Bus Snooping and the IBM 486 DX2/DX4 Microprocessor

Application Note

Author: Scott Pheasant

Introduction

This paper explains the benefits of bus snooping and the differences among the IBM 486 DX2/DX4 revisions and the Intel® 486DX2. Please note that discussions about the Intel processors with write-through cache refer to the original 486DX2 and the Write-Back Enhanced 486DX2 which is operating in write-through mode. All write-back cache discussions reference the Write-Back Enhanced 486DX2 processor.

Ever since the first computer was designed, engineers have been searching for ways to increase system speed or performance. One of the largest restrictions on system speed was slow access to external memory. It simply took too long to perform writes and reads to and from memory that was limited by system bus speed and memory control logic.

The solution that is now a standard in microprocessors is the design of an "on-chip" memory that is much faster than external memory. This memory is known as cache memory, or simply cache. Cache size is much smaller than external memory, but allows for a quick storage space for most software applications' source code. With the invention of this quick memory, however, also came the problem of keeping external memory updated with changes that occur within the cache; this is termed "maintaining cache coherency." Write-through cache solves this by always writing the cache contents to external memory. A write-back cache system solves this by only writing those cache lines that have been updated ("dirty" cache lines) out to external memory.

A similar problem occurs when the processor is in a "held" state and other devices are updating external memory. In this case, cache will be inconsistent with external memory. The solution for this circumstance is called bus snooping. This method allows the external memory manager to invalidate cache addresses that are being updated in external memory. In general, a bus snoop cycle begins with an assertion of AHOLD, HOLD, or BOFF# by the system to force the processor to tri-state its address bus. The system then asserts EADS# to drive the modified address into the processor to allow for cache invalidations. Snooping does not enhance performance or reduce the final product's cost, it simply insures that data integrity is maintained within the processor so feasible application errors can be reduced. Depending on the type of cache and the processor you are using, snooping is implemented and initiated differently.

Snooping with Write-Through Cache - Intel & IBM

Both Intel's and IBM's 486 DX2/DX4 microprocessors have two methods for initiating snooping: a software method and a hardware method.[1]

For an Intel processor, the software method of performing a bus snoop is initiated by the issuing of a HLT (halt) instruction. When a software application issues a HLT instruction, the SL-Enhanced 486DX2 processor will automatically enter into a Power Down state. While the microprocessor is in this state, however, other devices which share the memory space with the processor may still be making changes to the external memory. To insure that the processor's cache is coherent with system memory, bus snooping (cache line invalidations) occurs. As previously explained, AHOLD, HOLD, or BOFF# is asserted followed by an EADS# to perform the snoop. In write-through mode, if the address that is driven with the EADS# into the processor is found within the processor's cache, it will be invalidated immediately.

The software method for snooping in an IBM processor is very similar to that of the Intel part. The biggest difference on the IBM processor, however, is the feature which allows a user the option of automatically suspending on a HLT instruction or not suspending on a HLT instruction. If bit 3 in the IBM's CCR2 register is set to a logic '1', then the CPU will automatically suspend whenever a HLT instruction is encountered. Otherwise, the IBM CPU operates as the Intel processor.

It is extremely important that the user understand which revision of the IBM microprocessor is being used for a particular application. Appendix B of the IBM Blue Lightning 486 DX2 Data Book explains the coding which is used for ordering specific processor revisions. If your order number does not have the code letter which identifies a revision, the part in question is of a revision which is previous to revision 4.2. IBM parts prior to revision 4.2 do not snoop whenever an automatic suspend on HLT is executed. Beginning with revision 4.2, all subsequent revisions will snoop when CCR2 register, bit 3, is set to a logic '1'.

The section entitled "Write Through Cache Coherency and the IBM Blue Lightning 486 DX2" covers solutions to possible data inconsistencies between the on-board cache and external memory.

As mentioned, there is also a hardware method for forcing bus snooping. If an Intel processor is in a stop grant state (the internal processor clock is electronically stopped after the assertion of the STPCLK# pin), then the processor will snoop on modified addresses. If the processor enters into system management mode, by the assertion of the SMI# pin, the processor is placed into a state which allows AHOLD, HOLD, BOFF#, and EADS# to continue functioning for bus snooping.

