AESOP

Topic Name: M68000 FAQs (by Device)
Subject:M68000 Family FAQ
Entry Prepared On: Jul 22 17:17:31 1994

Applies to all processors in the M68000 Family

QUESTION
Do the M68000 family processors have any internal bits that indicate how many external floating-point units are active?

ANSWER
There are no internal bits for this purpose. The approach taken in finding out how many floating-point units (or coprocessors) are active is with the coprocessor protocol violation exception. If a coprocessor is not present and an access is attempted, a bus error is generated via a bus-error time out (watchdog timer) and a coprocessor protocol violation occurs. If the coprocessor protocol violation occurs, then the floating-point unit (or coprocessor) is not present.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:04 1994

Topic Name: M68000 FAQs (by Device)
Subject:M68040 FAQ
Entry Prepared On: Jul 22 17:41:34 1994

QUESTION
The M68040 User's Manual states that contiguous memory blocks must be 32 bits wide. Does this imply that 16-bit bi-directional transfers cannot be accomplished on bytes 2 and 3 or bytes 2 and 1 without the aid of the bus sizer?

ANSWER
See the M68040 User's Manual (M68040UM/AD) for the data bus requirements for read and write cycles. Word transfers can occur from bytes 2 and 3 or bytes 1 and 0. Also, the MC68040 supports misaligned operands. The manual provides an example of a misaligned word transfer.

QUESTION
What happens to the MC68040 data bus when BG* is negated?

ANSWER
The data bus remains three-stated until BG* is asserted and BB* is negated.

QUESTION
Are there cases other than a MOVES instruction that can cause an alternate function code access (i.e., TT1=1 and TT0=0) on an MC68040?

ANSWER
No, alternate function code accesses only occur as a result of the MOVES instruction.

QUESTION
Which MC68040 signals should be used for address decoding to prevent address glitches?

ANSWER
On the rising edge of BCLK with TS* asserted and the address lines valid, generate the chip select.

QUESTION
What is the order in which exception stacking for an interrupt occurs?

ANSWER
1) An interrupt acknowledge cycle is performed
2) Write of the format word
3) Write of the program counter high word
4) Write of the program counter low word
5) Read of the vector table (long word)
6) Write of the status register word

QUESTION
Is it possible to predict when an interrupt acknowledge cycle is about to occur?

ANSWER
Immediately before the start of an IACK cycle, assuming that the caches and MMU are disabled, there is one instruction prefetch (ending with the assertion of TA or TEA) that occurs while PST3-PST0 indicate 1111. However, this is only a best guess and Motorola does not guarantee that this will work in all cases. If a case is found which contradicts this guess, please contact Motorola immediately at (512) 891-4524. Also note that this property was not a design goal, but just happened to be a result of not requiring the instruction side to be idle prior to the beginning of the exception stacking sequence.

QUESTION
Does Motorola provide a chip-level simulator for the MC68040?

ANSWER
Motorola does not provide a chip-level simulator for use outside of Motorola.

QUESTION
Can MC68040/MC68EC040 operation be guaranteed for clock frequencies below 16.67 MHz?

ANSWER
Due to resolution limitations with the digital phased-locked loop used in the current MC68040 and MC68EC040 designs, the MC68040/MC68EC040 is not
guaranteed to operate at frequencies below 16.67 MHz.

QUESTION
Does the MC68040 require two synchronized clocks, one at 50 MHz to PCLK and
one at 25 MHz to BCLK, or can two separate asynchronous oscillators, one at 50 MHz to PCLK and one at 25 MHz to BCLK, be used?

ANSWER
The PCLK and BCLK clock signals must be synchronized, with a relatively stable phase relationship. Typically, BCLK would be generated by running PCLK through a divide-by-two stage. Note that at 25 MHz, the allowable skew between PCLK rising edge and BCLK rising edge skew is 9 ns. max. For more information, see the M68040 User's Manual (M68040UM/AD).

QUESTION
I have programs which seem to run okay with the MC68040 data cache in copyback mode, but do not run with the data cache in write-through mode. What would cause this?

ANSWER
When the data cache is configured in write-through mode, many more writes occur than when it is in copyback mode. Check your write timing parameters, especially those which assume back-to-back write cycles (because these will be more prevalent in write-through mode).

QUESTION
My MC68040 locks up under the following circumstances:

1) The instruction cache is on and the data cache is off. If the instruction cache is off, there
   is no problem.
2) The MC68040 does a cache line burst read.
3) During that burst read, another device (an ethernet) is granted the bus
4) The MC68040 completes the read and the ethernet does two bus cycles
5) At the end of the second cycle, the MC68040 has BR* and BG* asserted
6) The MC68040 locks up - it never takes the bus (although it delivers the correct address).
   What could cause this problem?

ANSWER
The MC68040 should not lock up when the bus is taken from it.

Potential causes include:

1) Check the status of the BB* pin to the MC68040. If BB* is asserted, the MC68040 will not take the bus.
2) The series of events just prior to the arbitration sequence will probably be the most important, so pay particular attention to those cycles.
3) Verify that the MC68040 is not snooping without receiving the signal of the end of a bus cycle.
4) Refer to the MC68040 User's Manual (MC68040UM/AD) for information on the implications of implicit bus ownership. Perhaps some of the issues discussed apply to your system.
5) Verify that your clocks meet the timing specifications. A bad input clock has been known to cause the MC68040 to lock up.
6) Verify that the timing specs for bus arbitration are being observed.

QUESTION
How do you make an address strobe for an MC68040?

ANSWER
The MC68040 does not generate an MC68030-like address strobe. The transfer in progress (TIP*) signal can not be used as an address strobe because the

addresses are not guaranteed to be valid when TIP* asserts and the TIP* is not guaranteed to negate at the end of a bus cycle (TIP* remains asserted on back-to-back accesses). The user can create an address strobe (user address strobe => UAS*) by asserting the UAS* off the rising edge of the BCLK that TS* is asserted. The UAS* is negated off the BCLK edge in which the final TA* is asserted or the edge that TEA* is asserted. Assuming no retries or bus errors, the final TA* assertion is the first TA* assertion if (1) the transfer size is byte, word, or long word, or (2) the transfer size is line and TBI* is asserted. The final TA* assertion is the fourth TA* assertion if the transfer size is line and TBI* is negated.

Another alternative is to use the MC68150 dynamic bus sizer. This part generates an AS*-like signal from the signals asserted by the MC68040 and provides dynamic bus sizing capability for the MC68040.

QUESTION
On the MC68040, if no TTR register matches the physical address on an instruction or data fetch, what is the default caching mode?

ANSWER
If the physical address does not match any of the TTR registers, the processor assumes the default caching mode, write-through. Therefore, you can make the most use of the MC68040 TTR registers by defining the noncaching and copyback sections of your memory map and allowing the rest of the memory map to default to the write-through mode.

QUESTION
For PCLK on the MC68040, do you have to meet minimum pulse width numbers only, or do the maximum numbers matter as well?

ANSWER
You must meet all of the specs for PCLK on the MC68040 in order to insure proper operation of the part. The MC68040 has fairly tight specs for this clock and we recommend the use of the MC88915 and MC88916 clock drivers which provide both PCLK and BCLK and are known to work very well with the MC68040.

QUESTION
If the MC68040 is set up to snoop the bus and a snoop miss occurs, how many clocks after TS* will the MI* signal remain active?

ANSWER
When the bus is arbitrated away from the MC68040, it asserts the MI* signal. When the new bus master asserts TS*, the MC68040 will attempt to source the data for the access from within its cache. If snooping is turned off or if the data corresponding to the access is not in the cache (snoop miss), then the processor will negate MI* in 2 to 4 clock cycles. The reason there is some uncertainty here is that the amount of time required to determine whether or not MI* should be released depends on what state the MC68040 is in at the time the TS* assertion occurs. The processor may be using the cache at the time, and it could take up to 2 clocks before it releases it to the snoop controller. The snoop controller may then take up to 2 clocks to decide that it should negate the MI* signal. It should be noted here that after the bus cycle is completed (after the last TA* assertion), the MC68040 will again assert the MI* signal.

QUESTION

Does the MC68040 processor have an on-chip floating-point unit?

ANSWER
Yes, the MC68040 has a floating point unit on board. However, it does not support all of the instructions available on the MC68882 (for example, SIN and COS are not supported). Also, there are some unsupported data types in the MC68040 floating-point unit. These include DENORM, UNNORM, and PACKED
BCD. The floating-point software package (FPSP) for the MC68040 (available separately) provides full IEEE floating-point standard compliance. Please see the M68040 User's Manual (M68040UM/AD) floating-point unit section for details on what the FPSP provides.

QUESTION
The MC68040 cache is a physical cache, meaning cache addresses are physical addresses. I am addressing the same physical address from two different virtual addresses. One virtual address uses the copyback cache mode and writes something onto the physical memory. Then, from the second virtual address, which uses cache inhibited serialized mode, I read the same physical memory. The contents I read is not the same as what I wrote using the first virtual address. But if I do a CPUSH to flush that particular page to memory before I read, I read the correct value. Please explain.

ANSWER
When the MC68040 is in copyback mode and a write occurs, it reads in a cache line from memory into the cache, sets the valid bit for that line, then performs the write and marks that particular long word as dirty. At this point, only the cache has the most current copy of that particular memory location. The only time the external memory is updated is when a replacement of that cache line is made, or when a CPUSH instruction is executed to reconcile the cache with memory.

When you access the address in the cache using a cache-inhibited page descriptor, the processor bypasses the cache and directly fetches the operand from external memory, hence, explaining the stale data that you are getting.

This is a normal operation of the MC68040.

QUESTION
On an MC68040, is a misaligned long-word access involving multiple bus cycles completed in one bus tenure, or can the bus be arbitrated away?

ANSWER
Bus arbitration is independent of the type of bus access. On a misaligned long word in which the operand transfer is completed in more than one bus cycle, the bus may be arbitrated away after the first bus cycle. When the bus is returned to the processor, the processor continues to complete the operand transfer.

QUESTION
Please discuss the activity of the MC68040 TLNx lines during a normal burst mode line operation.

ANSWER
The TLNx signals are useful only on data line reads.

The TLNx signals indicate which line in the set of four data cache lines is being accessed. The TLNx signals are only valid during line read operations. During line read operations, the cache-line becomes allocated in the cache. Line reads caused by the MOV16 instruction do not allocate into the cache; therefore, the TLNx signals must be ignored in this case. The TLNx signals must also be ignored on any instruction prefetches. Byte, word, or long-word accesses indicate a cache-inhibited access.

QUESTION
Can the MC68040 and MC68EC040 be interchanged in the same footprint?

ANSWER
Yes and No. If a board were designed for an MC68EC040, the MC68040 should fit in with no problems, with the exception of a slightly higher power dissipation. On the other hand, placing the MC68EC040 into an existing MC68040 socket may present some problems because there are modes such as data latch enable and multiplexed bus modes and buffer sizes that exist on the MC68040, but are unsupported on the MC68EC040. Additionally, software differences and the fact that the JTAG BSDL file is slightly different could also cause problems.

QUESTION
How do I get better than 16 Mbytes (MC68EC040) or 4 Kbytes (MC68040) resolution in the choice of a caching mode?

ANSWER
A possible solution is to divide the 4-Gbyte addressing range into four 1-Gbyte aliases. For instance if a peripheral is located in $0ABC0000, it is "aliased" to locations $4ABC0000, $8ABC0000, and $CABC000 if address bits A31 and A30 are ignored when generating chip selects for that peripheral.

The A31 and A30 address bits can now be used to indicate the cache mode of that particular address range. The access control registers (MC68EC040) or the transparent translation registers (MC68040) may then be used to configure each alias to be of a certain cache mode.

