

# QUICC™ Ethernet Address Filtering with DSP56002

By Michael Leung

February 26, 1994

## Introduction

The MC68EN360 *Quad Integrated Communications Controller* (QUICC™) is the first Integrated Processor with built-in Ethernet Local Area Network (LAN) capability and multi-protocol Serial Communications Controllers [1]. In fact, one can build a 2-port Ethernet to Ethernet bridge, or an Ethernet to Wide Area Network (WAN) Bridge/Router using a single device. Larger systems can utilize additional QUICC's in slave mode (as peripherals), or attach a MC68EC040 processor to the QUICC to obtain 5X CPU processing power increase for maximum protocol handling and packet throughput.

In an Ethernet bridge or router implementation, there is a requirement to *ignore* (ie. not forward) frames that are not destined for nodes attached to network segments connected to outgoing ports. This is done to avoid flooding the whole network with local traffic in order to conserve usable network bandwidth. This document discusses the filtering of Ethernet frames based on their destination addresses and an implementation using the Motorola 24-bit DSP56002 Digital Signal Processor.

## Address Filtering Techniques

### (1) CPU Initiated Address Filter

The simplest method for address filtering requires that *all frames* from each Ethernet port be received and transferred to memory. It is up to the CPU to read the destination address from each packet header and look up a list of node addresses to determine whether the frame needs to be forwarded and through which port. Any frame that does not need to be forwarded is discarded. The lookup mechanism can be performed by a variety of search algorithms or can be hardware assisted using a Content Addressable Memory (CAM) such as the MU9C1480 LANCAM from MUSIC Semiconductors [2]. The software based lookup is low cost but its slower speed can severely limit the packet throughput of the bridge/router. The hardware CAM approach is much faster but still requires the CPU to pass each destination address to the CAM for lookup. In either case, memory bus bandwidth is wasted since all unwanted frames must still be completely transferred to memory.

## (2) Ethernet Controller Interfaced Address Filter

A more bus-bandwidth efficient technique would be to *terminate reception* of a frame as soon as it is determined that it does not need to be forwarded. A significant advantage is that the termination can be made transparent to the CPU such that no interrupts are generated at all for discarded frames. This method is supported by the QUICC's ability to reject a frame during reception if the Receive Reject pin (\RRJCT) is asserted by the external address filtering hardware. The destination address of each received frame is captured from the QUICC either serially or in parallel. The serial form is obtained by tapping the Receive Data (RXD) pin fed by the MC68160 10Base-T/AUI Enhanced Ethernet Serial Transceiver (EEST) while the parallel form is obtained by capturing the frame header while it is being written to memory by the QUICC's Ethernet Serial DMA controller. The parallel interface option additionally allows a Tag byte to be appended to each accepted frame for routing information. Special control signals are provided to signify the timing of the destination address information.

## (3) On-Chip Address Filter

The most bus-bandwidth efficient technique is to build into the LAN controller address filtering hardware such as a custom content addressable memory. This way, only frames that need to be forwarded are transferred to memory and no memory bandwidth is wasted on discarded frames. However, such approach is only practical for small CAM size due to die size constraints. Therefore on-chip CAM's are typically used for end-node or small bridge applications. An example is the MC68840 Integrated FDDI System Controller which contains both a 32-entry on-chip CAM for end-node applications and an external CAM interface for bridge/router applications.

In the following, we will examine external address filter implementations for the QUICC using a combination of the first two techniques.

