MOTOROLA

Communications and Advanced Consumer Technologies Group

Order this document by AN1703/D

AN1703

Application Note

Low Power on the SCM68000 (EC000 Core)

The SCM68000 (EC000 core) has been redesigned to provide low-power, fully static operation. This document describes the recommended method for placing the SCM68000 into low-power mode to reduce power consumption to its quiescent value[1] while maintaining the internal state of the processor. The low-power mode described below is part of the SCM68000 test vectors that Motorola provides.

Note: The terms *assertion* and *negation* are used in this document to avoid confusion when describing a mixture of "active-low" and "active-high" signals. The term *assert* or *assertion* indicates that a signal is active or true, regardless of whether a high or low voltage represents that level. The term *negate* or *negation* indicates that a signal is inactive or false. The names of all "active-low" signals end with the letter *B*.

RESETTING THE SCM68000

Although the SCM68000 is a fully static device, it cannot be put into low-power mode until after it has completed the reset exception processing that is described in **Section 4.3.1 Reset** of the *EC000 Core Processor User's Manual*.

Warning: During and after reset exception processing, the SCM68000 will not be in a low-power state. The processor must be reset with the clock running, which is detailed in **Section 4.3.1.1 Reset Operation** in the *EC000 Core Processor User's Manual*. Once the SCM68000 is properly reset, the internal circuitry is in a valid state and can be put into low-power mode. Low-power mode process that is described in this document must be followed or the SCM68000 will not be in a low-power state.

ENTERING LOW-POWER MODE

To successfully enter the low-power mode, the SCM68000 must be in the supervisor mode. The **trap** instruction should be used to enter low-power mode because it causes the processor to begin exception processing and enter supervisor mode. During the trap routine, the external circuitry should follow a sequence of steps to enable low-power mode.

---

1. The current drain specification for the SCM68000 while in low-power mode is I_{dd} < 2μA at 3.3V and <5μA at 5V.

Warning: If RESETIB is asserted while the SCM68000 is in low-power mode, the clock to the processor will be restarted and the processor will begin reset exception processing. Refer to **Section 4.3.1 Reset** in the *EC000 Core Processor User's Manual* to properly reset the device.

1. Externally detect a write to the low-power address. This user-selected address can be any in the 4G addressing range of the SCM68000. A write to the low-power address can be detected when RWB is a logic low, the function codes have a five (101) on their output, and the address bus contains the low-power address. At this time, the user-designed circuitry should assert the ADDRESS_MATCH signal shown in Figure 1 or 2.