The following is a write-up that describes in more detail how this may be achieved:

This article discusses a method for selecting caching modes on 16 byte boundarys for a 68EC040. The 68EC040 delivers high performance at a low system cost for embedded control by providing the same high integer performance and large 4k instruction / 4k data caches as the 68040 without the floating point unit and memory management unit. The 68040Us MMU makes the caching mode selectable for each 4k or 8k page within memory. The use of several caching modes within the same address range is made more difficult without the MMU. The access control unit of the 68EC040 provides two access control registers each for data and instructions. Each access control register allows caching modes to be defined in 16M byte to 4G byte sections. This coarse division of the memory map is not ideal for all embedded applications. This article explains an addressing scheme that allows all parts of the memory map to be independently available using any of the EC040 caching modes on line boundaries (a line equals 16 bytes). Since any part of the memory map is accessible as cache inhibited, copyback or write through, there is no requirement to split caching modes on 16M boundaries. Further, since the cache mode can be selected down to the line

boundary, different areas within the same address region (e.g. DRAM) are accessible in any or all of the three caching modes.

Two address bits are used to divide the 68EC040's 4G byte addressing range into four 1G byte sections. The only difference between the sections is the caching modes. The caching mode of one section is made cache inhibited-serialized (address bits = 00), the next section is made cacheable-copyback (01) and the last two sections are made cacheable-write through (1X). The address bit ordering (00, 01, 1X) allows the 1 G byte sections to be nested. Address bits = to 00 are at the bottom of the address map, address bits 01 are in the middle and address bits 1X occupy the top half of the address map. Each of the 1G byte sections are mirrored on to every other section to provide a single 1G byte addressing range. The address mirroring is done by externally ignoring the two address bits used to select the caching mode. The regions are mapped into memory using the remaining 30 address bits. The caching mode of any part of the 1G byte address range is now selected by software. If the address bit used for cache mode selection are 00, then the access is cache inhibited. If the address bits are 01, then the access is a copyback access. If the address bits are 10 or 11, then the access is write through. The address bits can be any of the top 8 address bits (A31 - A24) and do not need to be contiguous. This cache addressing method does not provide any internal protection from incorrectly accessing an address with the wrong caching scheme. The answer is to rely on the software to correctly access with the correct caching mode or to externally qualify the accesses with the caching mode address bits. Special care is required to avoid mixing caching modes within the same memory line. An example of the problem is a cacheable and non-cacheable access to the same line. A copyback access to one long word of a line will cause all four long words of the line to be read into memory. A cache inhibited access to another long word of the same line would not hit in the cache, but in the external memory. A cache line push of the copyback line can now overwrite the cache inhibited long word in external memory. A subsequent memory access to the cache inhibited long word of the same line will now differ based on whether or not the cache push of the line has occurred. The solution is to use line boundaries when choosing caching modes. If you want to change the cache mode of a line from cacheable to cache inhibit, do a CPUSH or CINVAL of the line when making the change. This insures the cache does not contain a copy of a cache inhibited line. To switch from cache inhibited to cacheable, the internal caches do not require any change.

To better understand how this fits within the cache operation of the 68EC040, here is a brief explanation of how the 68EC040 caches and access control unit works. When the 68EC040 comes out of reset, the caches and access control units are disabled and all accesses are in cache inhibited, non-serialized mode. When the caches are first enabled, the information in the cache is unknown so be sure to invalidate the caches before enabling. The caches are enabled by accessing the cache access control register (CACR). The data access control unit is enabled by enabling one or both of the data access control registers. When a data access is made, the 68EC040 compares the upper 8 bits of the address to the base address and address mask of the enabled data access control registers (DAC0 and DAC1). An address match with the data access control register occurs if the upper 8 bits of the access address matches the base address or is masked off by the data access control register address mask. If the address does not match either data access control register, the caching mode is the default (cache inhibited, non-serialized if the cache is disabled, write through if the cache is enabled). If the address does match one of the data access control

registers, the cache mode bits of the matching data access control register select the caching mode. If both data access control registers match, DAC0 takes priority over DAC1. The instruction accesses work the same, except the instruction access control unit is used.

To enable the multiple caching modes, enable DAC0 and IAC0 for cache inhibited - serialized (cache mode bits = 10 for cache inhibited serialized), mask out (set to 1s) all but two address bits and set the remaining two address bits to the cache inhibited - serialized region (e.g. address bits equal to 00). Enable DAC1 and IAC1 for cacheable - copyback (cache mode bits = 01 for cacheable - copyback), mask out (set to 1s) all but two address bits and set the remaining two address bits to the cache inhibited - copyback region (e.g. address bits equal to 01). The write protect bit, user page attribute bits and function code bits are set as the user requires for both DAC0 and DAC1. When the caches are enabled, all accesses in which the non-masked address bits are 00 use DAC0 or IAC0 and are cache inhibited - serialized. All accesses in which the non-masked address bits are 01, use DAC1 or IAC1 and are cacheable - copyback. All accesses in which the non-masked address bits do not match either access control regiser, are not affected by the access control units and default to cacheable - write through.

Here is the code for enabling the access control unit for this caching scheme. Depending on the assembler used, substitute DTT and ITT for DAC and IAC.

```
MOVE.LD0, -(A7)
CINVABC
MOVE.L#$003FC020,D0
MOVECD0, DAC0
MOVECD0, IAC0
MOVE.L#$403FC010,D0
MOVECD0, DAC1
MOVECD0, IAC1
MOVE.L#$80008000,D0
MOVECD0,CACR
MOVE.L(A7)+,D0
```

QUESTION
Is the metal flash on the lid of the MC68040 PGA grounded through the package?

ANSWER
No, the metal flash on the lid of the MC68040 is not grounded.

QUESTION
If the MC68040 is in burst mode and accesses memory that is cache inhibited, will the processor fill the cache line?

ANSWER
The MC68040 will never cache data from an area that is designated as cache inhibited. In fact, the only time the MC68040 will burst from memory that is cache inhibited is when a MOVE16 instruction is executed. If both the source and destination for a MOVE16 instruction reside in cache-inhibited memory, the MC68040 will burst four long words from the source into a buffer and then burst the same four long words from the buffer to the destination. At no time is any space in the cache allocated for such a

move.

QUESTION
I am interfacing an SRAM to the MC68040. The SRAM requires a toggle of the WE* signal to latch the data. During the burst cycle of the MC68040, the R/W* signal stays low for the duration of the cycle. How can I create a toggling WE* signal for my SRAM?

ANSWER
I do not know the specific timing requirements of your SRAM, but you might try creating a WE* signal by qualifying the R/W* signal with BCLK. In other words, if the R/W* signal from the MC68040 is low and BCLK is low, then your WE* signal to the SRAM will be low. By doing this, the WE* signal will toggle and latch data. This may not be the exact solution your design requires, but something similar to this will probably work.

QUESTION
I am observing out-of-order execution of reads and writes on my logic analyzer traces of my MC68040-based system. My peripheral cannot tolerate this out-of-order execution. Is this out-of-order execution supposed to happen? If so, how do I stop it from happening to this peripheral?

ANSWER
The out-of-order execution you observe is not unusual because the MC68040 generally prioritizes reads over writes to improve overall performance. The MC68040 provides two ways of fixing the problem with your peripheral. One way is to use the transparent translation registers to designate the peripheral address as cache inhibited serialized. The second way is to create page tables to designate that peripheral address as cache inhibited serialized.

If it is possible to change the source code, there is actually a third way to solve the problem. NOPs should be placed after the instruction that causes a write to that peripheral to force the completion of writes.

QUESTION
Can we run a 25 MHz MC68040 with a 20 MHz clock? How do the bus specifications change?

ANSWER
The 25 MHz MC68040 can run at 20 MHz, but keep in mind that 20 MHz is the minimum guaranteed frequency for this device. All input setup and hold times remain the same. All output times relative to clock edges can be adjusted as follows:

(Spec at 25 MHz) - (9 ns) + (11.5 ns) = (Spec at 20 MHz)

QUESTION
What is the input capacitance on BCLK and PCLK for the MC68040?

ANSWER
The typical input capacitance is 20 pF for the MC68040. The maximum input capacitance is 25 pF.

QUESTION
Where can I get RC termination packages for the MC68040? What values should I use?

ANSWER
Contact the manufacturer, Sprague at BP Sales (512) 341-9186. Ask for RC networks.

Termination values highly depend on board characteristics. Contact the applicable PCB manufacturer for recommendations, or manufacture a board with sockets and try different values.

QUESTION
What is the cache access latency of the MC68040 when the MC68040 is snooping alternate bus master accesses?

ANSWER
The latency will vary from 1 to 2 clocks. During this time, the MC68040 will assert MI* (memory inhibit) to indicate to the slave memory subsystem not to close the current access.

QUESTION
When the MC68040 is snooping an alternate bus master burst read and an inconsistency is found in the MC68040 data cache, will the MC68040 allow the burst?

ANSWER
First, a couple of general statements regarding snooping:

The MC68040 cache may operate in either copyback or write-through mode. When operating in write-through mode, there is never stale data in the cache since the write-through operation is synchronous. When the MC68040 is snooping, the cache tags will be compared, and if there is a write by the alternate master, then the MC68040 will invalidate the entire cache line.

When the copyback mode is invoked and a cache hit occurs, the MC68040 can be configured to assume bus slave mode and source the current cache data. The MC68040 will burst the entire cache line.

Our experience indicates that when snooping is active, the preferred cache mode of operation is write through. The memory subsystem interface is greatly simplified, since it needs only to respond to MI* in addition to the normal transfer signals. The copyback mode requires a more complex memory control to allow the possibility of the MC68040 sourcing data during a snoop (as described above). Customer experience suggests that the performance delta is negligible and not enough to warrant the additional complexity.

To specifically answer the question, the MC68040 will never assert burst inhibit when it is sourcing data.

QUESTION
Can a memory system which responds to burst-mode only be implemented?

ANSWER
No. Even with the caches active, there are specific types of machine cycles that the MC68040 will initiate that will not result in a burst. Among them are exceptions processing stacking, for example.

QUESTION
Please discuss clock stability at power-up.

ANSWER
The M68040 User's Manual (M68040UM/AD) states that neither BCLK or PCLK inputs can be gated off at any time during the operation of the MC68040. Failure to apply a valid clock might, in some cases, result in quality or reliability problems for the MC68040. For most cases, providing a stable clock within specifications is not an issue, but some concerns arise for a power-up condition.

Specifications exist which require Vih to be less than Vcc on the MC68040. This prohibits the possibility of driving the clocks to the MC68040 before power is applied. AC specifications introduce a new problem of how soon the clock must be valid after power-up. The following table shows maximum times the processor can go without
clocks:

Clock	Fmin	Period at Fmin	wc Duty Cycle	T max no clock
BCLK	16.67 MHz	60 ns	60%	36.00 ns
PCLK	33.33 MHz	30 ns	52.5%	15.75 ns

This implies that the MC68040 must receive a valid PCLK within 15.75 ns and a valid BCLK within 36 ns of Vcc being within spec as illustrated in the attached illustration.

Providing stable clocks so soon after power up can be a problem. Many clock driver manufacturers do not provide specifications describing the operation of their chips for power-up conditions. This document attempts to provide an understanding of Motorola's philosophy on this issue.

While chip designers attempted to ensure that unstable nodes were eliminated, the possibility exists that without a valid clock signal to clear internal nodes at power up, a high current path can exist internal to the MC68040. Such a high current path could theoretically damage sensitive circuitry. While there is the possibility of causing damage, no damage has been seen in the factory due to a failure to clock an MC68040.

Limitations in automated test equipment makes it unfeasible to test every MC68040 Motorola manufactures to determine the maximum time it can survive without a clock. However, characterization data from both a lab setup and automated test equipment suggests that during power up an MC68040 can last 100 us without a stable clock. This is not to imply that this is a relaxation of the time-without-a-clock specification; the specification still stands as a maximum of 15.75 ns without a clock transition. Properly designed clock driving circuitry can provide a valid clock within less than 10 us.

