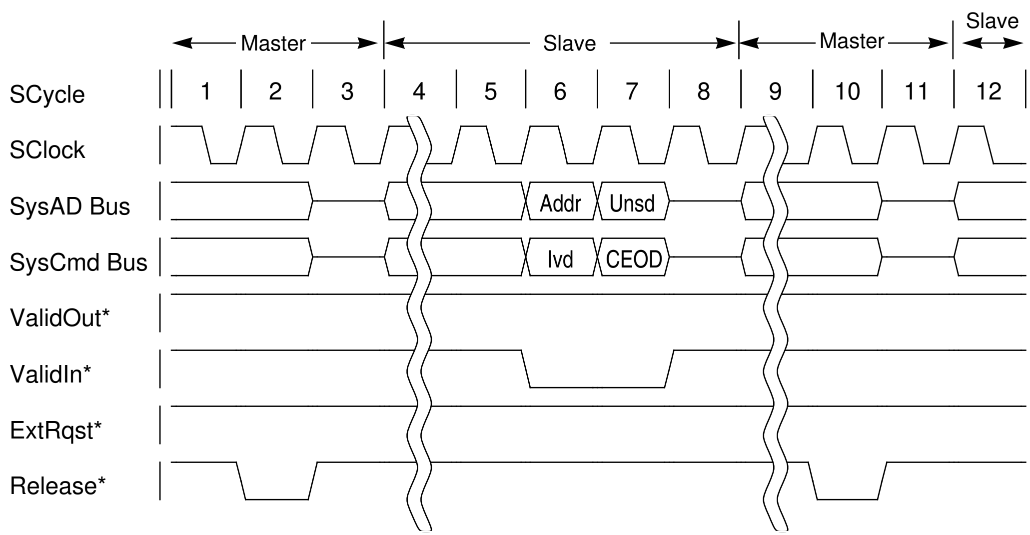

Figure 12-33 External Invalidate Request following an Uncompelled Change to Slave State

External Intervention Request Protocol

External intervention requests use a protocol similar to that of external read requests, except that a cache line size block of data can be returned along with an indication of the cache state for the cache line. The cache state indication depends upon the state of the cache line and the value of the *data return* bit in the intervention request command.†

The *data return* bit indicates either *return on dirty* or *return on exclusive*:

- If the *data return* bit indicates return on dirty, and the cache line that is target of the intervention request is in the dirty exclusive or dirty shared state, the contents of the cache line are returned in response to the intervention request.
- If the *data return* bit indicates return on exclusive, and the cache line that is the target of the intervention request is in the clean exclusive or dirty exclusive state, the contents of the cache line are returned in response to the intervention request.

If neither of the two cases above are true, the response to the intervention request does not include the *contents* of the cache line, but simply indicates the *state* of the cache line that is the target of the intervention request.

The case in which the processor returns a cache line state, but not cache line contents, is described in the following steps:

1. The external agent asserts **ExtRqst*** to arbitrate for the System interface.
2. The processor releases the System interface to slave state by asserting **Release***.
3. The external intervention request is driven onto the **SysCmd** bus and the address onto the **SysAD** bus. **ValidIn*** is asserted for one cycle.
4. The processor drives a coherent data identifier that indicates the state of the cache line on the **SysCmd** bus and asserts **ValidOut*** for one cycle.
5. The **SysAD** bus is not used during the data cycle.
6. The data identifier indicates a response data cycle that contains a last data cycle indication.

† If the cache line that is the target of the intervention request is not present in the cache—that is, the tag comparison for the cache line at the target cache address fails—the cache line that is the target of the intervention request is considered to be in the invalid state.

Figure 12-34 shows an external intervention request to a cache line found in the shared state, with the System interface initially in a master state. Figure 12-35 shows an external intervention request to a cache line found in the dirty exclusive state, with the System interface initially in a slave state.

NOTE: Timings for the **SysADC** and **SysCmdP** buses are the same as those of the **SysAD** and **SysCmd** buses, respectively.