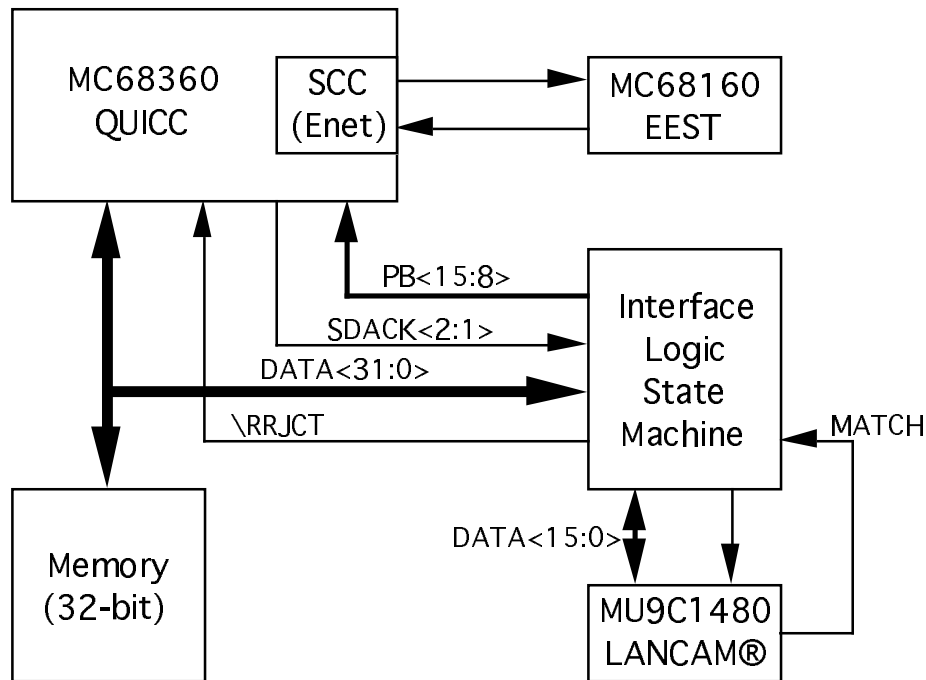
### Ethernet Address Filter Design Considerations

The key constraint for the external address filter is that the Receive Reject pin must be asserted before the end of the frame in question. Since the shortest Ethernet frame length is 64 bytes from the Destination Address up to and including the Frame check Sequence, it leaves only 58 bytes after the 6-byte Destination Address is captured. Therefore, using the serial interface option, the external address filter has a worst case of  $58 * 8 / 10\text{Mbps} = 46.4\mu\text{s}$  to respond with a Receive Reject output. The parallel interface option reduces the available response time since transfer to memory only occurs after address recognition, buffer descriptor lookup, assembly into 32-bit FIFO width, and memory bus arbitration. The worst case latency is bound by the 4-longword (SCC2) Receive FIFO depth, leaving 42 bytes (or  $33.6\mu\text{s}$ ) of response time. Longer frames would naturally allow much longer delay for the external address filtering hardware.

**External Address Filter Implementation Alternatives**

(1) CAM Address Filter

The fastest address filter is one based on a *Content Addressable Memory*. The address lookup is performed in parallel and a match condition can be obtained in one memory cycle. The MU9C1480 LANCAM®, for example, holds 1K 64-bit entries and can be cascaded to achieve deeper CAM size. The device has a 16-bit data bus interface and is packaged in a 44-pin PLCC. The QUICC's parallel CAM interface option can be used to pass the destination address to the LANCAM®, as shown in Figure 1. A state machine is required to decode the Serial DMA Acknowledge signals (SDACK<2:1>), latch the destination address from the 32-bit data bus and sequence writing the comparand register of the LANCAM® in 16-bit words. Based on the MATCH output and whether the CAM holds addresses to be accepted or rejected, either the \RRJCT pin is asserted or an associated tag byte is read from the CAM and output to PB<15:8>. The interface logic must also allow host access to the LANCAM® for address table maintenance. This implementation, however, can guarantee a response within the latency allowed by the shortest Ethernet frame, regardless of table size.