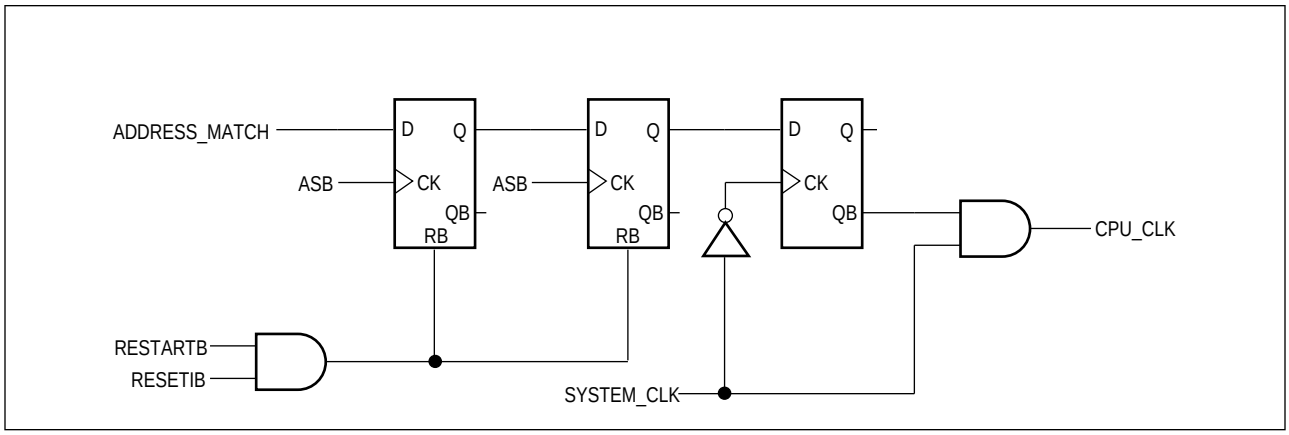

Figure 1. Low-Power Circuitry for 16-Bit Data Bus

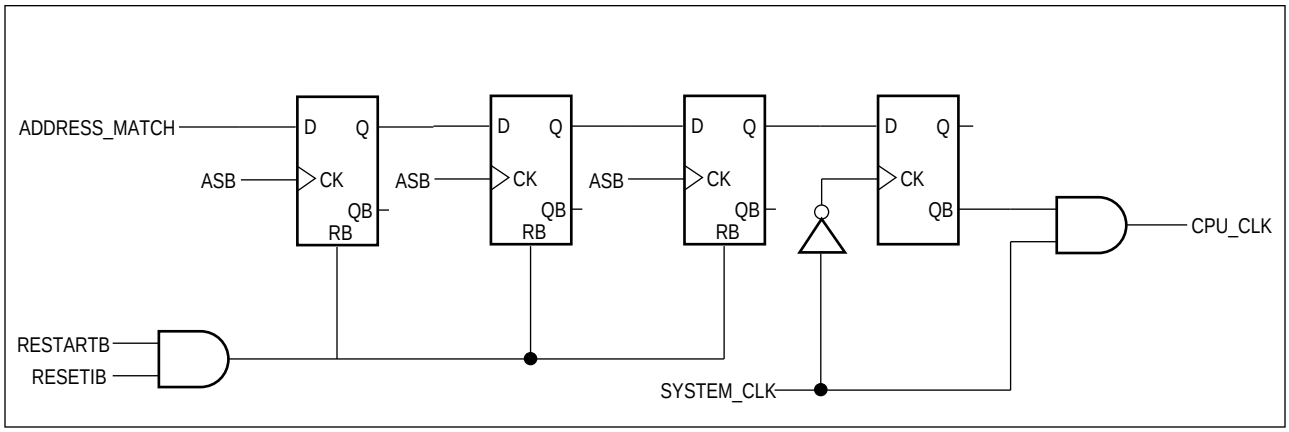

Figure 2. Low-Power Circuitry for 8-Bit Data Bus

2. Execute the **stop** instruction. Using the **stop** instruction, set the interrupt mask level to bring the SCM68000 out of low-power mode. For example, the instruction **stop** #$2300 sets the interrupt mask level to 3 and any interrupt higher than level 3 will tell the processor to start executing instructions. The external circuitry shown in Figures 1 and 2 will count the number of bus cycles starting with the write to the low-power address. It will stop the processor's clock on the first falling edge of the system clock after the bus cycle that reads the immediate data of the **stop** instruction. Figure 2 has one more flip-flop than Figure 1 because the SCM68000 in 8-bit mode requires two bus cycles to fetch the immediate data of the **stop** instruction. After the processor's clock is disabled, you can disable the clock to other sections of your system. Use caution to ensure that runt clocks and spurious glitches are not presented to the SCM68000. Figures 3 and 4 illustrate the timing diagrams for the 8- and 16-bit modes, respectively.

