Time Stamping Using The MUSIC LANCAMs®

INTRODUCTION

Time stamping using the MUSIC LANCAM provides an efficient method for aging and for performing preventive maintenance on a station list. When an entry is added to the station list, a value corresponding to the "current" time is written to the Associated Data field. If an address already exists in the station list, the time stamp is updated to the "current" time stamp each time it is matched. To facilitate preventive maintenance, a purge routine is run at predetermined intervals to search for entries marked with an "old" time stamp. The MUSIC LANCAM then removes all the old entries with a single instruction that changes their validity status to "Empty." While it is optimum to purge the station list before it is full, the Full flag can also be monitored to determine if the CAM is full and requires purging.

Sometimes it is desirable to keep entries in the station list permanently; i.e., they will not age out of the list. These permanent, or static, entries can be marked by setting a bit in the Associated Data field that prevents it from being set to "Empty" during the purge routine. This is accomplished by configuring a mask register to check for the static bit during a time stamp compare.

One form of time stamping uses a single bit for the time stamp. The time stamp for each entry is either "old" or "current." At every aging interval, the local processor removes "old" time stamps. Then it swaps the definition of "old" and "current" time stamp value, that is, the "current" time stamp becomes the "old" time stamp and the "old" time stamp becomes the "current" time stamp. As a result, entries left in the list start with "old" time stamps. Entries are updated with the new time stamp as they are accessed during the next time stamp interval.

A second type of time stamping uses multiple bits for the time stamp. A state machine or local processor generates a "current" time stamp code using a wrap-around counter. The state machine controls the aging by running a purge routine on the CAM at a predetermined interval. Each entry is stamped with the "current" time stamp when it is added or updated in the station list. Generally an "old" time stamp is purged just before a new time stamp is issued to the CAM. The time stamp that gets purged can become the "current" time stamp. Alternatively, a different value may be used for the "current" time stamp allowing multiple, discrete time divisions whose length can be easily controlled by the local processor.

The examples that follow use a 4-bit time stamp and a static bit. A 4-bit time stamp provides 16 discrete intervals. After every aging interval, the local processor increments the time stamp. Routines are written using ANSI C and memory mapped cycle types and lengths. For example, in a program statement, CWS is a Command Write cycle type with a Short cycle length. Binary time stamping requires that the global variables "old_ts" and "current_ts" are initially set to 0x4000 and 0x0000. Multiple bit time stamping requires that the global variable "current_ts" is initially set to 0x0000. The global variable "old_ts" may be offset from the "current_ts."

TIME STAMPING WITH A MUSIC 16-BIT I/O LANCAM

The following example is appropriate for a typical Ethernet Switch using the MU9C1480A or any other MUSIC "A" or "L" family 16-bit I/O LANCAM.

CAM CONFIGURATION

Speed and efficiency in purging, filtering, and learning routines in these applications can be maximized by taking advantage of the Background and Foreground Register set selection features of the LANCAM. The Background Register set is used for time stamp purging. The Foreground Register set is used for filtering and learning. Before proceeding, initialize the CAM by setting the Page Address registers. (Refer to the appropriate LANCAM Data Sheet.) See Figure 1 on page 2.

Configuring the Background Register Set

1. Select the Background Register set with the SBR instruction.

2. In the Control register, set the partitioning to 48 RAM, 16 CAM (CAM in Segment 0). This will allow comparisons on only Segment 0.

3. In Mask Register 1, set all the static or time stamp bits to 0 and all other bits to 1. This limits the comparison to the time stamp and static bits. Note, if using a binary time stamp with bit 14 for the time stamp and bit 15 for the static bit, the mask would be 0xFFFF FFFF FFFF 3FFF.

4. Set the Segment Control register to write to Segment 0 only.

5. Return the Persistent Destination to the Comparand.

Configuring the Foreground Register Set

1. Select the Foreground Register set with the SFR instruction.

2. In the Control register, set the partitioning for 48 CAM, 16 RAM and set Enhanced mode.

3. Configure the Segment Control register to write to Segments 0 to 3 and read from Segment 0 by sending the value 0x18C0 after the TCO_SC instruction. Note, the importance of this configuration is not readily seen by this application. However, a single Data Read cycle would provide Port ID or other Associated data information if it was required.