**Figure 1. MC68360 Parallel Interface to CAM Address Filter**

(2) DSP Address Filter

An alternative approach is to use a fast *Digital Signal Processor (DSP)* as an address filter (Figure 2). This approach off-loads the address filtering task from the main CPU by using a programmable processor with a fast instruction cycle time (30 to 50ns). The Ethernet destination address can be passed to the DSP serially, providing both advantages of a simple hardware interface and longer available latency. The DSP then uses a search algorithm to check if the frame is to be accepted. If not, the \RRJCT pin is asserted. If it is to be accepted, a Tag byte is output to the QUICC which will then be appended to the frame in memory. Communication with the CPU is done via an independent parallel Host Port on the DSP.

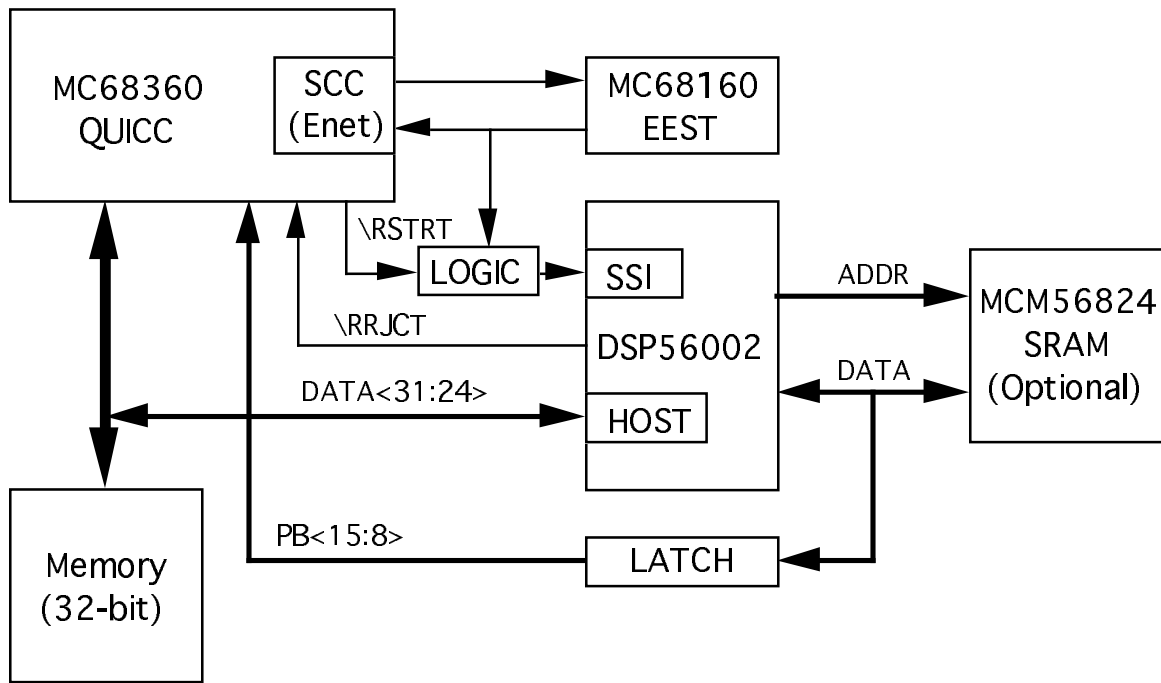


Figure 2. MC68360 Serial Interface to DSP Address Filter

The Motorola DSP56002 24-bit Digital Signal Processor is chosen in this approach for its fast execution time of 30ns (using a 66MHz device) and its wide 56-bit accumulator width which allows a full 48-bit Ethernet address comparison to be done in a single instruction. The device has internal Program and Data RAM of 512 words each, which are sufficient to hold both the address filtering code and up to 256 addresses (including the destination address itself). An optional single-chip MCM56824 static RAM expands the memory with an additional 8K words. The host port is used to bootstrap the DSP program RAM and to update the address table. The programmable nature of the DSP also allows more sophisticated learning algorithms to be implemented.

DSP Address Filter Design Details

The schematic and timing diagrams of the simple serial interface to the QUICC are shown in Figures 3 to 5.