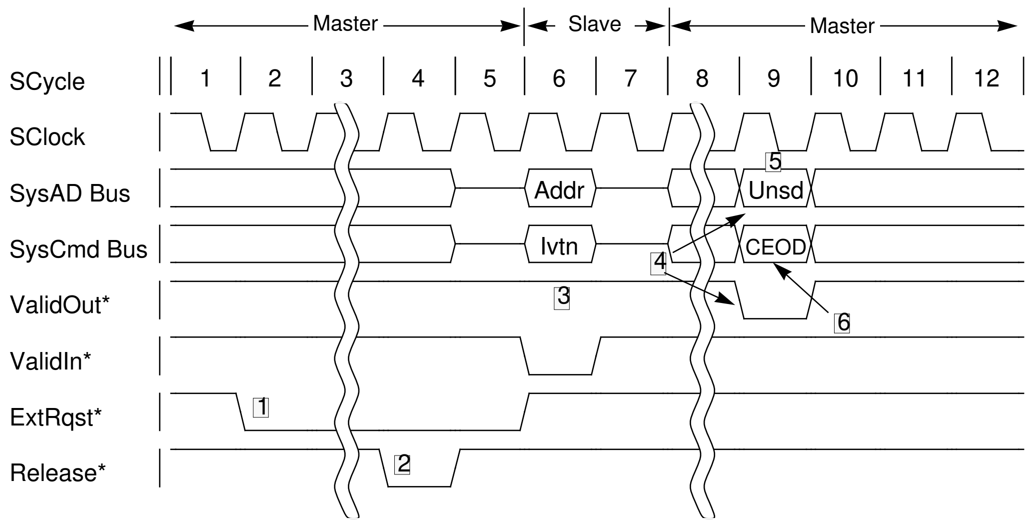

Figure 12-34 External Intervention Request, Shared Line, System Interface in Master State

The case in which the processor returns cache line contents is described in the steps below. In this example, the system is already in slave state.

1. The external intervention request is driven onto the **SysCmd** bus and the address onto the **SysAD** bus. **ValidIn*** is asserted for one cycle.
2. The processor drives data on the **SysAD** bus and a data identifier on the **SysCmd** bus. The processor asserts **ValidOut*** for each data cycle.
3. The data identifier associated with the last data cycle must contain a last data cycle indicator.

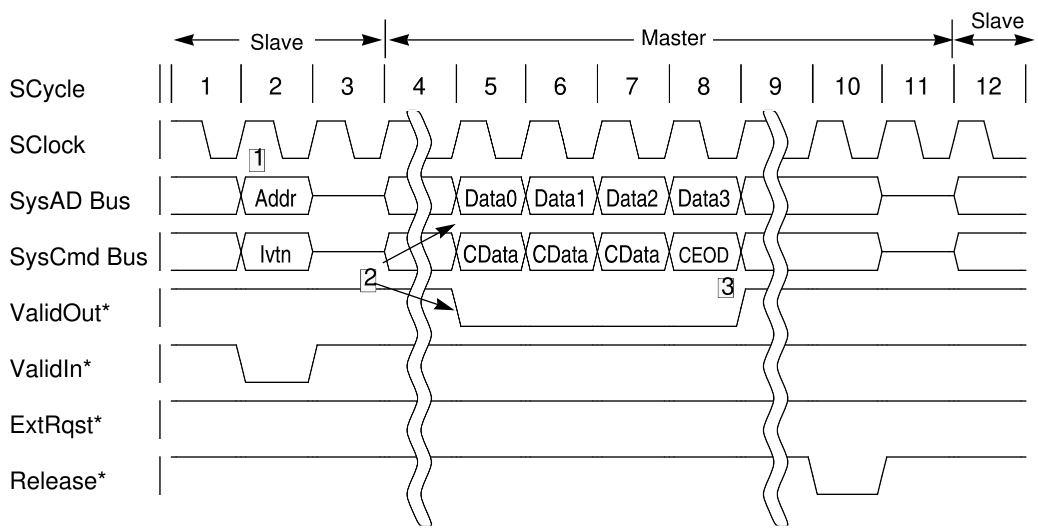

Figure 12-35 External Intervention Request, Dirty Exclusive Line, System Interface in Slave State

The processor returns the contents of a cache line, along with an indication of the cache state in which it was found, by issuing a sequence of data cycles sufficient to transmit the contents of the cache line, as shown in Figure 12-35. The data identifier transmitted with each data cycle indicates the cache state in which the cache line was found, together with an indication that this data is response data. The data identifier associated with the last data cycle contains a last data cycle indication.

If the contents of a cache line are returned in response to an intervention request, they are returned in subblock order starting with the doubleword at the address supplied with the intervention request. Note, however, that if the intervention address targets the doubleword at the beginning of the block, subblock ordering is equivalent to sequential ordering.

External Snoop Request Protocol

External snoop requests use a protocol identical to the external read request protocol, except that, instead of returning data, the processor responds with an indication of the current cache state for the targeted cache line. This protocol is described by the following steps:

1. The external agent asserts **ExtRqst*** to arbitrate for the System interface.
2. The processor releases the System interface to slave state by asserting **Release***.
3. The external snoop request is driven onto the **SysCmd** bus and the address onto the **SysAD** bus. **ValidIn*** is asserted for one cycle.
4. The processor drives a coherent data identifier on the **SysCmd** bus and asserts **ValidOut*** for one cycle.
5. The **SysAD** bus is unused during the snoop response.
6. The processor continues driving the **SysCmd** and **SysAD** buses after the snoop response is returned, to move the System interface back to master state.

