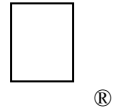


New Technical Notes

Macintosh



Developer Support

Byte Smear(ing) Tactics

Hardware

Written by: Dave Radcliffe & Carl Hewitt

August 1990

This Technical Note discusses a feature of the current Macintosh hardware which will not be supported in the future. Macintosh hardware developers and driver writers should be aware of this limitation as it affects current and future products.

Current Macintosh hardware supports a feature of MC68020 and MC68030 processors which is herein referred to as “byte smearing.” Future Macintosh platforms may not support this feature, and if you have hardware or software dependent on this feature you need to revise it. Likewise you should be aware of this limitation if you are developing new hardware or software.

Spreading the Bytes Around

MC68020 and MC68030 processors have a “feature” which causes the data for byte and word transfers to be replicated (smeared) across all 32 data lines. An example illustrates the problem.

Consider the following code:

```
MOVE.L    #$12345678,D0        ; Stuff some data
MOVE.B    D0,$102              ; Write a byte of data
```

The data actually placed on the data bus, with and without byte smearing, is shown in Figure 1.

With Byte Smearing

CPU Bit	31	24	16	8	0
Byte Data	\$78	\$78	\$78	\$78	
Byte Address	\$100	\$101	\$102	\$103	

Without Byte Smearing

CPU Bit	31	24	16	8	0
Byte Data	xx	xx	\$78	xx	
Byte Address	\$100	\$101	\$102	\$103	

Figure 1—Effect of Byte Smearing

With byte smearing, the byte of data is replicated across all the byte lanes; without smearing, the other bytes are undefined. A similar replication of data can occur with word transfers.

As an example where this can cause trouble, suppose you have a NuBus™ card with a device register which expects to be byte addressed at byte \$102. With byte smearing it is actually possible to get away with writing a byte to any address from \$100 through \$103; without byte smearing, the card only sees the correct data when addressed at the correct byte \$102.

Conclusion

The lack of “byte smearing” as a feature should not be a problem for most developers; after all, why would anyone write to byte \$100 when they really meant to write to byte \$102? Well, sad to say, at least one case of this happening has been uncovered, so if you have, either inadvertently or by design, relied on this feature, you should revise your products to run on future Macintosh platforms.

Further Reference:

- MC68020 User’s Manual, pp. 7-9
- MC68030 User’s Manual, pp. 7-9

NuBus is a trademark of Texas Instruments.