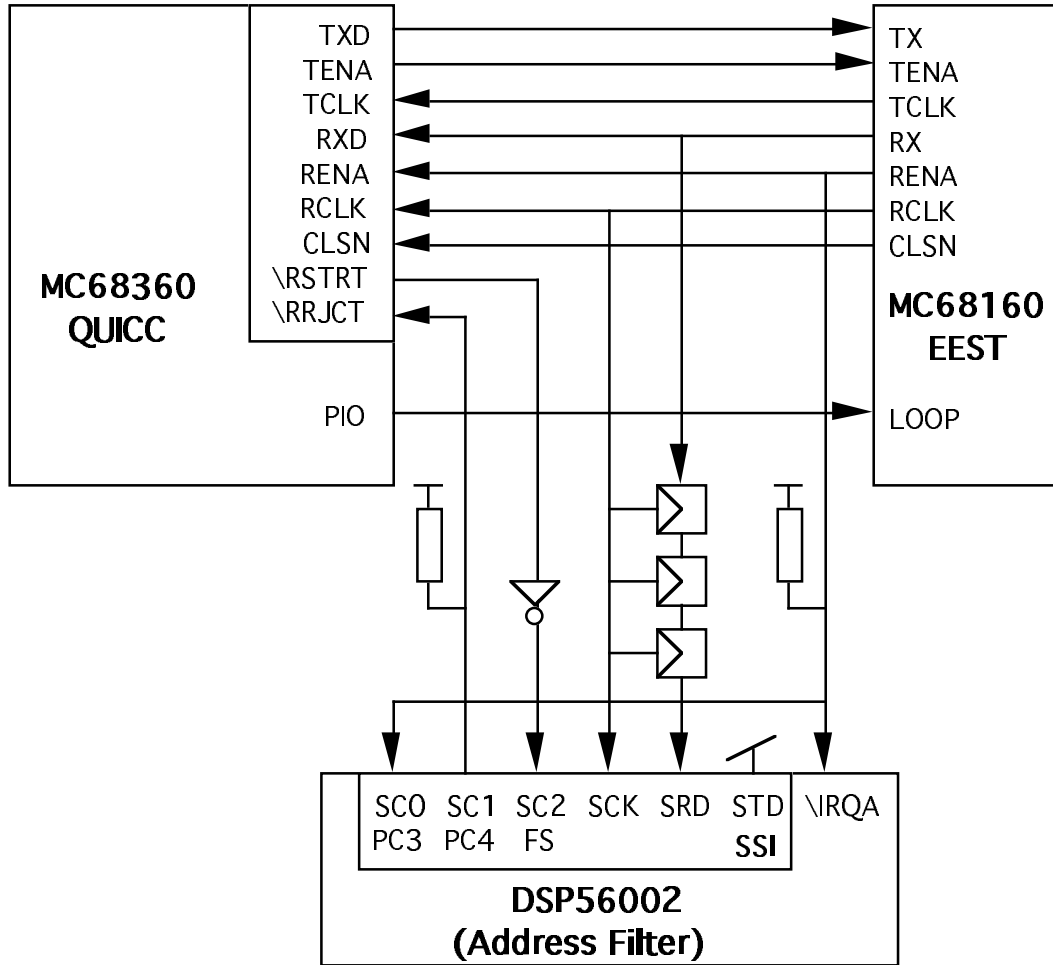
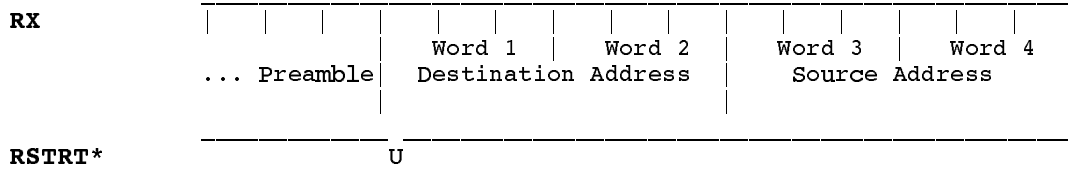