QUESTION
Does the duty cycle specification change when running a 25 MHz MC68040 at 20 MHz?

ANSWER
No, the duty cycle is very important for proper clock design. The phased-locked loop of the MC68040 is very sensitive to input duty cycle on the clock. Simply meeting the min/max specs without meeting the duty cycle is unacceptable.

QUESTION
Does the duty cycle specification change when running a 25 MHz MC68040 at 20 MHz?

ANSWER
No, the duty cycle is very important for proper clock design. The phased-locked loop of the MC68040 is very sensitive to input duty cycle on the clock. Simply meeting the min/max specs without meeting the duty cycle is unacceptable.

QUESTION
When the MC68LC040 is doing byte or word write cycles, are unused sections of the bus driven?

ANSWER
When the MC68LC040 is performing byte or word write cycles, the unused sections of the bus are driven. There is no guarantee as to what is on the unused sections of the data bus. For a read byte or word cycle, if the unused sections of the data bus are driven, they are ignored.

QUESTION
On the MC68LC040, are TCI* and TBI* only sampled during TA*? Is their trailing edge phase with relation to BCLK any problem if they come up when TS* is asserted?

ANSWER
TCI* and TBI* are only sampled during the first valid TA*. There is not a problen if TCI* or TBI* is asserted when TS* is asserted.

QUESTION
The AC timing of the MC68EC040 is measured "at the pin." Does this refer to the pin of the MC68EC040 or the receiving buffer?

ANSWER
The MC68EC040's pin.

QUESTION
Could you specify where timing spec #17 (BCLK to Data-In High-Z for a read followed by a write) is measured? It does not appear in any figure.

ANSWER
Spec #17 should be drawn in the read/write timing diagram on the D31-D0 IN (READ) data lines. It starts at the rising edge of the clock and ends where the data goes to high impedance. (Find spec #16 and then stretch #16 to where data goes to high impedance.)

QUESTION
Does the MC68EC040 insert a wait state between a read cycle and a write cycle?

ANSWER
No, the MC68EC040 does not insert a wait state. However, due to the high hit rate of the MC68EC040's caches, there are many idle states on the bus.

QUESTION
Is it true that the MC68EC040 does not do alternate logical function code accesses?

ANSWER
The MC68EC040 supports alternate function code accesses. Alternate function code accesses via the MOVES instruction are valid only for those encodings that are not defined as either supervisor code space, supervisor data space, user code space, or user data space.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:05 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68000/HC000/008/010 FAQ
Entry Prepared On: Jul 22 17:34:53 1994

QUESTION
Is it okay to tie the HALT* and RESET* lines together on the MC68000 together?

ANSWER
Tying HALT* and RESET* together will not cause any problems for the MC68000.As you probably know, RESET* on the MC68000 should never be asserted without asserting HALT* as well. You must realize, however, that tying the two lines together introduces the following three restrictions:
1) You can not do hardware single-stepping of the processor.
2) You can not perform a retry.
3) The processor will reset itself if you get a double bus error (by asserting HALT*).

Many systems can function within the bounds of the above restrictions and operate without problems when HALT* and RESET* are tied together.

QUESTION
Should I assert RESET* on the MC68000 without asserting HALT* as well?

ANSWER
On the MC68000/MC68EC000/MC68008 processors, RESET* should never be asserted without asserting HALT* as well. You may bring about the assertion of HALT* with RESET* through external logic or by tying the two lines together. You must realize, however, that tying the two lines together introduces the following three restrictions:
1) You can not do hardware single-stepping of the processor.
2) You cannot perform a retry.
3) The processor will reset itself if you get a double bus error (by asserting HALT*).

QUESTION
Is it okay to assert the MC68000 VPA* and DTACK* inputs at the same time?

ANSWER
DTACK* should never be asserted while VPA* is asserted. Asserting DTACK* and VPA* at the same time will produce unpredictable results.

QUESTION
Will the MC68000 support 4 megabytes of DRAM using 1x9 1-Mbyte SIMMs?

ANSWER
You should have no problems interfacing standard DRAM SIMMs to the MC68000.Note that the MC68000 does not burst so you do not need any special type of DRAM such as page-mode DRAM which would normally be required to support bursting.

QUESTION
What kind of cache design would you recommend for the MC68000?

ANSWER
A system can be designed around an MC68000 which uses relatively cheap (80 ns or 100 ns) DRAM and yet provides zero wait-state accesses. Therefore, we do not recommend using a cache with the MC68000.

QUESTION
How do you perform a long-word write with an MC68000 with a 16-bit bus?

ANSWER
The MC68000 instruction set supports long-word moves. When the processor executes an instruction that calls for a long-word write, it will perform two 16-bit writes. Note that bus arbitration is done on a cycle-by-cycle basis, so the bus can be arbitrated away from the MC68000 after the first of the two 16-bit writes, and the processor will complete the long-word write after it retakes the bus.

QUESTION
What is the difference between an MC68000 and an MC68008?

ANSWER
1) The MC68000 is a 16-bit processor, whereas the MC68008 is an 8-bit processor.
2) Therefore, UDS* and LDS* on the MC68000 are replaced by DS* and A0 on the MC68008.
3) The MC68008 uses two-wire bus arbitration, whereas the MC68000 uses three-wire bus arbitration.
4) The MC68000 supports three interrupt lines and, therefore, seven interrupt levels. The MC68008 supports only two interrupt lines and, therefore, three interrupt levels. The MC68008 PLCC package supports three interrupt lines like the MC68000.
5) The MC68000 provides 16 Mbytes of addressable space. The MC68008 has 20 address lines creating 1 Mbyte of addressable space. The MC68008 PLCC package has 21 address lines for 2 Mbytes of addressable space.
6) The MC68008 does not have a VMA* signal as does the MC68000, but this signal can be generated with some simple external logic (see M68000 Users Manual (M68000UM/AD)).
7) The MC68000 and MC68008 are not pin compatible.

It should be noted that the MC68EC000 provides a lower cost solution compared to the MC68008 and is available at more frequencies than the MC68008. The MC68EC000 can operate in either 8-bit or 16-bit mode and does not include the 6800 peripheral interface, but is virtually identical to the MC68000 in every other way. If you are planning to do a design with the MC68008, it would be worth your time to consider the option of using the MC68EC000 instead.

QUESTION

During a byte access on the MC68000, which bits of the data bus are read/written?

ANSWER
If the MC68000 asserts UDS*, then bits 15-8 of the data bus will be used. If the MC68000 asserts LDS*, then bits 7-0 of the data bus will be used.

QUESTION
The MC68230 is an 8-bit peripheral. On which byte lane of the MC68000 should I place the MC68230?

ANSWER
The MC68230 should be placed on the lower byte lane (i.e., on data bits 7-0) of the MC68000. This is required because when the MC68230 provides an interrupt vector for the MC68000 it must be right-aligned. In all other respects, the two byte lanes of the MC68000 would be equally qualified for attaching the MC68230. Note that your software must be written so that all accesses to the MC68230 are byte accesses to even addresses to make sure the proper byte lane (bits 7-0) of the MC68000 are used.

QUESTION
What is the state of AS* during reset on the MC68000 and how long after reset before AS* becomes valid?

ANSWER
AS* remains inactive during reset. It will take an undetermined amount of time after reset before AS* will become valid. However, if you are not getting an AS* after you come out of reset, you should check the following signals:
1) Check to make sure BR* and BGACK* are negated to make sure you are not arbitrating the bus away from the MC68000.
2) Check to make sure HALT* is negated.

QUESTION
Intel has a pin that allows the signals to tri-state. Does the MC68000 have something similar?

ANSWER
Motorola does not provide such a pin. However, if you want to cause the signals from the MC68000 to tri-state, you can arbitrate the bus away from the MC68000 which will cause it to tri-state all signals with the exception of the bus arbitration signals.

QUESTION
In an MC68000 system, we desire that all registers of an 8-bit peripheral be mapped on byte boundaries. How can we do this?

ANSWER
Essentially, what is needed is pseudo-"dynamic bus sizing." This feature is implemented on the MC68020 and MC68030, but not on the MC68000. There is no glueless way of implementing dynamic bus sizing on the MC68000. The cheapest solution is to place the peripheral on one of the byte lanes and use the MOVEP instruction. The UDS/LDS signals will uniquely determine the byte of the data bus.

QUESTION
What is the interrupt latency for the MC68000 family?

ANSWER
Interrupt latency is the time from when an interrupt occurs to when the first instruction of the interrupt handler is started. The amount of time required for this will depend the processor, what the processor is doing when the interrupt occurs, and the speed with which the memory responds to reads and writes.

Worst case interrupt latency is the absolute maximum amount of time it will take for the processor to respond to the interrupt for a given set of conditions. A given interrupt may respond sooner (and usually will), but it will never take longer. The selection of the conditions can have a major impact on the interrupt latency. For example, the number of wait states on writes will effect the time it takes to stack the interrupt stack frame. An infinite number of wait states on writes would result in an infinite interrupt latency.

The following lists the interrupt latency for a given processor and the conditions assumed.

MPU	Conditions	Latency
MC68EC000 16-bit data bus clks	MOVEM.L xxx.L,D0-D7/A0-A7 DIVS xxx.L, Dn interrupt at beginning of MOVEM	378
MC68EC000 8-bit data bus clks	MOVEM.L xxx.L,D0-D7/A0-A7 DIVS xxx.L, Dn interrupt at beginning of MOVEM	574
MC68EC020 MC68EC030 clks 32-bit bus	MOVEM.L <memory indirect ea>,D0-D7/A0-A7 interrupt at the beginning of MOVEM no bus errors	197
MC68EC040 clks	any integer instruction stream, no back-to back 64-bit divide instructions no CAS,TAS,CAS2 instructions	78

QUESTION
How many transistors are on each of the MC68000 family devices?

ANSWER

DEVICE	TRANSISTORS
MC68000	68,000
MC68EC000	68,000
MC68HC000	68,000
MC68HC001	68,000
MC68008	70,000
MC68010	84,000
MC68020	190,000
MC68EC020	190,000
MC68030	273,000
MC68EC030	251,000

MC68040	1,170,000
MC68EC040	962,000
MC68LC040	813,500
MC68230	16,700
MC68302	320,000
MC68330	235,000
MC68340	350,000
MC68440	26,000
MC68450	40,000
MC68451	34,000
MC68661	6,000
MC68681	9,743
MC68851	210,000
MC68881	155,000
MC68882	176,000
MC68901	9,100

QUESTION
If BR* is asserted before RESET* negates, when will the MC68HC000 take control of the bus?

ANSWER
If BR* is asserted when the MC68HC000 comes out of reset, it will assert BG* in order to grant the bus to the requesting device. The requesting device will then most likely respond with BGACK*. The MC68HC000 will not take back the bus until BR* and BGACK* are both negated.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:06 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68020/030/040 FAQs
Entry Prepared On: Jul 22 17:11:15 1994

QUESTION
On the MC68020/MC68030/MC68000/MC68340, there are two specifications that describe the data bus hold time. One specification describes the data hol
d
time relative to the falling clock edge, and another specification describes the data hold time relative to the AS* (address strobe) signal. Why should both of these specifications be met? Does the data hold time relative to AS* also apply to the DS* (data strobe) signal?

ANSWER
Motorola usually provides two specifications to satisfy two different design methodologies. The first design methodology is to design the system "asynchronously". The main advantage of an asynchronous design is that there is no need to distribute a copy of the MPU clock throughout the entire PCB, resulting in a less noisy design. The main disadvantage is the lack of predictability of when the bus cycle ends. That is, since an asynchronous design generally has no idea where the MPU clock is, it is not possible to predict when a termination signal such as DTACK* takes effect.