Note that if the cache line that is the target of the snoop request is not present in the cache—that is, a tag comparison for the cache line at the target cache address fails—the cache line that is the target of the snoop request is considered to be in the invalid state.

Figure 12-36 shows an external snoop request submitted with the System interface in the master state. Figure 12-37 shows an external snoop request submitted with the System interface in slave state.

> **NOTE:** Timings for the **SysADC** and **SysCmdP** buses are the same as those of the **SysAD** and **SysCmd** buses, respectively.

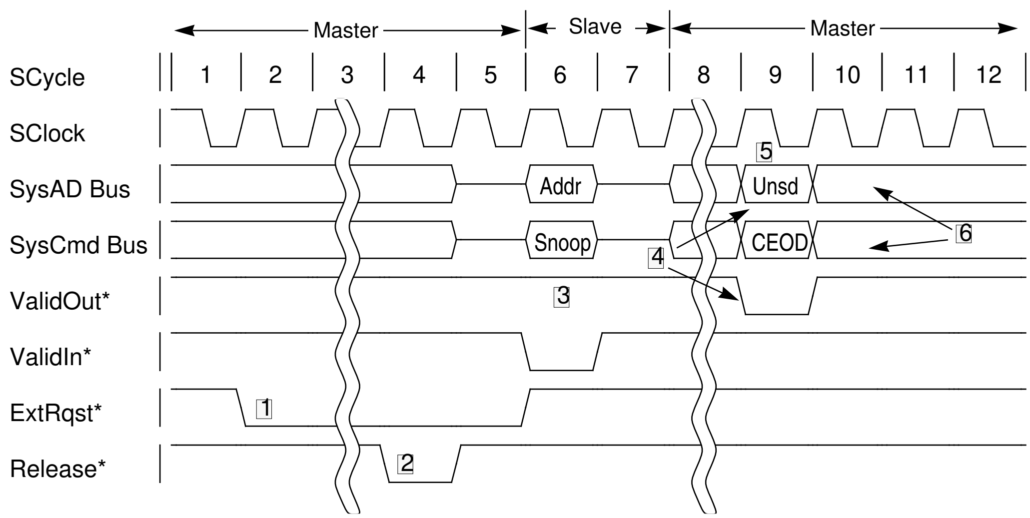

Figure 12-36 External Snoop Request, System Interface in Master State

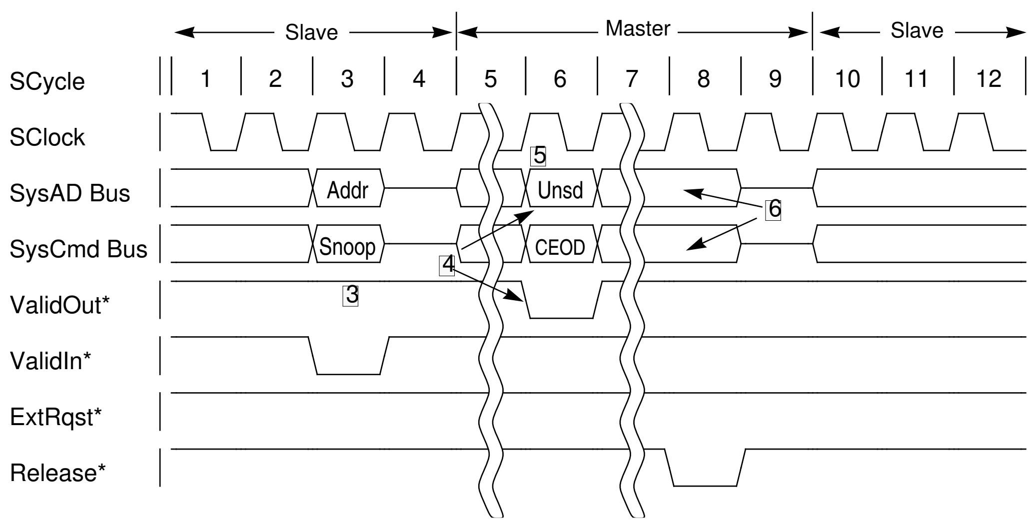

Figure 12-37 External Snoop Request, System Interface in Slave State

Read Response Protocol

An external agent must return data to the processor in response to a processor read request by using a read response protocol. A read response protocol consists of the following steps:

1. The external agent waits for the processor to perform an uncompelled change to slave state.
2. The processor returns the data through a single data cycle or a series of data cycles.
3. After the last data cycle is issued, the read response is complete and the external agent sets the **SysCmd** and **SysAD** buses to a tri-state.
4. The System interface returns to master state.

   NOTE: The processor always performs an uncompelled change to slave state after issuing a read request.
5. The data identifier for data cycles must indicate the fact that this data is *response data*.
6. The data identifier associated with the last data cycle must contain a *last data cycle* indication.