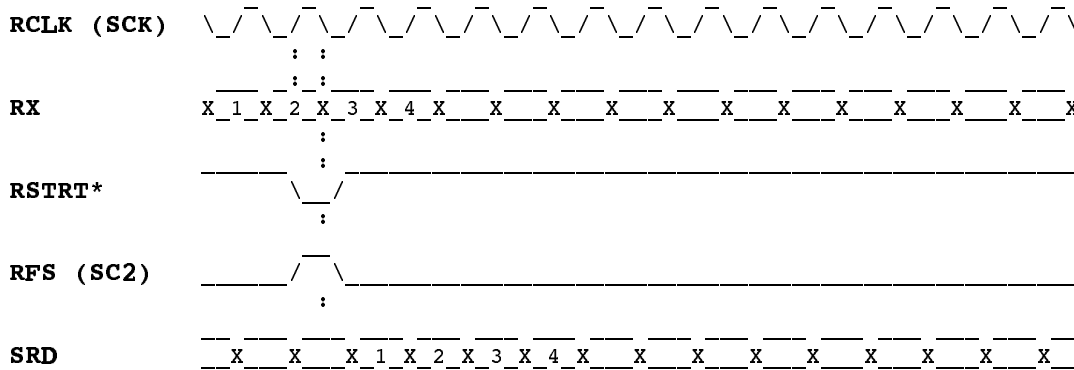
Figure 3. MC68360 to DSP56002 Serial Interface Logic

Since the Receive Start signal ( $\backslash$ RSTRT) is asserted during the second bit of the destination address field, but the DSP56002's Synchronous Serial Interface (SSI) requires a frame sync signal one bit time prior to the start of the receive word, the received data is first delayed by 3 bits. The inverted  $\backslash$ RSTRT (frame sync) signal, receive clock and delayed data are connected to the SSI which is set up to receive 24-bit words in timeslot (i.e. network) mode. After the first 2-word destination address is received, or optionally the following 2-word source address, the SSI is reset to wait for the next frame. When the address filter routine is completed, either the  $\backslash$ RRJCT Parallel I/O (PIO) pin is toggled to reject the frame, or the external Tag Latch is updated with the outgoing port

number. Should the frame finish before the end of the search routine, the routine is aborted by the Receive Enable (RENA) de-assertion interrupt, and the short frame is passed to the CPU for filtering.



**Figure 4. Serial Interface Byte Level Timing**



**Figure 5. Serial Interface Bit Level Timing**

The flow chart of a rudimentary address filter program is shown in Figure 6. After initialization, the program waits for the 2 word Destination Address field to be received by the SSI. The external Tag latch is initialized to an invalid value (\$FF) and then each address table entry is compared 48-bits at a time. If a match is found, the tag associated with the address table entry is written to the Tag latch. If the tag is \$00, then the \RRJCT pin is toggled to reject the frame. Should the frame end prior to the completion of the search, the \IRQA interrupt generated by the deassertion of Receiver Enable (RENA) aborts the search and reinitializes the program to be ready for the next frame. Any frame that leaked through to memory is flagged by an invalid tag byte of \$FF and can be later filtered by the CPU.

The core of the Address Filter routine is shown in Figure 7. The address table is structured to take advantage of the parallel architecture and addressing capability of the DSP56002. Each 48-bit address entry is stored as a long word consisting of both X and Y Data Memory locations. In order to utilize a tight loop with only an "equality" exit, the end of the search is detected by comparing the destination address with itself. The modulo addressing feature is used to automatically wrap the table pointer back to the beginning "exit" entry. The key instruction is the 56-bit Compare which also performs a parallel load of the next address table entry. The search loop takes 3 instruction cycles to

execute assuming the address table is stored in on-chip RAM. With an external RAM table, the loop takes an extra instruction cycle to transfer the 48-bit address via the single expansion bus. By pre-sorting the address table so that more frequently used "local" addresses are stored in internal memory and other addresses are stored in external memory, performance of the address filter can be optimized. The search algorithm thus statistically requires much less than the average of half the table to be compared. At 3 cycles or 90ns per address, the DSP56002 can go through roughly 500 address entries within the shortest Ethernet frame. By allowing the possibility of occasional overruns to be handled by the CPU, the pre-sorted address table can be much longer.