4. Mask Register 2 limits the update to the time stamp bits.

5. Set the Persistent Destination to the Comparand at the Persistent Source to the Highest-Priority match location.

```
void configuration ()                    // for 4-bit time stamp
{
        CWS     (TCO_DS);
        CWS     (0xFFFF);
        CWS     (SBR);                   // set background
        CWS     (TCO_CT);
        CWS     (0x8111);                // 48 RAM, 16 CAM, use MR1, Enhanced mode
        CWS     (TCO_SC);
        CWS     (0x18C0);                // write to Segments 00–11, read Segment 0
        CWS     (SPD_MR1);               // MASK REGISTER 1
        DWL     (0x07FF);                // adjust value for other than four time stamp bits
        DWL     (0xFFFF);
        DWL     (0xFFFF);
        DWL     (0xFFFF);
        CWS     (TCO_SC);
        CWS     (0x0000);                // write Segment 0 only
        CWS     (SPD_CR);
        CWS     (SFR);                   // set foreground
        CWS     (TCO_CT);
        CWS     (0x8041);                // 48 CAM / 16 RAM, Enhanced mode
        CWS     (TCO_SC);
        CWS     (0x18C0);                // write to Segments 00–11, read Segment 0
        CWS     (SPD_MR2);               // MASK REGISTER 2
        DWL     (0x8700);                // adjust value if not using four time stamp bits
        DWL     (0xFFFF);                // B14–11 time stamp, B7–0 Port ID
        DWL     (0xFFFF);
        DWL     (0xFFFF);
        CWS     (SPD_CR);                // set persistent destination to Comparand
        CWS     (SPS_HM);                // set persistent source to Highest-Priority match
}
```

Figure 1: Cam Configuration (16-bit I/O LANCAM)

ADDING TIME STAMPING TO THE LEARN ROUTINE

Typically, the learn routine looks up the source address in the station list. The decision to add a new entry or indicate that it already exists is made in the routine in which the time stamping occurs. See Figure 2.

1. The source address, time stamp, and Port ID are written to the Comparand. Note, the static bit is not set; i.e., it is 0.

2. The state machine reads the Match flag to check for match conditions.

3. One of the following occurs:

   - If the Match flag is equal to zero, there is a match and the time stamp and Port ID are updated.

   - If the address is not found, the Source Address is written to the CAM with the time stamp and Port ID.

PURGING ENTRIES

Purging entries is based on the "old" time stamp as long as the entry's static bit is not set. The CAM has to be repartitioned to allow comparison of just the Associated data field and configured to use a Mask register for compares. See Figure 3.

1. Use the SBR instruction to switch configurations. The Background Register set has the Destination Segment counter set to write to Segment 0 and the Control register set to 0x8111. Use the SBR instruction to switch to this configuration.

2. Load the time stamp to purge to Segment 0 of the Comparand.

3. All matching locations (i.e., those with the "old" time stamp) have their validity bits set to "Empty," which effectively clears out all the "old" addresses.

4. Use the SFR instruction to return the CAM/RAM partitioning, segment counters, and mask configuration to their original values.

```
void SA_Lookup (unsigned int PortID, unsigned int seg1, unsigned int seg2, unsigned int seg3)
{
        DWS         (current_ts | PortID);
        DWS         (seg1);
        DWS         (seg2);
        DWLEC       (seg3);                 // trigger compare
        if ( ( MF== 0 ) {                   // State machine reads Match flag for decision
                CWL (MOV_HM_CRMR2);         // update time stamp and Port ID
        }
        else    {                           // no match
                CWL (MOV_NF_CR_V);          // learn SA
        }
}
```

Figure 2: Learn Routine with Time Stamping (16-bit I/O LANCAM)

```
Void Purge ()

{
        CWS     (SBR);                      // select Background register set
        DWL     (old_ts);                   // load time stamp to purge, trigger compare
        CWL     (VBC_ALM_E);
        CWS     (SFR);                      // return to Foreground register set
        update_ts();
}
```

Figure 3: Purging Entries (16-bit I/O LANCAM)

5. A call to update_ts() sets the values for the "current_ts" (used in the SA_Lookup routine) and the next value to purge.

The "update_ts" routine is different for multiple-bit and single-bit time stamps. Figure 4 shows a 4-bit time stamp. The time stamp bits are B14–B11. To increment these bits, the 16-bit variable "current_ts" must be incremented by 0x0800 (or 0b0000 1000 0000 0000) each update interval. Also, to purge eight time stamps away, the "old_ts" variable would be initialized to 0x4000 and incremented the same as the "current_ts."

Figure 5 shows the update routine for a binary time stamp. It swaps the "old_ts" and the "current_ts" at each interval.

ADD PERMANENT ENTRIES

As discussed previously, some applications may require the addition of permanent entries to the station list. These entries will not be deleted during a purge routine based on a time stamp. One method of achieving this is shown in subroutine add_perm_addr in Figure 6.

1. The variables passed to this routine are PortID, seg1, seg2, and seg3. Seg1 to 3 contains only the address information.