On an IBM processor, the SUSP# pin is used to place the processor in its suspended mode (if bit 7 of CCR2 register is set to a logic '1'). However, no revision of the IBM 486 DX2/DX4

---

[1] See items 2 and 5 in the References section at the end of this document for sources pertaining to snooping with write-through cache.

processor allows for snooping during this hardware-initiated suspend mode. System Management mode snooping for an IBM processor occurs just as with the Intel CPU.

Snooping and Write-Back Cache - Intel and IBM

The hardware and software methods of entering the different modes which trigger bus snooping are the same for a write-back cache as they are for the write-through cache. The actual implementation of the cache invalidations (bus snooping), however is a little different.

With Write-Back cache enabled, both the Intel 486DX2 microprocessor and the IBM microprocessor function very similarly.[2] When a snoop cycle occurs, AHOLD, HOLD, or BOFF# is asserted, followed by EADS#. If the address being snooped is found within the internal cache, that particular line is first checked to see if the data is dirty. If the data in the cache is dirty, the data is written to external memory and then marked invalid. If the cache line does not contain dirty data, that line is marked invalid.

Regardless of the cache mode which is being utilized, the IBM processor revision level is again extremely important. Revisions previous to rev 4.2 will not initiate snooping during suspend mode. This includes both the software (automatic suspend on HLT) and hardware (SUSP# pin) methods of implementing suspend mode. Beginning with rev 4.2 and above, whenever the automatic suspend on HLT bit (bit 3 of CCR2) is set, snooping will occur when the HLT instruction is executed. As with the other revisions, however, when using the SUSP# pin for suspend mode, snooping will not occur.

Snooping on both IBM and Intel CPUs always occurs during SMM mode.

Write-Through Cache Coherency and the IBM 486 DX2/DX4

When using an IBM processor, there are some instances when the processor will be placed into a "held" state to allow external devices direct control to the system memory, yet bus snooping will not occur. It is these cases in which cache coherency can become a real problem if system hardware or software is not designed properly.

In revisions 4.1 and below, no snoop will occur on any suspend cycles.[3] On revisions 4.2 and above, snooping will only occur when automatic suspend on HLT is initiated. If the IBM processor is set up for an automatic suspend on HLT (CCR2 register, bit 3, is set to logic '1') and a user is writing software code to be used on the processor, it is required that a WBINVD (Write-back Invalidate) instruction be issued before the implementation of the HLT instruction. The WBINVD instruction will flush all cache data before placing the processor into a suspended state. This will ensure that, when the processor is brought out of suspend mode, the cache will

---

[2] See items 2 and 5 in the References section at the end of this document for sources pertaining to snooping and write-back cache.

[3] See item 5 in the References section at the end of this document

have clean data because the CPU will be required to access external memory for the cache lines that were invalidated. If a system user is using pre-written software, it is recommended that the CCR2 register, bit 3, be set to a logic '0' so that an automatic suspend on a HLT instruction will never occur.

In hardware, the logic designer should force the processor's FLUSH# pin to be asserted first whenever a SUSP# pin assertion is requested. This will create the same results as the execution of a WBINVD instruction before a suspend on HLT instruction.

Write-Back Cache Coherency and the IBM 486 DX2/DX4

As with a write-through cache, bus snooping will not always occur when the processor is suspended, which could cause cache-memory inconsistencies.

The same methods described in the "Write-Through Cache Coherency and the IBM Blue Lightning 486 DX2/DX4" section of this paper are also true for this processor configuration.

References:

1. IBM 486 DX2 Addendum to the IBM Blue Lightning 486 DX2 Databook, August 11, 1995

2. Intel 486 Microprocessor Family Databook, 1994

3. Enhanced Am486™ Microprocessor Family Datasheet, May, 1995

4. IBM 486 DX4 Addendum to the IBM Blue Lightning 486 DX2 Databook, September 12, 1995

5. IBM Blue Lightning 486 DX2 Databook, 1994

6. IBM Blue Lightning Microprocessor Datasheet, February 7, 1994

7. Pentium™ Family User's Manual, 1994