For read responses to coherent block read requests, each data identifier must include the cache state of the response data. The cache state provided with each data identifier must be the same and must be clean exclusive, dirty exclusive, shared, or dirty shared. The behavior of the processor is undefined if the cache state provided with the data identifiers changes during the transfer of the block of data, or if the cache state provided is invalid.

The data identifier associated with a data cycle can indicate that the data transmitted during that cycle is erroneous; however, an external agent must return a data block of the correct size regardless of the fact that the data may be in error. If a read response includes one or more erroneous data cycles, the processor then takes a bus error.

Read response data must only be delivered to the processor when a processor read request is pending. The behavior of the processor is undefined when a read response is presented to it and there is no processor read pending. Further, if the processor issues a read-with-write-forthcoming request, a processor write request or a processor null write request must be accepted before the read response can be returned. The behavior of the processor is undefined if the read response is returned before a processor write request is accepted.

Figure 12-38 illustrates a processor word read request followed by a word read response. Figure 12-39 illustrates a read response for a processor block read with the System interface already in slave state.

NOTE: Timings for the **SysADC** and **SysCmdP** buses are the same as those of the **SysAD** and **SysCmd** buses, respectively.

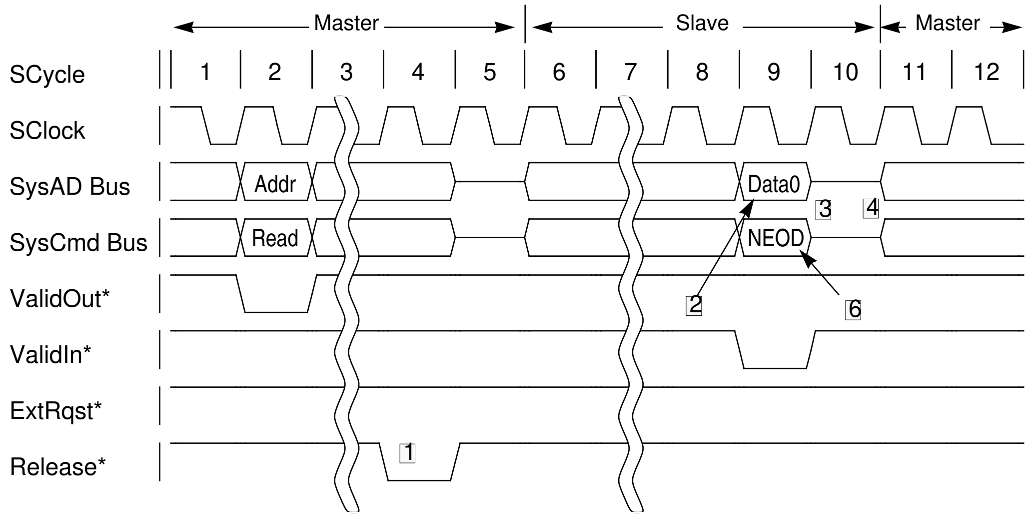

Figure 12-38 Processor Word Read Request, followed by a Word Read Response

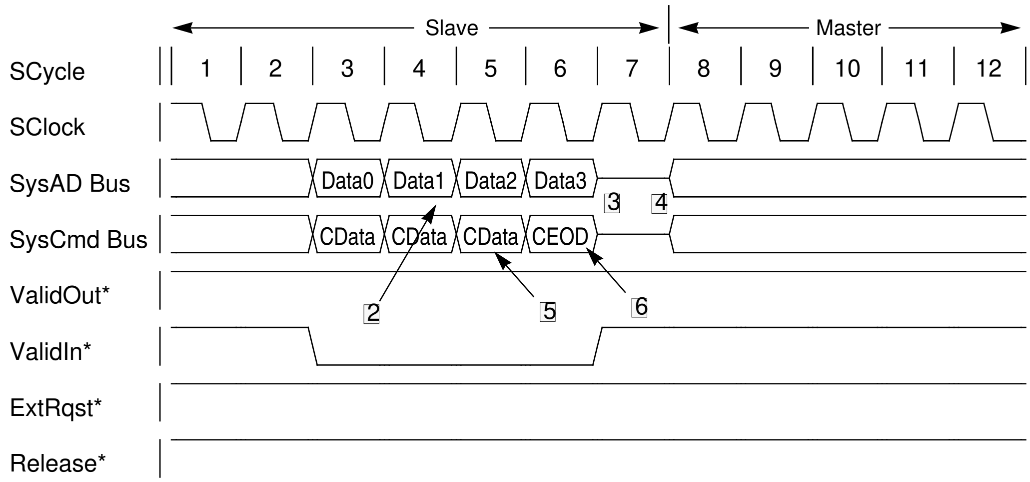

Figure 12-39 Block Read Response, System Interface already in Slave State