2. The Port ID is bitwise "OR'd" with the static bit. This is written to the Segment 0 of the Comparand, followed by writes of seg1 through seg3 to Segment 1 through Segment 3.

```
Update _ts()
{
        current_ts = current_ts + 0x0800;          // increment 4-bit time stamp in bits 11–14
        if (current_ts=0x8000) {
            current_ts = 0; }
        old_ts = old_ts +  0x0800;                 // increment purge time stamp
        if (old_ts = 0x8000) {
           old_ts = 0; }
}
```

Figure 4: Update_ts Routine (4-bit Time Stamp)

```
Update_ts ()
{
        unsigned int temp;

        temp =old_ts;                              // swap old and current time stamps
        old_ts = current_ts;
        current_ts = temp;
}
```

Figure 5: Update_ts Routine (1-bit Time Stamp)

```
void add_perm_addr (unsigned int PortID, unsigned int seg1, unsigned int seg2, unsigned int seg 3)
{
        DWS    (0x8000 | PortID);                  // B15 is high for permanent entry
        DWS    (seg1);
        DWS    (seg2);
        DWS    (seg3);
        CWL    (MOV_NF_CR_V);                      // add permanent entry
}
```

Figure 6: Add Permanent Entries Routine (16-bit I/O LANCAM)

3. Use the Command Write instruction MOV_NF_CR_V to add the entry to the list. This moves the contents of the Comparand to the Next Free location and marks it valid.

DELETE PERMANENT ENTRIES

There must be a way to delete permanent entries if an application requires adding them to the station list. The address information is the only required data to delete an entry. See Figure 7.

1. Pass the data to the routine through the variables seg1, seg2, and seg3.
2. Execute a dummy write to Segment 0 (avoids changing the segment counters).
3. Seg1 through seg3 are written to Segments 1 through Segment 3 of the Comparand.
4. Use the Command Write instruction VBC_HM_E to set the validity bits of the Highest-Priority match location to "Empty."

TIME STAMPING WITH A MUSIC 32-BIT I/O LANCAM

The following example is appropriate for a typical Ethernet Switch using the MU9C1485A or any other MUSIC "A" or "L" family 32-bit I/O LANCAM.

CAM CONFIGURATION

Similar to the 16-bit example, speed and efficiency in these applications are facilitated by the Background and Foreground Register set selection features of the LANCAM. Before proceeding, initialize the CAM. (Refer to the appropriate LANCAM Data Sheet.) See Figure 8 on page 6.

Configuring the Background Register Set

1. Select the Background register set with the SBR instruction.
2. In the Control register, set the partitioning to 48 RAM, 16 CAM by writing the TCO_CT instruction, followed by 0x8111. This will allow comparisons only on the lower 16 bits of Segment 0 and through Mask Register 1. Bit 0 of the Control register selects Enhanced mode.
3. Set Mask Register 1 to allow the Compare cycle to look only at the time stamp and static bits. Set all other bits in the Mask register to 1s. Note, by selecting bit 14 and bit 15 for a 1-bit time stamp with a static bit, the mask would be 0xFFFFFFFF FFFF3FFF.
4. Set the Segment Control register to write to Segment 0 only.
5. Return the Persistent Destination to the Comparand register.

Configuring the Foreground Register Set

1. Select the Foreground register set with the SFR instruction.
2. In the Control register, set the partitioning for 48 CAM, 16 RAM and set the Enhanced mode.

```
void del_perm-addr (unsigned int seg1, unsigned int seg2, unsigned int seg 3)
{
        DWS     (0x0000);                       // dummy write
        DWS     (seg1);
        DWS     (seg2);
        DWLEC   (seg3);
        CWL     (VBC_HM_E);                     // clear permanent entry
}
```

Figure 7: Delete Permanent Entries (16-bit I/O LANCAM)

3. Configure the Segment Control register to write to Segment 0 to Segment 1 and read from Segment 0 by sending 0x0800 after the TCO_SC instruction. Note, the importance of this configuration is not readily seen by this application. However, a single Data Read cycle would provide Port ID or other Associated data information if it was required.

4. Mask Register 2 limits the update to the time stamp bit.

5. Set the Persistent Destination to the Comparand and the Persistent Source to the Highest-Priority match location.

The CAM partitioning is now ready to accommodate DA/SA data.

ADDING TIME STAMPING TO THE LEARN ROUTINE

Typically, the learn routine looks up the source address in the station list. It decides to add a new entry or update that the entry if it already exists. See Figure 9.

