AGING A STATION LIST WITH A TIME STAMP

Network bridges utilizing associative memories, such as the MU9C1480A LANCAM®, usually need to implement some form of aging of the station list where the LAN node addresses are tagged with some form of coded access history information, then periodically scanned, and the oldest entries purged from the list. Many bridge applications can efficiently implement a binary aging routine where each LAN address stored in the station list is either tagged as aged or not aged, and the aged locations are purged from the list at some selected interval. An alternate method of time stamping extends the same basic concept by providing several or many time intervals.

A 4-bit time stamp would provide 16 discrete intervals, or ticks, to implement this approach since we essentially need two sets of 8 ticks each. The purging cycle commences after the first 8 ticks have been counted, and thereafter at each tick when the time stamp is to be updated. A purge routine is also executed purging locations time stamped 8 ticks behind the current count value.

The local processor purge routine program can effectively keep track of the current time stamp value and the value of the oldest time stamp to be purged by maintaining a 16-bit counter for the time stamp intervals, and deriving the proper count value of the aged locations to be purged by merely reversing the MSB of the time stamp counter.

During the initial set of 8 ticks, counts 0 through 7, only the time stamp will be updated; no purges will occur. During counts the next set of 8 ticks, 8 through 15, the time stamp values 0 through 7 will be purged at each respective tick. When the 16-bit tick counter maintained by the local processor wraps around to a count of 0, the purging of locations stamped with binary 8 will be purged, and the cycle repeats indefinitely (see Figure 1 on page 3). The designer can determine the duration of each tick interval that will be optimal for the application.

STORING THE TIME STAMPS IN THE LANCAM A

The user can utilize the “RAM-only” Validity bit tag feature of the MU9C1480A to store each time stamp value in the LANCAM in separate scratchpad locations, thereby enabling a one-cycle or two-cycle 64-bit internal move of the time stamps rather than a multi-cycle 16-bit external write from the databus. During system initialization, 16 locations are written by issuing a Set Persistent Destination Memory at Address, Location Set RAM-Only instruction to access the designated locations, followed by the appropriate number of data write cycles containing the 4-bit time stamp codes in the RAM partition of the CAM word (the low-order 16-bit segment); the remainder of the segment is available for other purposes.

The user can utilize a workable feature of the MUSIC MU9C1480A LANCAM to accomplish the tagging of hit locations. When the device is set to 48 CAM bits, only the upper 48 bits of the Comparand register participate in the comparison, allowing the use of the lower 16 bits (segment 0) as a separate resource. Segment 0 of the Comparand does not take part in the compare operation; moreover, those lower 16 bits are not affected when the Comparand is loaded from the external 16 bit data bus when the segment counters are set to load segments 1 through 3. As a result, any data stored in Segment 0 of the Comparand will remain until specifically changed by the user, and can be used as a further storage resource to implement logical operations, such as appending Associated data to a data word after associative operations. This characteristic is designed to facilitate the use and manipulation of Associated data, one of the prime reasons to have a width of 64 bits in the first place.

During initialization, a MOV CR, aaaH (Move to Comparand from specified address) instruction is performed; aaaH is the address of the scratchpad location containing the value with the selected bit locations set to the binary code equating to the first time interval. The Comparand will now be loaded with a valid time stamp for the first interval stamp to be stored back into each location as it is accessed or hit. As the Comparand is loaded by the application with data words to be matched (the 48 bit node address to be searched), the lower 16 bits will be unchanged by virtue of the fact that the device is set to 48-bit (3-segment) loads by the state of the Segment Control register. This condition persists, so the proper time stamp remains in the lower 16 bits of the Comparand until changed by the application’s local processor to reflect the next time stamp value. After each interval time-out (tick), the value of the succeeding time stamp is loaded into the Comparand by accessing the MU9C1480A scratchpad location containing that next value

via a MOV CR aaah instruction, where Aaah is the address of the next desired time stamp.

TAGGING EACH LOCATION WITH THE TIME STAMP

After every successful compare operation, a MOV HM, CR instruction is performed (Move Comparand to the Highest Priority match). This instruction writes the time stamp to the associated data field of the stored node address with no effect on the CAM data field, since the Comparand and the data location contain the same value by virtue of the match. Only the Associated Data field will change to the new time stamp value.

PURGING THE LANCAM OF AGED ENTRIES

After the expiration of several time intervals as determined by the needs of the application (eight in this example), the oldest entries are purged via a purge routine. Instead of a mask of all bits but one being loaded into the Mask register, the 4 bits of the time stamp are left unmasked.

This operation is accomplished by loading the 4-bit binary value of the desired time stamp to be purged into the lower 16 bits (segment 0) of the Comparand register. A CMP V instruction is then performed, followed by a VBC ALM, E instruction (Set Validity Bit at All Matching Locations to Empty). This instruction will set to empty all locations containing the time stamp to be purged. At the expiration of the next interval, the above purge process is repeated, using the next oldest interval to be purged.

The value to be compared to locate the time stamps to be purged will change at every purge interval, until all intervals are purged, after which the cycle repeats itself by starting again at the first time stamp interval value. Additionally, since multiple time stamps are used, it is not necessary to reset the remaining unpurged locations to "not hit," because the remaining locations are already tagged or updated with a newer time stamp value by virtue of the stamp value being incremented and appended to the location during each address filtering match or learn cycle. After the initial 8 ticks have elapsed, it is no longer necessary for the system software to cause a separate MOV CR, aaaH to be executed at the completion of each tick to update the time stamp, as the purge routine should contain the MOV instruction as its last instruction. Since the purge routine will be executed at each tick after the initial 8 ticks, the time stamp in the Comparand will be updated every tick. If the purge routine is aborted due to a network interupt, the software will need to retry as soon as possible in order to maintain continuity.

TIME STAMP AND PURGE INTERVAL CYCLES

A further feature of this time stamp approach is its adaptability to be dynamically varied in real time as the local processor may determine necessary. An elastic tick interval can be programed to allow dynamic modification of the aging routine. For example, a good practice would be for the local processor to check the Full flag before a learn routine is performed and at the start of each purge routine, and keep track of how frequently Full flags occurred over a given (arbitrary) period of time. If the LANCAM array is often found to be full, the time interval can be shortened for more frequent purging by the local processor with no impact on the LANCAM control instruction sequence. With the implementation of more arithmetic routines by the local processor, the purge routine could be made to purge more than one value of time stamp per purge cycle. Conversely, if the array were not often full, the time cycle could be lengthened.

								p0	p1	p2	p3	p4	p5	p6	p7	p8	p9...
t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t0	t1...

T0 thru Tx = time stamp intervals
P0 thru Px = purge intervals

Figure 1: Time Stamp and Purge Interval Cycles

NOTES

MUSIC Semiconductors Agent or Distributor:

MUSIC SEMICONDUCTORS

http://www.music-ic.com
email: info@music-ic.com

USA Headquarters
MUSIC Semiconductors
254 B Mountain Avenue
Hackettstown, New Jersey 07840
USA
Tel: 908/979-1010
Fax: 908/979-1035
USA Only: 800/933-1550 Tech. Support
888/226-6874 Product Info.

Asian Headquarters
MUSIC Semiconductors
Special Export Processing Zone 1
Carmelray Industrial Park
Canlubang, Calamba, Laguna
Philippines
Tel: +63 49 549 1480
Fax: +63 49 549 1023
Sales Tel/Fax: +632 723 62 15

European Headquarters
MUSIC Semiconductors
Torenstraat 28
6471 JX Eygelshoven
Netherlands
Tel: +31 45 5462177
Fax: +31 45 5463663