Synchronous designs on the other hand, are becoming more common due to improvement of PCB technology (less noisy) and the existence of powerful clock drivers such as the 74F803, the 88915 and 88916. These clock drivers are capable of providing multiple copies of the CPU clock throughout the board while introducing less than 1 ns of skew between each copy of the clock. As the term "synchronous" implies, the clock is the heartbeat of the system.

When employing an asynchronous design, it is a common practice to use AS* to define when an address is valid and DS* to define when the data bus is valid. This is the reason that specifications are defined relative to AS* and DS*.

When employing a synchronous design, it is sufficient to know which clock edge a bus cycle begins and which clock edge/edges the termination signals and data will be valid, held, etc. This is the reason that specifications are defined relative to the clock.

Motorola's policy of drawing the bus cycle timing diagrams with both synchronous and asynchronous specifications in the same drawing admittedly causes some confusion. However, the solution is simple. If you are employing asynchronous design techniques, then the specifications relative to AS* or DS* are most important since it is most likely that your system would become constrained by these specifications. If you are employing synchronous design techniques, then the synchronous (i.e., relative to the clock) specifications should be met. In any case, only one specification needs to be met. The specification that needs to be met is defined by your design
methodology.

QUESTION
Please discuss the implications of spec #28 of the MC68000/MC68020/MC68030.

ANSWER
Spec #28 defines the maximum time that DTACK* (on the MC68000) or DSACKx*
(on the MC68020/MC68030) may be kept asserted following the negation of AS*. The reason for this specification is to guarantee that a subsequent bus cycle does not incorrectly terminate due to a late DSACKx* from a previous bus cycle.

QUESTION
On an MC68030 or MC68040, if a burst operation is not allowed to complete normally and only two long words are burst into the cache, what happens in the cache?

ANSWER
Given that the MC68030/MC68040 has one valid bit per long word, the first two long words that burst successfully are marked valid, and the final 2 long words that were aborted are marked invalid.

QUESTION
When may DSACKx* be negated on an MC68000/MC68020/MC68030/CPU32 bus cycle?

ANSWER

The DSACKx* signals may be negated in one of two ways. The first way is to ensure that DSACKx* is present for two consecutive falling edges of the clock and that specs #47a and #47b are met. The second way is to assert DSACKx* and then wait until AS* or DS* negates. Please be aware that spec #28 and spec #28a define a maximum time that DSACKx* may be asserted without affecting the next bus cycle.

QUESTION
On the MC68020/MC68030 bit-field instructions, is bit 0 the most significant or the least significant bit?

ANSWER
On a bit-field instruction, bit 0 represents the most significant bit. The Programmer's Reference Manual (M68000PM/AD) discusses the various data formats and provides examples of how each appears.

QUESTION
Why isn't my MC68020/MC68030 working?

ANSWER
The most common problems are clocks that are out of specification (i.e., duty cycle, rise/fall time, undershoot or overshoot), power supplies out of specification (i.e., noise on the ground pin, improperly decoupled power and ground, spikes on Vcc, etc.), ground bounce on input signals, and not accounting for signal skews (i.e., addresses buffered when AS* is not buffered, MC68EC030 clock to logic clock skew, wire delays, improper hold times, etc.).

QUESTION
On an MC68020/MC68030/MC68040/CPU32, can I ground AVEC?

ANSWER
If all interrupts are to be autovectored, then it is ok to ground AVEC*. Please note that on an MC68EC000, AVEC* must be asserted only during
interrupt acknowledge cycles. Also, note that for CPU32-based processors, if AVEC* is shared with another signal on the same pin, the pin must be configured as AVEC*.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:06 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68020/EC020 FAQ
Entry Prepared On: Jul 22 17:31:44 1994

This entry contains one or more associated non-viewable file. To download or have this file faxed to you, select illust from the menu.

QUESTION
When using the MC68EC020 with an 8-bit external data bus, can data be placed on the bus by means of decoding only A0 and A1 (ignoring SIZ0 and SIZ1)?

ANSWER
SIZ0 and SIZ1 are not needed for decoding byte enables for an 8-bit bus. SIZ0 and SIZ1 are only used for 16- and 32-bit bus sizes.

QUESTION
Are there any requirements for the HALT* signal upon reset of the MC68EC020?

ANSWER
There are no requirements of the HALT* signal during the reset operation.

QUESTION
Signal Parameter Relationships

ANSWER
The simple answer is that spec #13 and spec #25 do override the apparent overlap shown by specs #12, #8 and #53. The following discussion hopefully helps in clarifying the issue:

Synchronous And Asynchronous Signal Parameter Relationships:

Motorola microprocessors are synchronous machines. All parameters that describe a signal relationship to the CLK always take precedence over parameters which describe relationships between non-CLK signals. Synchronous parameters are to be used to understand how the bus works. Asynchronous parameters, which describe relationships between non-CLK signals, are given only for convenience. If these asynchronous parameters cause any confusion, they should be ignored.

Understanding the above statements will clear up most of the confusion whenever both synchronous and asynchronous parameters are used.

The following refers to the illustration attached to this file.

An example of a common problem is the relationship between parameters 6, 9, and 11. Parameter 6 describes the relationship of CLK to the address bus. Parameter 9 describes the relationship of CLK to AS*. Because parameters 6 and 9 are based on CLK, they are synchronous parameters. Parameter 11 describes two non-CLK signals, and is therefore an asynchronous parameter.

To justify the existence of parameter 11, let us pretend that parameter 11 is not provided and that this parameter must be derived from parameter 6 and 9. The equation is as follows:

$$(\text{Parameter11})_{min} = (1/2\ \text{CLK}) - (\text{Parameter6})_{max} + (\text{Parameter9})_{min}$$

Given this method, and using an MC68020 at 25 MHz as an example,

$$(\text{Parameter 11})_{min} = 20 - 25 + 3 = -2\ \text{ns}.$$

This means that it is possible for AS* to be actually slower than the address bus, even though it has a half-clock head start. In reality however, the address will always be ready prior to the assertion of AS*. The above equation is too conservative to be of use to system designers. If there is no setup time of the address bus relative to AS*, why use AS*?

At this point, the justification for parameter 11 is evident. The fallacy of the above equation is that it assumes that the environmental conditions (voltage and temperature) that cause parameter 6 to be maximum are also what causes parameter 9 to become minimum. To account for these common mode effects, Motorola guarantees parameter 11 (an asynchronous parameter) which is better than can be derived from the synchronous parameters alone.

QUESTION
For an MC68020 design, is there a way of interfacing synchronous SRAMs without using data latches? The data hold time creates some problems.

ANSWER
An MC68020 asynchronous design is difficult to do because of the short data hold time coupled with the long DS* negation time. Therefore, another level of logic for chip select generation is impossible without latching the data. However, by creating a separate synchronous DS*-like signal which negates within a short period of time after the clock edge (as illustrated in the attached document), this type of design is possible.

An MC68EC030 provides additional data hold time over an MC68020.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:06 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68030/EC030 FAQ
Entry Prepared On: Jul 22 17:26:40 1994

QUESTION
I would like to call procedures written in Motorola MC68020 assembly language from an ANSI C program. Do you have any information regarding how to develop mixed-language calls with the MC68020 processor?

ANSWER
How to mix assembly with "C" will depend on the compiler being used. Check the documentation that comes with your compiler or contact the compiler company directly.

QUESTION
How does the MC68020 provide cache coherency with other bus masters?

ANSWER
The MC68020 only has an instruction cache, so cache coherency is not an issue unless the other bus master is modifying the code. Dynamic modification of code is not recommended under any circumstances.

QUESTION
Can I have 8-bit memory for code and stack on an MC68020?

ANSWER
The feature we call "dynamic bus sizing" allows the MC68020 to run out of 8-bit EPROMs. The MC68030 and CPU32-based chips such as the MC68340 and MC68330 also support dynamic bus sizing. Dynamic bus sizing is also available for the MC68040 through an external device called the MC68150.

QUESTION
How does the MC68030 provide cache coherency with other bus masters?

ANSWER
The MC68030 has both an instruction and a data cache. Cache coherency on the instruction cache is not an issue unless the other bus master is modifying the code. Dynamic modification of code is not recommended under any circumstances.

Cache coherency on the data cache is not an issue because the data cache is a write-through cache. This means if the MC68030 writes to a location that is in the data cache, an external write will also occur. The data cache is write allocated, so an entry is placed in the data cache only when a read is performed and the CIIN* signal is negated on the response.

QUESTION
Is the MC68EC030 pin compatible with the MC68030?

ANSWER
From the pin-out perspective, the only difference between the MC68030 and the MC68EC030 is the lack of the MMUDIS* pin on the MC68EC030. The MC68EC030 designates that pin as a NC (no connect). Be aware that if the application requires the use of the PMMU, that the system will fail to work properly.

QUESTION
The MC68030 timing specifications are measured off which voltage level?

ANSWER
The answer depends on the frequency. Specs #2 #3 are measured off 1.5 V to 1.5 V for everything above and including the 33 MHz devices.

QUESTION
What DRAM controller does Motorola recommend for the MC68030?

ANSWER
There is a DRAM Controller manufactured by the V3 Corporation. V3 can be reached at (416) 285-9188.

QUESTION
If during a burst fill a word which is not available (from external cache) is requested, how do I signal the MC68030 not to cache this word and to abort the burst?

ANSWER
The following conditions can abort a burst fill:
    CIIN* asserted
    BERR* asserted
    CBACK* negated.
When a burst fill is aborted, the data is not cached for the entire line. Remember that the CBACK* signal corresponds to the next long word transfer, not the current one.

QUESTION
Can I use my MC68030 assembler for my MC68EC030?

ANSWER
An MC68030 assembler can be used for an MC68EC030. The only difference is the naming of the access control unit (ACU) registers. The access control unit status register (ACUSR) decodes as the MMUSR on an MC68030 assembler. Access control register 0 (ACR0) and access control register 1 (ACR1) decode as TT0 and TT1 on an MC68030 assembler. The MC68030 assembler will assemble PFLUSH, PFLUSHA, and PLOAD. In addition, PMOVE and PTEST will assemble to with opcodes that do not match MC68EC030 opcodes. These instructions will cause undefined results on the MC68EC030 and should be avoided. These undefined results are not guaranteed to be consistent between MC68EC030 mask revisions. These instructions should not be used on an MC68EC030.

QUESTION
Is the MC68EC030 pin compatible with the MC68030?

ANSWER
Yes, but note the no connect pin (E-12) of the MC68EC030. The MC68EC030 will plug into an MC68030 socket. The MC68030 will not plug in to an MC68EC030 socket because of the additional guide pins of the MC68030.

QUESTION
How do I drive the MC68030 40 MHz clock?

ANSWER
There are several clock drivers that work with the MC68030. The MC88916 makes an excellent clock driver because of its low price, conformance to MC68030 40 MHz specifications, and the addition of the RESET* signal that drives the MC68030 RESET* during power-up.

QUESTION
What sort of performance does the MC68030 deliver?

ANSWER
At 25 MHz, the MC68030 delivers up to 5.8 MIPs. At 40 MHz, the MC68030 delivers up to 9.2 MIPs.

QUESTION
Are there any software differences between the MC68030 and the MC68EC030?

ANSWER
Yes. For an MC68EC030 whose date code is 9104 or later, the following procedure allows a software detection of an MC68EC030. The method depends on making an access thorough the memory management unit (MMU). If the part is an MC68EC030, there will be no difference between an access with the MMU enabled and the MMU disabled. If the part is an MC68030, the physical location will match the logical location.

First initialize the MMU to map one page of memory onto another page as if the part was an MC68030. Examples of this can be found in the MC68030 User's Manual (MC68030UM/AD). If the MMU instructions cause an illegal instruction, then the part is an MC68EC030. If the MMU instructions do not cause an illegal instruction, then write a flag to the mapped page. Disable the MMU and read the location with the flag. If the flag's physical address matched the logical address, the part is an MC68EC030. If the flag's physical address does not match the logical address, then the part is an MC68030.