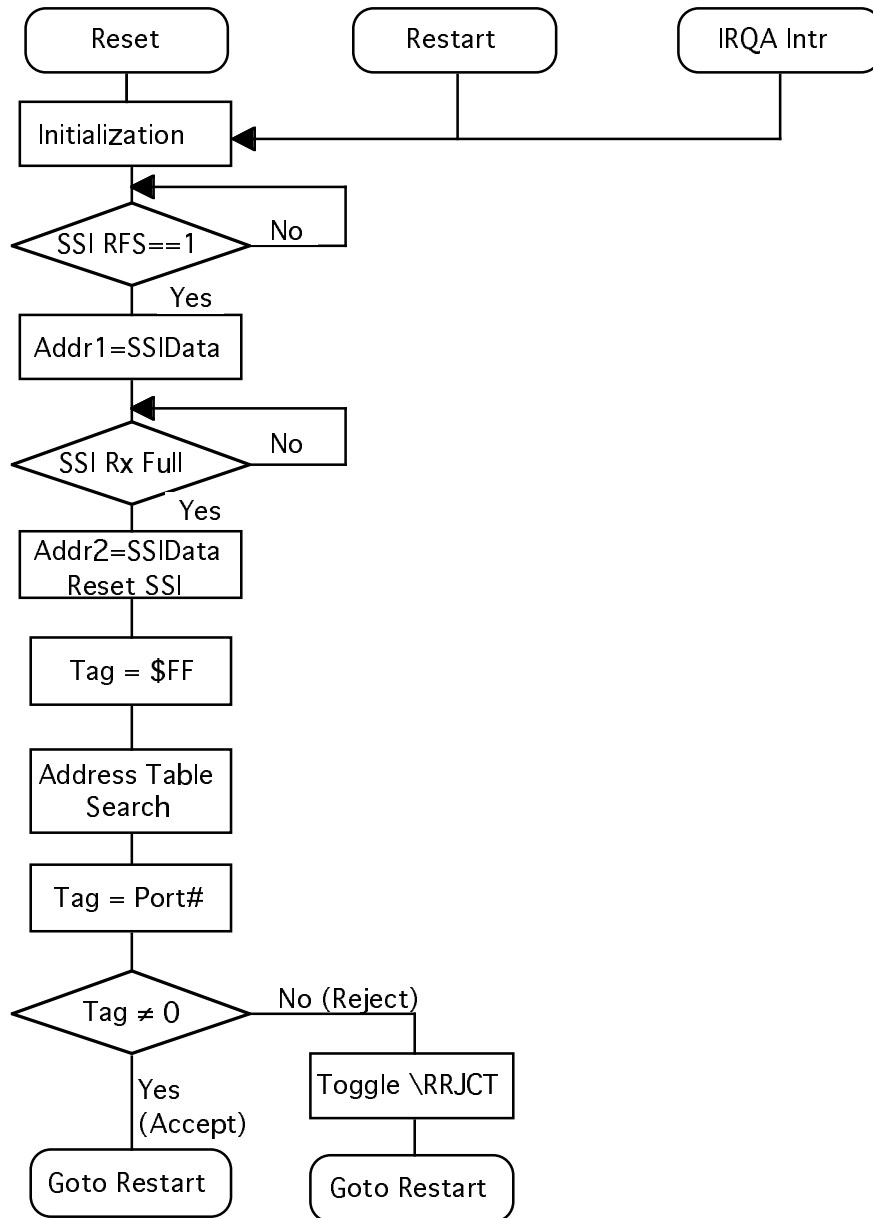


Figure 6. Flow Chart of Sample Address Filter DSP Program

```

;-----
; Address Table Structure :
;           X-Data Memory:           Y-Data Memory:
;           DstWord1                 DstWord2           <-----+
;           NotFoundTag
; (R0)-->  AddrWord1[0]               AddrWord2[0]
;           TagWord[0]                -
;           AddrWord1[1]               AddrWord2[1]
;           TagWord[1]                -
;           :                          :
;           AddrWord1[n-1]             AddrWord2[n-1]
;           TagWord[n-1]              -----+
;
; Let R0 = Pointer to Address Table entry [0]
; N0 = Pointer Offset = 2
; M0 = Pointer Modulus = Table Length (Incl. DstAddress) - 1
; R1 = Pointer to external Tag Latch
; X1 = #FFFFFF
;
; A = Input Destination Address in A1:A0
; B = Address Table Value in B1:B0
;-----

```

```

[1]          MOVE          X1,Y:(R1)          ; Set Tag to Invalid
[1]          MOVE          L:(R0)+N0,B       ; Load First 48-bit Addr
[1]  Loop    CMP           A,B                L:(R0)+N0,B ; Compare with Dest Addr
[2]          JNE          Loop                ; Repeat if not the same
[1]          MOVE          (R0)-N0           ; Move Ptr back 1 Entry
[2]          MOVE          X:-(R0),A         ; Read Tag Word
[1]          TST          A                   A,Y:(R1)   ; Write to Tag Latch &
[2]          JNE          Start                ; Done if accepted
[2]          BCLR         #4,X:PCDATA        ; Else Reject frame by
[1]          NOP          ; Asserting PC4=\RRJCT
[1]          NOP          ; For >= 1bit time
[2]          BSET         #4,X:PCDATA        ; De-Assert PC4=\RRJCT
[2]          JMP          Start                ; Go handle next frame

```

Figure 7. Core of Address Filter Routine

Summary