1. The variables passed into this routine are PortID, seg0, and seg1.

```
void configuration ()
{
        CWS     (TCOW_DS, 0xFFFF);
        CWS     (SBR);
        CWS     (TCOW_CT, 0x8111);      // 48 RAM / 16 CAM, use MR1, Enhanced mode
        CWS     (TCOW_SC, 0x0800);      // write to Segments 0–1, read Segment 0
        CWS     (SPD_MR1);              // set Mask Register 1
        DWL     (0xFFFF 07FF);          // B15 static bit, B14–11 time stamp
        DWL     (0xFFFF FFFF);          // adjust this for other than four time stamp bits
        CWS     (TCOW_SC, 0x0000);      // write Segment 0 only
        CWS     (SPD_CR);
        CWS     (SFR);
        CWS     (TCOW_CT, 0x8041);      // 48 CAM / 16 RAM, Enhanced mode
        CWS     (TCOW_SC, 0x0800);      // write to Segments 0–1, read Segment 0
        CWS     (SPD_MR2);              // set Mask Register 2 for updates
        DWL     (0xFFFF 8700);          // B14–11 time stamp, B7–0 Port ID
        DWL     (0xFFFF FFFF);          // adjust this for other than four time stamp bits
        CWS     (SPD_CR);
        CWS     (SPS_HM);
}
```

Figure 8: Cam Configuration (16-bit I/O LANCAM)

```
void SA_Lookup (unsigned int PortID, unsigned int seg0, unsigned long int seg1)
{
        unsigned   long int xseg0;
        xseg0=seg0;
        xseg0=xseg0<<16;
        DWS             (xseg0| current_ts | PortID);
        DWLEC           (seg1);
        if ( MF==0 ) {                          // State Machine reads the Match flag for decision
          CWL           (MOV_HM_CRMR2);         // update time stamp and Port ID
        }
        else  {                                 // no match
          CWL           (MOV_NF_CR_V);          // learn SA
        }
}
```

Figure 9: Learn Routine with Time Stamping (16-bit I/O LANCAM)

2. Seg0 contains only the lower 16 bits of address information. It is converted to 32-bit data format then shifted 16 bits to the left to accommodate placement in the first CAM segment (the upper 16 bits of Segment 0 are CAM, the lower 16 bits are RAM).

3. The modified Segment 0 value is bitwise "OR'd" with the current time stamp and Port ID. Note, the static bit is not set; i.e., it is 0. Both comprise the Associated Data field. This 32-bit value is written to the Comparand Segment 0. Then, value seg1 is written to Segment 1.

4. One of the following occurs:

   - If the Match flag is equal to zero, there is a match and the time stamp and Port ID are updated.
   - If the address is not found, the Source Address is written to the CAM with the time stamp and Port ID.

PURGING ENTRIES

Purging entries in a 32-bit I/O LANCAM can be implemented the same way as in a 16-bit I/O LANCAM. Since the Background register configuration only looks at the lower 16 bits of Segment 0 through a mask, no reformatting or appending of the "old_ts" is necessary. The "update_ts()" routine is identical to the one used for 16-bit LANCAMs.

ADD PERMANENT ENTRIES

As discussed previously, some applications may require the addition of permanent entries to the station list. These entries will not be deleted during a purge routine based on a time stamp. One method of achieving this is shown in subroutine add_perm_addr. This routine is very similar to the learning routine. This routine does not compare the new entry to the existing list and the static bit is set. However, in the learning routine, the source address is compared to the existing list and the static bit is not set; i.e., it is 0. See Figure 10.

```
void add_perm_addr (unsigned int PortID, unsigned int seg0, unsigned long int seg1)
{
        unsigned  long  int apseg0;
        apseg0=seg0;
        DWS     (apseg0<<16 );                  // B15 is high for permanent entry
        DWS     (seg1);
        CWL     (MOV_NF_CR_V);                  // add permanent entry
}
```

Figure 10: Add Permanent Entries (32-bit I/O LANCAM)

```
void del_perm_addr (unsigned int seg0, unsigned long int seg1)
{
        unsigned  long  int dpseg0;
        dpseg0=seg0;
        DWS     (dpseg0<<16 );
        DWLEC   (seg1);
        CWL     (VBC_HM_E);                     // clear permanent entry
}
```

Figure 11: Delete Permanent Entries (32-bit I/O LANCAM)

DELETE PERMANENT ENTRIES

To delete a permanent entry, only the address information is required for a Compare cycle. See Figure 11.

1. Pass the data to the routine through seg0 and seg1. Once again, seg0 is a 16-bit value that is converted to a 32-bit value then shifted left 16 bits.

2. Write this to Segment 0 of the Comparand, followed by writing seg1 to Segment 1 of the Comparand.

3. Execute the Command Write instruction VBC_HM_E to set the validity bits of the Highest-Priority match location to "Empty."

SUMMARY

Whether one is using 16-bit I/O MUSIC LANCAMs or 32-bit I/O MUSIC LANCAMs, time stamping data is an efficient method of maintaining an access history for lists or tables of data. Through a series of relatively simple hardware-implemented software routines, data can be added, deleted, updated, or purged from the list based on the permanence or time stamp information.

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