QUESTION
Should I decouple the power and ground plane of my design?

ANSWER
The higher the frequency, the greater the importance to properly decouple the power and ground planes. Decoupling is performed by placing a capacitor with one node of the capacitor connected to the Vcc and the other to ground. Failure to decouple can result in excessive noise and thus an unreliable design. Unfortunately, decoupling is not an exact science. No hard and fast rules or capacitor values apply for every situations.

There are two types of decoupling: bulk and filter. A bulk decoupling capacitor stores charge for the entire board and provides instantaneous current during the delay before the power supply can respond. The bulk decoupling capacitors are typically 100 uF capacitors and placed around the board. Bulk capacitance is not needed on all boards. Filter decoupling is meant to filter out noise between the power and ground plane, and to provide instantaneous current to parts. When a high-speed part is switching its outputs, it will get the necessary current. Without decoupling, this current may be supplied by other parts near the switching parts. This can cause the neighboring parts to fail only when certain patterns occur on the board (e.g., the addresses switching from $FFFF FFF0 to $0000 0000). The typical value for the decoupling capacitor is 0.1 uF or 0.01 uF. A good rule of thumb is to provide one decoupling capacitor for every power and ground pair of a part.

For even greater noise immunity, multiple power and ground planes can be provided that separate the more sensitive logic from the logic that will provide the greater noise due to its switching. An example of this is the MC68030, where the Vcc and GND are split between address bus, data bus, and three groupings of control signals and internal logic. This would allow the user to connect the address and data bus to one power and ground plane (since they would be switching more) and the internal logic and control signals on another power and ground plane, further reducing the noise seen by the part. This is not usually required.

QUESTION
My MC68EC030 40 MHz clock does not meet clock rise and fall time specs. What should I use?

ANSWER
Recommended crystal manufacturers which meet high frequency specifications include Kyocera and Champion.

QUESTION
Can I connect to the no connect pin (E-12) of my MC68EC030?

ANSWER
Yes, but only connect to ground. The pin should be left floating or grounded.

QUESTION
Can I gate off the MC68EC030 clock?

ANSWER
NO. The MC68EC030 may be permanently damaged by removing the clock while

powered up. If power is removed from the MC68EC030, the clock must also be removed. Otherwise, the Vin while clock is high will exceed Vcc beyond specifications.

QUESTION
How do cold and warm resets differ?

ANSWER
The cold, or power-on, reset takes the MC68EC030 from the unknown state to a known state. During power-up, the MC68EC030 will at some time begin executing the instructions it finds in its pipeline. These instructions are the random state of the internals of the MC68EC030 as the Vcc ramps from 0 to 5 volts. If the random instruction was a software reset, then the MC68EC030 will assert RESET for up to 512 clocks. Since the MC68EC030 can not distinguish between the user's externally applied reset and the internally asserted reset, the 520+ clocks after Vcc and the clock being within tolerance are needed to ensure the MC68EC030 recognizes the reset. The warm reset requires only 10+ clocks because it is defined as 10 clocks from RESET* asserted to RESET* negated. If the MC68EC030 was executing a software reset, then the warm reset would have to wait until RESET* was negated or be applied for 520+ clocks.

QUESTION
For the MC68EC030, what should I initialize on start up?

ANSWER
The program counter and the interrupt stack pointer must be initialized. Then the other stack pointers and status register must be initialized. The vector base register should be set to the start of the vector table, the ACU (if used) should be initialized and enabled, and finally, the caches (if used) should be invalidated and enabled.

The interrupt stack pointer is set on reset by the MC68EC030 fetching the long word interrupt stack pointer value from memory location $0. The program counter is set on reset by the MC68EC030 fetching the value of the starting address from memory location $4. The remainder of the initialization is handled by the code which starts at the address pointed to by the data in address $4.

The following is an example of an MC68EC030 initialization. Your initialization may vary. MSP_START, USP_START are the start addresses for the respective stack pointers. VBR_START is the base address offset for the vector table. ACR0_INIT, ACR1_INIT, CACR_INIT, SR_INIT are the initialization values for the respective registers. START is the label for the start of the program after initialization is complete.

```
INIT        MOVE.L     #MSP_START,%D0  ;get master stack address
            MOVEC      %D0, %MSP       ;initialize master stack
            MOVE.L     #USP_START,%D0  ;get user stack address
            MOVEC      %D0, %USP       ;initialize user stack
            MOVE.L     #VBR_START,%D0  ;get vector base address
            MOVEC      %D0, %VBR       ;initialize vector base
            PMOVE.L    ACR0_INIT,%AC0  ;initialize ACR0
            PMOVE.L    ACR1_INIT,%AC1  ;initialize ACR1
            MOVE.L     #CACR_INIT,%D0  ;get cache initialization
            MOVEC      %D0, %CACR      ;initialize cache control
            MOVE.W     #SR_INIT, %SR   ;initialize status register
```

```
           JMP        START           ;begin program

                    +++++  End of AESOP Entry +++++

                                    AESOP
                Today Is Thu Sep  1 04:29:07 1994

   Topic Name: M68000 FAQs (by Device)
   Subject:MC680X0 FAQ
   Entry Prepared On: Jul 22 17:21:46 1994
------------------------------------------------------------------------

Applies only to the MC680X0 parts such as MC68000, MC68010, MC68020
MC68030, MC68040, etc.
------------------------------------------------------------------------
QUESTION
How can I extend my physical address space beyond the 4 Gigabytes (for
MC680x0 processors with 32 external address bits) or 16 Megabytes (for
MC680x0 processors with 23 external address bits)?

ANSWER
The way to effectively increase your physical addressing space depends on
the processor. All MC680x0 processors allow external hardware to segment
the memory. Some of the MC680x0 processors allow physical address expansion
based on function codes. In addition, the paged memory management unit
(PMMU) on the MC68030, MC68LC040, MC68040 allows the logical expansion of
memory so that the same external physical space becomes effectively larger
as pages are switched in and out as needed. The PMMU can also be used to
make a smaller physical memory appear to be a larger logical memory.

The function codes can be used to divide the memory map between supervisor
and user code, supervisor and user data, CPU accesses, or any combination
of the preceding. An example of this is to have separate memory for
supervisor and user space (both code and data). An access to a memory
location (e.g., $0000 1000) would require the additional decoding of the
function codes to determine if this is the user or supervisor bank of DRAM.
The privilege level (user or supervisor) is set in the status register. All
instructions are available in supervisor level. Some instructions are
privileged and will cause a privilege exception if executed with the status
register set to user mode.

Hardware segmentation requires the additional address bits to be latched on
a separate bus cycle. All subsequent accesses are to the segment of memory
indicated by the latched addresses. If another segment of memory is
required, then a new set of extended address bits must be latched. This
requires the program to know which segment it is on and when it passes
through a segment barrier.

QUESTION
How do I do floating-point instruction emulation in software?

ANSWER
The 68K Source document (BR729/D) provides information on third party
vendors and their tools. Included is information on floating-point
emulation packages for various MC680x0 processors. The software to emulate
floating-point instructions works either by taking an exception for each
floating-point instruction (which allows code to be written independently
```

of the system being used, but will result in a large penalty for non-floating-point equipped systems), by substituting a subroutine for each floating-point instruction (which is also portable, but will not gain performance in systems equipped with floating-point units), or to set a switch when compiling the code to use one or the other method.

QUESTION
What are the op codes for unimplemented instructions for the MC680x0 family?

ANSWER
All the unimplemented op codes for the MC680x0 family have either an 'A' (1010) or an 'F' (1111) for the first nibble of the op code. The 'F' line opcodes are used by Motorola to expand the instruction set (e.g., adding floating-point instructions to the 68020). The 'A' line instructions are generally used by customers to provide system call routines that can be modified by patching the exception vector table (as opposed to a complete ROM change).

QUESTION
Where can I get an MC680x0 compiler for my IBM PC?

ANSWER
The 68K Source document (BR729/D) provides information on third party vendors and their tools. Included is information on cross compilers for various MC680x0 processors and platforms. In addition, Motorola OEMs Intermetrics C compiler which includes an assembler. This software package runs on several different platforms.

QUESTION
Where can I get a socket for my MC680x0 processor?

ANSWER
Motorola does not sell microprocessor sockets nor do we have specific recommendations for microprocessor socket manufacturers. Because of the wide variety of sockets available from third party vendors, the best way to select a socket vendor is to contact your local distributor which will usually represent several vendors.

QUESTION
Where can I get a software development package for the MC680x0 family?

ANSWER
The 68K Source document (BR729/D) provides information on third party vendors and their tools. Included is information on software development packages for various MC680x0 processors and platforms. In addition, Motorola OEMs Intermetrics software development package.

QUESTION
Where can I get an emulator for my MC680x0 processor?

ANSWER
A partial list of the emulators available for various MC680x0 processors is attached. The list is not exhaustive because of the wide range of companies providing emulator support and the constant addition to the processors covered by those companies. The attached list includes phone numbers where you can obtain more information on the availability, pricing and features

of these emulators.

M68000 Family Emulator Support List

Company Name: Applied Micro
Phone: (800) 426-3925
Processor: MC68008
Max Speed: 10 MHz
Memory: 128K-2M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, SCSI, 2K Trace Buffer, Compiler Support

Company Name: Applied Micro
Phone: (800) 426-3925
Processor: MC68000
Max Speed: 16.67 MHz
Memory: 128K-2M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, SCSI, 2K Trace Buffer, Compiler Support

Company Name: Applied Micro
Phone: (800) 426-3925
Processor: MC68010
Max Speed: 12.5 MHz
Memory: 128K-2M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, SCSI, 2K Trace Buffer, Compiler Support

Company Name: Applied Micro
Phone: (800) 426-3925
Processor: MC68020
Max Speed: 25 MHz
Memory: 128K-2M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, SCSI, 2K Trace Buffer, Compiler Support

Company Name: Applied Micro
Phone: (800) 426-3925
Processor: MC68EC030
Max Speed: 25 MHz
Memory: 128K-2M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, SCSI, 2K Trace Buffer, Compiler Support

Company Name: Applied Micro
Phone: (800) 426-3925
Processor: MC68030
Max Speed: 33 MHz
Memory: 128K-2M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, SCSI, 2K Trace Buffer, Compiler Support

Company Name: Applied Micro
Phone: (800) 426-3925
Processor: MC68302
Max Speed: 16.67 MHz
Memory: 128K-2M

Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, SCSI, 2K Trace Buffer, Compiler Support

Company Name: Cadre Technologies
Phone: (714) 474-1170
Processor: MC68000
Max Speed: 16.67MHz
Memory: 512K-1M
Host Computer(s): PC, VAX, SUN
Features: RS232, Ethernet, 2K Trace Buffer, Compiler Support

Company Name: Cadre Technologies
Phone: (714) 474-1170
Processor: MC68020
Max Speed: *33 MHz
Memory: 512K-1M
Host Computer(s): PC, VAX, SUN
Features: RS232, Ethernet, 2K Trace Buffer, Compiler Support

Company Name: Cadre Technologies
Phone: (714) 474-1170
Processor: MC68030
Max Speed: *33 MHz
Memory: 512K-1M
Host Computer(s): PC, VAX, SUN
Features: RS232, Ethernet, 2K Trace Buffer, Compiler Support

Company Name: Hewlett-Packard
Phone: 800-752-0900
Processor: MC68008
Max Speed: 10MHz
Memory: 32K-128K
Host Computer(s): HP-9000
Features: RS232, RS-422, 1K Trace Buffer, Compiler Support