The DSP56002 offers QUICC Ethernet bridge/router designers a flexible and cost effective option for address filtering. Depending on the required address table size, one of the alternatives listed in Figure 8 can be chosen. For example, a small bridge with up to 255 address table entries would only require the lower cost 40MHz DSP56002FC40 using only on-chip memory. With external memory, the 60MHz device can handle 500 entries with no overrun, and much larger tables if occasional CPU assistance is acceptable. Multiple DSP56002's can also be used in parallel to increase performance at large address table sizes. The CAM approach is more attractive if guaranteed response is desired in large bridge/router applications.



<i># Address Entries</i>	<i>Implementation</i>	<i>External Memory</i>	<i>Overflow Possibility</i>
≤ 255	DSP56002FC40	No	None
≤ 500	DSP56002FC66	Yes	None
> 500	DSP56002FC66	Yes	Occasional
≤ 1024	MU9C1480	No	None
≤ N x 1024	N x MU9C1480	(Multiple CAM's)	None

**Figure 8. Address Filtering Alternatives**

## Acknowledgement

The author would like to thank Hugh McGugan and Robert O'Dell for their reviews and helpful comments.

## References

- [1] MC68360 Quad Communications Controller User's Manual, MC68360UM/AD, Motorola Semiconductor Products Sector, 1993
- [2] The MU9C1480 LANCAM® Handbook, Music Semiconductors, 1992
- [3] DSP56000 Digital Signal Processor Family Manual, DSP56KFAMUM/AD, Motorola Semiconductor Products Sector, 1993
- [4] DSP56002 User's Manual, DSP56002UM/AD, Motorola Semiconductor Products Sector, 1994
- [5] DSP56002 Data Sheet, DSP56002/D, Motorola Semiconductor Products Sector, 1992

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document. Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.



(c) Copyright Motorola, 1994

QUICC™ Ethernet Address Filter Design Concept

**For More Information On This Product,  
Go to: [www.freescale.com](http://www.freescale.com)**