Company Name: Hewlett-Packard
Phone: 800-752-0900
Processor: MC68000
Max Speed: 16.67MHz
Memory: 32K-512K
Host Computer(s): PC, Apollo, HP-9000
Features: RS232, RS-422, 1K Trace Buffer, Compiler Support

Company Name: Hewlett-Packard
Phone: 800-752-0900
Processor: MC68010
Max Speed: 12.5MHz
Memory: 32K-512K
Host Computer(s): PC, Apollo, HP-9000
Features: RS232, RS-422, 1K Trace Buffer, Compiler Support

Company Name: Hewlett-Packard
Phone: 800-752-0900
Processor: MC68020
Max Speed: *25MHz
Memory: 256K-4M
Host Computer(s): PC, Apollo, HP-9000
Features: RS232, RS-422, 1K Trace Buffer, Compiler Support

Company Name: Hewlett-Packard
Phone: 800-752-0900
Processor: MC68030
Max Speed: *33MHz
Memory: 256K-4M
Host Computer(s): PC, Apollo, HP-9000
Features: RS232, RS-422, 1K Trace Buffer, Compiler Support

Company Name: Hewlett-Packard
Phone: 800-752-0900
Processor: MC68040
Max Speed: 25 MHz
Memory: 256K-4M
Host Computer(s): PC, Apollo, HP-9000
Features: RS232, RS-422, 1K Trace Buffer, Compiler Support

Company Name: Hewlett-Packard
Phone: 800-752-0900
Processor: MC68302
Max Speed: 16.67 MHz
Memory: 256K-4M
Host Computer(s): PC, Apollo, HP-9000
Features: RS232, RS-422, 1K Trace Buffer, Compiler Support

Company Name: Hewlett-Packard
Phone: 800-752-0900
Processor: MC68340
Max Speed: 16.78 MHz
Memory: 256K-4M
Host Computer(s): PC, Apollo, HP-9000
Features: RS232, RS-422, 1K Trace Buffer, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68008
Max Speed: 10 MHz
Memory: 256K-1M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68000

Max Speed: 16.67MHz
Memory: 256K-1M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68001
Max Speed: 16.67 MHz
Memory: 256K-1M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68010
Max Speed: 12.5 MHz
Memory: 256K-1M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68EC020
Max Speed: 25MHz
Memory: 256K-2M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68020
Max Speed: 33MHz
Memory: 256K-2M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68EC030
Max Speed: 33MHz
Memory: 256K-4M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68030
Max Speed: 33MHz
Memory: 256K-4M
Host Computer(s): PC, Apollo, SUN

Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68302
Max Speed: 16.67MHz
Memory: 256K-1M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68330
Max Speed: 16.78MHz
Memory: 256K-1M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68340
Max Speed: 16.78MHz
Memory: 256K-1M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Huntsville Micro
Phone: (205)881-6005
Processor: MC68040
Max Speed: 25MHz
Memory: 256K-1M
Host Computer(s): PC, Apollo, SUN
Features: RS232, Parallel Interface, 8K Trace, Compiler Support

Company Name: Microtek
Phone: (213)321-2121
Processor: MC68008
Max Speed: 10MHz
Memory: 256K-1M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, Parallel Interface, 2K Trace, Compiler Support

Company Name: Microtek
Phone: (213)321-2121
Processor: MC68000
Max Speed: 16.67MHz
Memory: 256K-1M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, Parallel Interface, 2K Trace, Compiler Support

Company Name: Microtek
Phone: (213)321-2121
Processor: MC68010
Max Speed: 12.5MHz
Memory: 256K-1M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, Parallel Interface, 2K Trace, Compiler Support

Company Name: Microtek
Phone: (213)321-2121
Processor: MC68020
Max Speed: *25MHz
Memory: 256K-1M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, Parallel Interface, 2K Trace, Compiler Support

Company Name: Microtek
Phone: (213)321-2121
Processor: MC68030
Max Speed: *25MHz
Memory: 256K-1M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, Parallel Interface, 2K Trace, Compiler Support

Company Name: Microtek
Phone: (213)321-2121
Processor: MC68302
Max Speed: 16.67MHz
Memory: 256K-1M
Host Computer(s): PC, VAX, Apollo, SUN
Features: RS232, Parallel Interface, 2K Trace, Compiler Support

Company Name: Zax Corp.
Phone: 800-421-0982
Processor: MC68008
Max Speed: 10 MHz
Memory: 128K-512K
Host Computer(s): PC, SUN
Features: RS232, Ethernet, Compiler Support

Company Name: Zax Corp.
Phone: 800-421-0982
Processor: MC68000
Max Speed: 16.67 MHz
Memory: 128K-512K
Host Computer(s): PC, SUN
Features: RS232, Ethernet, Compiler Support

Company Name: Zax Corp.
Phone: 800-421-0982
Processor: MC68010
Max Speed: 12.5MHz
Memory: 128K-512K

Host Computer(s): PC, SUN
Features: RS232, Ethernet, Compiler Support

Company Name: Zax Corp.
Phone: 800-421-0982
Processor: MC68020
Max Speed: 20MHz
Memory: 256K-1M
Host Computer(s): PC, SUN
Features: RS232, Ethernet, Compiler Support

Company Name: Zax Corp.
Phone: 800-421-0982
Processor: MC68030
Max Speed: 20MHz
Memory: 256K-1M
Host Computer(s): PC, SUN
Features: RS232, Ethernet, Compiler Support

*Requires wait-state(s)

QUESTION
How long should interrupts to the MC680x0 processor be asserted?

ANSWER
The MC680x0 processors can recognize an interrupt in as few as two clock edges. To ensure that a spurious interrupt is not taken, the interrupt signals should remain asserted until the corresponding interrupt acknowledge cycle is performed.

QUESTION
On an MC680x0 or CPU32-based processor, if an exception such as a bus error or illegal instruction occurs, is there a guarantee that the first instruction of the handler will be executed prior to servicing a pending interrupt?

ANSWER
The architecture does not guarantee that the first instruction executes.

QUESTION
I am trying to derive the rise time of outputs generated by the MC680x0 and CPU32-based processors. When using the Iol values, I calculate this to be 780 ns. What's wrong?

ANSWER
The problem is that the Iol reflects the value of current at Vol. This in no way specifies what the current level is during the actual transition. The current during transition is possibly an order of magnitude higher than what Motorola specs as the Iol.

QUESTION
It is desired that the final write bus cycle on an MC680x0/CPU32 be properly written if a warm reset is asserted during that final write cycle. Please advise.

ANSWER
One solution is to assert HALT* to the processor and then to assert RESET*

only when HALT* is asserted and AS* is negated. This, however requires additional external hardware.

QUESTION
Will the MC680x0 and CPU32-based processors retry vector fetches out of reset?

ANSWER
Yes, the processor retries the vector fetch until it succeeds. Only a bus error or address error causes a catastrophic error (i.e., the processor asserts HALT* and freezes).

QUESTION
Does Motorola provide derating specs for the MC680X0 family of processors and associated peripherals?

ANSWER
Motorola does not provide derating specs for the MC680X0 processors or their associated peripherals. These parts are tested with a 130 pF load because this is where the specs for the parts are set. You should not overload these parts at more than 130 pF because this will increase the power dissipation in the part and, therefore, adversely affect its reliability. Also, if you underload these parts, you must still meet the appropriate specs in order to guarantee the proper operation of the part.

QUESTION
Can you recommend reset circuitry for the MC680X0 family?

ANSWER
The reset circuitry required will depend on the processor. The NMOS processors require a fixed amount of time whereas the HCMOS processors require a fixed number of clocks. Please see the electrical specs to determine the requirements for the processor you are using.

There are two ways you can set up your circuit:
1) Use an RC circuit to meet the timing requirements.
2) Count clocks to meet the timing requirements.

Either type of circuit could be used for either NMOS or HCMOS processor.

QUESTION
I would like to know the output buffer characteristics for the MC680x0 family processors. Specifically, I would like a typical AC current draw produced by the output buffers.

ANSWER
It sounds as though you are trying to get derating specs for the MC680x0 processors. Motorola does not provide derating specs for the MC680x0 processors or their associated peripherals. These parts are tested with a 130 pF load because this is where the specs for the parts are set. You should not overload these parts at more than 130 pF because this will increase the power dissipation in the part and, therefore, adversely affect its reliability. Also, if you underload these parts, you must still meet the appropriate specs in order to guarantee the proper operation of the part.

QUESTION

If all IPLx lines of an MC680x0/CPU32-based processor are asserted during reset, will the processor take an interrupt after coming out of reset?

ANSWER
It is recommended that the IPLx lines NOT be asserted during reset. Although no problem has been observed by Motorola and Motorola has not been able to duplicate this operation, there have been reports by customers that this causes erroneous behavior.

QUESTION
Do the M68000 family processors have any internal bits that indicate how many external floating-point units are active?

ANSWER
There are no internal bits for this purpose. The approach taken in finding out how many floating-point units (or coprocessors) are active is with the coprocessor protocol violation exception. If a coprocessor is not present and an access is attempted, a bus error is generated via a bus-error time out (watchdog timer) and a coprocessor protocol violation occurs. If the coprocessor protocol violation occurs, then the floating-point unit (or coprocessor) is not present.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:08 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68150 FAQ
Entry Prepared On: Jul 14 14:39:46 1994

QUESTION
Is a bus functional model of the MC68150 available?

ANSWER
The availability of the 68150, and questions regarding its use should be directed to Motorola's Logic Integrated Circuit Division which created the part. Their applications department phone number is 602-962-3397.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:08 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68230 FAQ
Entry Prepared On: Jun 20 13:11:30 1994

Q: What is the 68230 ESD rating?

A: MC68230 Mask C10R Rev.0/9 1.5KV Human Body Model

Q: When I programmed the PI/T timer to generate interrupts on the PC3/TOUT line upon timer timeout, the PC3/TOUT line would glitch, even though the timer itself had not been enabled. This glitch occurred during the write

cycle to the timer control register (TCR) that set up the PI/T to assert the PC3/TOUT line when the counter counted through zero.

A: If bit 3 of the Port C data direction register (PCDDR)is zero, indicating that the PC3/TOUT pin is defined as an input, no glitch occurs on TOUT. If bit three of the PCDDR is one, indicating that the PC3/TOUT pin is an output, then the glitch appears on PC3/TOUT when a write occurs to the TCR.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:08 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68300 FAQ
Entry Prepared On: Jul 22 17:32:29 1994

QUESTION
Which Motorola processors use the CPU32 as the core?

ANSWER
The M68300 integrated processor family parts incorporate the CPU32 as the core processor. This family includes the MC68306, MC68330, MC68331, MC68332, MC68333, and MC68340.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:08 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68306 FAQ
Entry Prepared On: Jun 6 09:15:18 1994

ClockBus ErrorCLKOUTBERR

Q: CLKOUT apparently is tri-stated. How can this happen?
A: Two possible reasons
a) If BERR is asserted during reset, all outputs will tristate. If BERR* is still asserted when RESET* negates, it will be latched and all outputs will remain tristated until the next reset, even if BERR negates later. This feature is for emulator support and does not appear to be documented very well. Note: If the BERR* logic is relying on the CLKOUT to turn the BERR* off, the CLKOUT tristating may prevent this.
b) JTAG has tristated CLKOUT - be sure TRST* is grounded.

Q: What should the XTAL and EXTAL waveforms look like?
A: In checking the SIerra 68306 board I made some measurements which may be of interest. The Sierra board uses a 16MHz crystal without Cext capacitor, similar to Richo's circuit. With a 1GHz sampling oscilloscope and a 1Mohm,6-9pf probe, I saw XTAL was an approximate squarewave, 0-5V with

some undershoot and overshoot. EXTAL is a rough triangle wave with the same amplitude.
Specifications

Q: Output high / low voltage is given at Ioh / Iol = rated maximum. What is rated maximum?
A: The rated maximum is the guaranteed current of that signal whenever the voltage is above (output high voltage) or below (output low votage) the specified voltage. The rated maximum for all outputs is 4 ma except for A(15:1), CLKOUT, UW*, LW*, OE*, RAS(1:0)*, CAS(1:0)*, DRAMA(14:0) and DRAMW* which are 8 ma.

Q: When will the 68306 assert BERR*
A: he BERR* signals will be asserted by the 68306 only if the bus tiemout is enabled (BTEN set in the system register), and the bus timeout period is met (programmed in bus timeout period register). Note: If the 68306 is asserting BERR* and the RESET* is asserted to the 68306, this will lock the processor into tristating the CLKOUT of the 68306 (only a power down will unlock the 68306).

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:09 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68307 FAQ
Entry Prepared On: Jun 6 09:17:31 1994

Q: Will we spec' the 68307 power dissipation when in the STOP mode (separately from the normal running mode)?

A: The power consumption values quoted in the UM are to follow the same form as the 68340. That is:
16.67MHz, 5V - whole chip running and whole chip stopped.
3.3V - whole chip running and whole chip stopped.
see 68307 Power Management FAX image

Q: Will we spec' the 68307 with a minimum frequency of 0 MHz? If so, does this mean that we can stop the clock to the 68307 for additional power savings?
A: Yes, the 307 will be spec 'ed from 0MHz to 16.67MHz. Yes, this means that additional power savings are available if the clock is stopped.
see 68307 Power Management FAX image

Q: What are the power dissipation numbers (preliminary is ok) available for the 68307 in the normal, STOP, and CLK disabled modes of operations?
A: see 68307 Power Management FAX image

Q: emulator solutions for the 307?
A: A company called Lauterbach in Germany has an Emulator available for the 307 however it is a high quality, high cost $28K solution. There U.S. distributer is BSO Tasking in Boston Contact Vaughn Orchard at 617-320-9400. Orion Instruments in San Jose will have an emulator in the

June/July time frame. Theirs will be lower cost $12 to $16K. Orion's contact is Jan Liband 415-327-8800. HP is working on an adapter that will allow their EC000 emulator to work with the 307 however this will not provide "Full-Blown" emulation instead it will be emulation of the EC000 core with the Peripherals disabled. We do not have a time frame for release of this product and knowing HP it will be end of year 94.

As far as adapters is concerned you may try ITT/Pomona, Their Tech Support # is 909-469-2912. They supply the clip on adapters for Orion Instruments emulators so they may have one for the 307.
Emulation technologies probably could help you as well as Interconnect Systems. I do not have contact names for any of those guys however.
If I can help any further please let me know.

Q: The question is on page 13 of Specification 1.1, Table 2-2. On the last line of the table, we have the following case:
UDS*=Low, LDS*=low, R/W*=write
The data pins D15-D8 writes valid data D15-D8, but the pins D7-D0 says "do not use". On an HC000, the way it was documented on page 3-5 of the M68000 User's Manual Rev 9 is that "these conditions are a result of current implementation and may not appear on future devices".

A: The "do not use" comment in table 2-2 is correct. As you know, the 68307 has a software selectable bus width for each of its chip selects. To do this, a dynamic bus sizing extension block has been added to the 68k bus interface. This introduces two cases: 16-bit and 8-bit mode interfaces for each chip select.

16-bit interface:
As per the 68000, the UDS*=Low, LDS*=High, and R/W=Low covers a byte write to an even location. The valid data byte on D8-D15 is reflected onto D0-D7 in the same way as the 68000 and HC000.

8-bit interface:
Given that the 8-bit data bus is D8-D15, a word write appears as two succesive byte writes with UDS*=Low, LDS*=High and R/W=Low. During the first byte write however, all 16-bit of the data word appear on the bus, although only the upper byte is validated by UDS.

In principal, the "Do not Use" statement applies to the 8-bit bus operation, with the same table covering both 8 and 16-bit. It may be that the table description could be embelished as such.

Q: The question pertains to read-modify-write cycles on the 307. Do the chip selects stay asserted during the read-modify-write cycles just like AS*?
I think that the answer is that the chip selects follow the behavior of AS*, and therefore, they stay asserted during read-mod-write transfers. I need confirmation on this issue.

A: Again this question relates back to the added dynamic bus sizing extension. When the 307 is running in 16-bit mode, the AS and associated CS line stay asserted throughout the Read-Modify-Write. In 8-bit mode (Core still 16-bit execution but 8-bit interface to it) the AS and CS lines negate between the Read and write of the RMC. So it seems that TAS can only be used with 16-bit memory. Our designers are investigating this? Any comments?

Q: How can one distinguish between instruction and data spaces without function codes?
As you know, the pin count of the 68307 was kept as low as possible to keep the chip cost down. One sacrifice was the function code pins. An emulator/analyser like the Orion 8800 will still be able to determine address spaces by monitoring the software (address and data bus) but there will be some restrictions.

A: There are several cases to consider:
1) Access to fixed internal registers,
2) Access to relocatable internal registers,
3) External Access within chip select range.
4)Interrupt acknowledges
5)External access outwith chip select range

From the top.
1) The MBAR and SCR are located at a fixed location in all address spaces. ($F0 to $FF reserved in all address spaces).This is a restriction for both user and analyser.

2) The MBAR is used to locate the base of the internal module regs, and assign them to a particular address space.If the compare function code bit (CFC) is SET, the register accesses are restricted to the assigned space. If CLEAR,registers are assigned to all address spaces.

If function codes are compared, the analyser can determine the address space by decoding the address with AS* asserted, and realising this is within the programmed (MBAR) module register address range.

If function codes are not compared, the user and analyser must assume that the register is accessed in all address spaces.

3) Each chip select address range can be assigned function codes. The analyser can see the CS asserted externally, and hence tie it to the assigned address space.

4) The interrupt aknowledge address space cannot be decoded. Must use A23:A4 = all 1's and AS asserted as decode, OR use a fixed decode area for the interrupt vector fetches (Deduce from PIVR). Make one of these a restriction

5) External accesses to addresses outwith chip select ranges have no function codes assigned, so all address spaces must be assumed.

It is not anticipated that customers will use external CS decode other than the 68307 chip selects, but for the sake of completeness, it must be considered. Such an external access can be recognised as: AS asserted, No CS asserted, and not an IACK or internal register location

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:09 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68340 FAQ

Entry Prepared On: Jul 22 17:34:03 1994

Question: What are the crystals for the MC68340?

Answer: The crystal should be a parallel resonant crystal, when used with the circuitry as laid out in the MC68340 User's Manual. There are some crystals that don't work for MC683XX parts. Specs for three watch crystal manufacturers that do work are:

(1) Lap-Tech 32.768 kHz:
- Freq. 32.768 kHxz
- Freq. Tolerance 120ppm
- Load Cap. 12.5 pF
- Drive Level 1.0 micro Watt max.
- Equivalent Rest. 30 k ohms max
- Q Factor 50,000 min.
- Insulation Rest. 500 M ohms at 100 Vdc
- Turnover Temp 25!C 15% !C

(2) Statek 32.768 kHz:
- Freq. 32.768 kHz
- Load Cap. 9 pF
- Drive Level 1.0 micro Watt max.
- Equivalent Rest. 30 k ohms max.
- Q Factor 60,000 min.(approx)

(3) Fox Electronics 32.768 kHz crystal is what is used on Motorola MC68340 boards.
- Freq. 32.768 kHz
- Load Cap. 12.5 pF (STD)
- Drive Level 1 micro watt max.
- Equivalent Rest. 35 k ohms max.
- Q Factor 50,000 min.
- Tolerance +/-20PPM
- Turnover Temp +25 C (+/-5 C)

The following crystal was used on Rev. H silicon and did not work. As soon as a better crystal was provided, the MC68340 started up.

ECLIPTEK:
- Freq. 32.768 kHxz
- Freq. Tolerance 120ppm at 25!C
- Load Cap. 12.5 pF
- Drive Level 10 micro Watt max.
- Equivalent Rest. 35 k ohms max.
- Q Factor 50,000 min.

NOTE: The quality of the crystal is very important. A more common name brand crystal will assure good results.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:09 1994

Topic Name: M68000 FAQs (by Device)

Subject:MC68450 FAQ
Entry Prepared On: Jul 22 17:13:07 1994

QUESTION
On the MC68450, is it possible to perform a bus cycle in four clocks?

ANSWER
No, the best that the MC68450 can do is a five-clock bus cycle. The documentation is in error.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:09 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68681/2681 FAQ
Entry Prepared On: Jul 22 17:14:30 1994

QUESTION
On the MC2681 or MC68681, assuming that the write cycle is chip-select controlled, what is the minimum data hold time after chip select is negated on a write cycle?

ANSWER
The answer is derived from a combination of two specifications. The first specification defines that there must be a setup time of the W signal relative to CS (Tch). The second specification defines the data hold time relative to W going high (Tdh). Since Tch = 0, and Tdh = 10 ns, adding them yields 10 ns for a data hold time relative to the CS.

QUESTION
The specification for the MC2681/MC68681 specifies a crystal series resistance less than 180 ohms. Does that still hold? Can a crystal of series resistance of 200 ohms suffice?

ANSWER
The specification that defines that a crystal must have a series resistance of 180 ohms still applies. It is recommended that any crystal with a larger series resistance not be used. Below is a list of manufacturers of crystal manufacturers that meet our specifications

Saronix, Palo Alto, CA (800) 227-8974
inside Ca. (800) 422-3355, part #NMP037

Bomar Crystal, Middlesex, NJ (201) 356-7787

Crystek Corporation, Fort Myers, FL (813) 936-2109

Electro-Dynamics Corp., Shawnee Mission, KS
(913) 262-2500

Allied Electronics, Fort Worth, TX (214) 265-9341
part # 994-0345

US Crystals, Fort Worth, TX (800) 433-7140

QUESTION
What value of shunt resistors must be used with the MC2681/MC68681?

ANSWER
The MC2681/MC68681 manual is incorrect. A shunt resistor must be added across the X1/CLK and X2 pins. This shunt resistor is needed to help provide a 50% duty cycle. The recommended value of a shunt resistor is 10-20 Mohm.

QUESTION
Which MC68681 signals require a pull-up resistor?

ANSWER
The following signals of the MC68681 require pull-up resistors: DTACK, IRQ. The OP7, OP6, OP5, OP4, OP3 signals may require pull-up resistors if they are used as open-drain, active-low outputs.

QUESTION
What happens when I read a reserved register on the MC68681/MC2681?

ANSWER
When a read of a reserved register (locations $02 or $0A) is attempted, the DUART is forced into a diagnostic mode. This mode is used to test the baud rate generator circuitry. When the DUART enters this mode, it outputs baud frequencies on the general purpose output lines which are multiples of the frequencies listed in the baud rate table. If the transmitter is enabled, the DUART may also transmit at these frequencies, regardless of the value selected in the clock select register (CSR). A read to the reserved registers may occur accidentally by a monitor program that performs a read/verify cycle after write cycles, by negating the write strobe prior to the chip select signal at the end of a write bus cycle, or possibly by asserting the write strobe after asserting CS* at the beginning of a write bus cycle. To avoid the last two situations, the chip select signal to the DUART should be qualified with DS* from the processor.

QUESTION
Describe the receiver FIFO of the MC68681/MC2681.

ANSWER
The DUART has a three-byte receiver FIFO that acts more like a circular queue. It has both a head pointer and a tail pointer. The head pointer is controlled by a read operation and is incremented to the next buffer location whenever a read of the receiver occurs. The tail pointer is incremented whenever a new character that has been assembled in the shift register is transferred to the receive holding register. After an external reset or the issuance of a reset receiver command, the head and tail pointers point to the same location in the FIFO. The contents of the FIFO are not flushed when a reset receiver command is issued. It takes three consecutive reads of the FIFO to move the head pointer in a circle until it gets back to its original location. Note that the head pointer can inadvertently be incremented if a monitor program that performs a read/verify cycle after a write cycle is in use.

The receiver ready bit (RxRDY) in either the interrupt status register (ISR) or the status register (SRx) should be polled before reading the

receiver. If the bit is set, the receiver should be read. The status register should again be read to determine the state of the RxRDY bit. If it is set again, the receiver should be read again. This should continue until the RxRDY bit is clear. If a read of the receiver is performed when the RxRDY bit is clear, the pointers will be incremented beyond the current valid data.

QUESTION
How can I detect the "end of break"?

ANSWER
In order to detect a break condition, the DUART receiver continuously samples the receive data input (RxD). When it senses a low for the start bit and the number of programmed bit times, it loads a zero character into the receive FIFO. If no stop bit is detected, the receiver samples beyond the character frame for one more bit time. If this bit is low, a framing error has occurred. Having the framing error bit in the status register set and a zero character in the receive FIFO forces the received break bit to be set in the ISR and the SRx.

The delta break bits in the ISR must be monitored on order to detect an end of break. The processor should not read the zero character in the receive FIFO nor clear the receive break bit in the SRx until the break is completely over. Once a break has been detected, the processor should issue a reset break change interrupt command, which resets the delta break bit in the ISR. The delta break bit will be set again whenever RxD transitions, indicating that the break condition is over. When this happens, the processor should issue a break change interrupt command, a reset error status command, and then read and discard the zero character in the receive FIFO.

QUESTION
What are the consequences of changing the DUART configuration without first disabling the receiver and transmitter?

ANSWER
Whenever you decide to write to the mode registers (MR1x and MR2x), the clock select registers (CSRx), or the auxiliary control register (ACR), the receiver and transmitter must first be disabled. If the mode registers change while serialization is still active, the transmission may be restarted under the new configuration. If serialization is complete and the transmit buffer is empty when the mode registers change, one of two things can happen; either TxD can go into the space condition for a short period of time or the transmitter ready bit (TxRDY) in the ISR may set and then reset. The above events are independent of the values written to the mode registers. In fact, rewriting the current values to the mode registers can produce the same results.

A more serious consequence may occur if you write to the CSRx or bit 7 of the ACR while the transmitter and receiver clocks are running. If the clocks are changed without first disabling the receiver and transmitter, clipped (shortened) clock pulses may appear during the change from one frequency to the next. These pulses may cause the receiver and transmitter to lock up.

The recommended (and best) way to disable the transmitter and receiver whenever you plan to change the configuration is to issue a software reset

command ($20 and $30 to the CRx). Not only does this disable the receiver and transmitter, it also places the DUART in a known state.

QUESTION
What programming sequence is necessary to operate in the Multidrop/Wake-up mode?

ANSWER
When the DUART is operating in the Multidrop mode, the transmitter sends data with the last bit of each character flagged as either address or data. Mode register 1, bit 2, determines whether the character being sent is an address or a data character. Thus, if you want to send an address character followed immediately by data characters, you must write to the mode register. Writing to the mode register without first resetting the receiver and transmitter could result in the incorrect transmission of data. Therefore, the following programming sequence should be observed when operating in the Multidrop/Wake-up mode:

1. Verify that the TxEMT bit is set to guarantee that the transmitter is not currently sending a character.
2. Reset the MR pointers, the receiver and the transmitter via the command register.
3. Load MR1 with the previously written data, insuring that MR1[2] = 1. Then, enable the receiver and transmitter.
4. Load the address character into the transmitter holding register (THR).
5. Wait until the TxEMT bit is set (character sent).
6. Reset the MR pointers, the receiver and the transmitter via the Command Register.
7. Load MR1 with the previous data, this time insuring that MR1[2] = 0. Then, enable the receiver and transmitter.
8. Load the data characters into the THR, until the message is complete. To send data to a different address, repeat steps 1-8.

Whenever a DUART in a secondary station recognizes an address character, it will set its RxRDY bit. The CPU must then immediately read the receiver to see if the accumulated address matches the station's address. If it does, the CPU must set RxEN in either CRA or CRB so that the message can be received. At higher baud rate frequencies, it might be difficult to perform the compare and the setting of the RxEN bit before the data begins to arrive. To alleviate this problem, you could enable the RxEN bit as soon as you receive the address character and then either reset the receiver if a match has not occurred or read the data characters from the receive FIFO if the address matches.

QUESTION
What are my options for driving the X1/CLK and X2 pins?

ANSWER
The DUART is very sensitive to its clock. The clock circuitry can be driven by either a crystal or a TTL-level signal. When using a TTL-level clock, the clock signal should be connected to X1/CLK and X2 should be grounded. If you do use a TTL-level clock to drive X1/CLK, you must guarantee a minimum high voltage of 4 Volts and a minimum high and low clock pulse width of 100 ns. The X1/CLK driver does not have to be an open collector.

The areas of most concern when using a crystal to drive the clock circuitry are the capacitance of C1 and C2, the duty cycle, and the rise and fall

times of the clock signal. Signetics recommends that the values of both capacitors be around 5 pF to insure proper charging during the power-on cycle. Our data sheet says that C1 should be between 10 and 15 pF and C2 should be between 0 and 5 pF. There are no known problems using these values.

Ideally, the duty cycle of the clock signal should be as close to 50% as possible. However, many DUARTs show a 60-40 duty cycle when hooked up to the crystal. To force a 50-50 duty cycle, add a 100 Kohm or greater resistor across X1/CLK and X2.

The rise and fall time problem may never occur assuming you have a good crystal to start with. Below is a list of vendors for 3.6864 MHz crystals that meet our specs. There are probably many more vendors; these are just the ones we know of.

Saronix, Palo Alto, CA (800) 227-8974
   inside Ca. (800) 422-3355, part #NMP037

Bomar Crystal, Middlesex, NJ (201) 356-7787

Crystek Corporation, Fort Myers, FL (813) 936-2109

Electro-Dynamics Corp., Shawnee Mission, KS
   (913) 262-2500

Allied Electronics, Fort Worth, TX (214) 265-9341
   part # 994-0345

US Crystals, Fort Worth, TX (800) 433-7140

Multiple DUARTs can be driven from the same crystal. Tap off the clock from X1/CLK, buffer it through an inverter, add a 1K pull-up resistor, and connect it to the other X1/CLK inputs (remember to ground X2 on these devices).

QUESTION
What is the potential problem with receiver-controlled RTS* negation?

ANSWER
When the receiver controls the negation of the RTS* output, a one should be written to the appropriate output port pin immediately after enabling the receiver. The receiver will negate RTS* whenever the FIFO is full and the start bit of a fourth character has been detected. Because of this, care must be taken in choosing the transmitter at the other end. Some transmitters, depending on the manufacturer, will stop transmission after sending out the character in the shift register while others stop after sending out the characters in both the holding register and the shift register. The latter transmitters will cause an overrun to occur in the DUART receiver.

QUESTION
How long must the pulse be on the input port pins before it is recognized internally?

ANSWER
The state change detection circuitry requires two successive samples of the

new state before the delta change bits are set in the input port change register (IPCR). This circuitry uses a 38.4 KHz sampling clock (generated from the baud rate generator). If the crystal frequency is 3.6864 MHz, then the sampling period of the internal state machine will be 25 microseconds. If the transition of the input pin occurs at the same time as the first sample pulse, then the new level must be present for 25 microseconds. If the level change occurs just after the sampling edge, the new level must be present for at least 50 microseconds to guarantee recognition.

QUESTION
Does an interrupt acknowledge cycle clear the bits in the interrupt status register (ISR) or do I have to clear them in software?

ANSWER
None of the bits in the ISR are cleared by an interrupt acknowledge cycle. The input port change status bit (ISR_7) is cleared when the processor reads the input port change register (IPCR), the channel A and channel B change in break bits (ISR_2 and _6) when the CPU issues a reset break change interrupt command for the corresponding channel, the channel A and B receiver ready or FIFO full bits (ISR_1 and _5) when the CPU reads the receiver buffer for the associated channel, the channel A and B transmitter ready bits (ISR_0 and _4) whenever the processor loads a character into the appropriate transmitter holding register, and the counter/timer ready bit (ISR_3) by a stop counter command in both the counter and timer modes.

QUESTION
When do the output ports go high after RESET is asserted?

ANSWER
The output ports (OP0-OP7) are placed in the logic high state about 80 to 90 ns after RESET asserts (asynchronous to the X1/CLK input).

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:10 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68840/850 FAQ
Entry Prepared On: Jul 22 17:16:14 1994

QUESTION
Does proper operation of the MC68440 require that the PCL line meet certain timing requirements in relation to the 10 MHz clock?

ANSWER
The MC68440 samples PCL on every clock of an active bus cycle, including the clock edge that causes AS to assert. This is not stated very clearly in the manual, and the implication of this timing is that if PCL is held too long, the next bus cycle might accidentally observe it as being asserted. This may result an erroneous recognition of the PCL signal on a subsequent bus cycle. The best way to avoid the unwanted PCL recognition on a subsequent bus cycle is to ensure that the PCL to the MC68440 is sampled negated prior to the beginning of the next bus cycle. It is highly recommended then, that the PCL signal be qualified with AS from the MC68440.

QUESTION
On the MC68440 and MC68450, I observe that the DMA devices occasionally negate BR for no apparent reason, even if the BG has been issued to them. Why is this so?

ANSWER
This is a normal operation for the MC68440 and MC68450. The reason for this behavior is to prevent a bus lockup in the event that the DMA asserts BR in response to a pending transfer, but at the same time, the processor tries to access the DMA registers. In this case, the MC68440 or MC68450 yields the bus to the processor, even if no DMA cycles has been run. Please note than all of the processors in the MC68000 family make provisions for this "never-mind" operation. Most common problems arise when using the MC68440 or MC68450 with an external bus arbiter that requires at least one bus cycle to be run per bus tenure.

It is highly recommended that before any external bus arbiters be designed, that the bus arbitration state diagrams for Motorola processors be studied.

QUESTION
On the MC68450, is it possible to perform a bus cycle in four clocks?

ANSWER
No, the best that the MC68450 can do is a five-clock bus cycle. The documentation is in error.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:10 1994

Topic Name: M68000 FAQs (by Device)
Subject:MC68EC00 FAQ
Entry Prepared On: Jul 22 17:35:43 1994

QUESTION
Is it okay to ground DTACK* if you are running an MC68EC000 with no wait states?

ANSWER
Yes, as long as all accesses (ROM, RAM, I/O, vector fetches) are zero-wait-state accesses, and you don't use autovectored interrupts since the MC68EC000 does not allow both AVEC* and DTACK* to be asserted at the same time.

QUESTION
Does the MC68EC000 have dynamic bus sizing?

ANSWER
The MC68EC000 has static bus sizing, but can operate in either 8-bit mode or 16-bit mode. The MODE input selects between the 8- and 16-bit operating modes. If this input is grounded at reset, the processor will come out of reset in the 8-bit mode. If this input is tied high or floating at reset, the processor will come out of reset in the 16-bit mode. This input should

be changed only at reset and must be stable two clocks after RESET* is negated. Changing this input during normal operation may produce unpredictable results.

+++++ End of AESOP Entry +++++

AESOP
Today Is Thu Sep 1 04:29:10 1994

Topic Name: M68000 FAQs (by Device)
Subject:Programmer's Reference Manual FAQ
Entry Prepared On: Jul 22 17:37:13 1994

QUESTION
How do I convert from hexadecimal object code to the mnemonic equivalents?

ANSWER
The Programmers Reference Manual (M68000PM/AD) provides the instructions in numerical order. To convert, match the hexadecimal object code with this listing, then turn to the instruction itself to decode the instruction fields (e.g., effective address, size, etc.)

+++++ End of AESOP Entry +++++