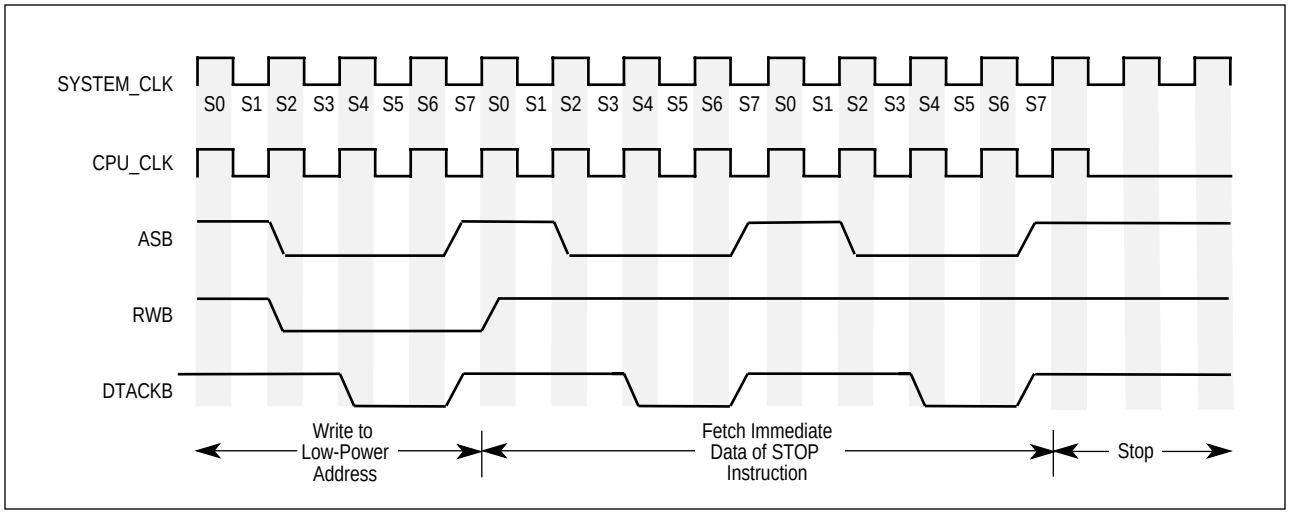

Figure 3. Clock Stop Timing for 8-Bit Data Bus

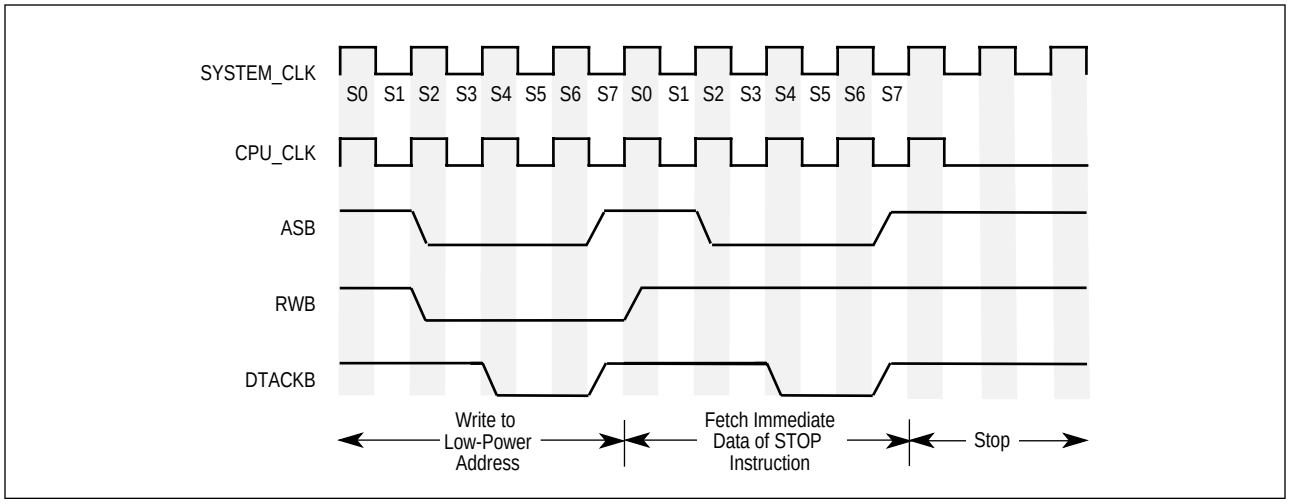

Figure 4. Clock Stop Timing for 16-Bit Data Bus

Note: While the SCM68000 is in low-power mode, all inputs must be driven to V_{DD} or V_{SS} or have a pull-up or pull-down resistor.

3. This step is optional depending on whether your applications need the three-stateable signals of the SCM68000 to be placed in a high-impedance state. To place the SCM68000 into a three-state condition, complete the proper method for arbitrating the bus during the status register data fetch for the **stop** instruction. The timing diagrams with the bus arbitration sequence are illustrated in Figure 5 (8-bit mode) and Figure 6 (16-bit mode). Refer to **Section 3.2 Bus Arbitration** in the *EC000 Core Processor User's Manual* for more information.

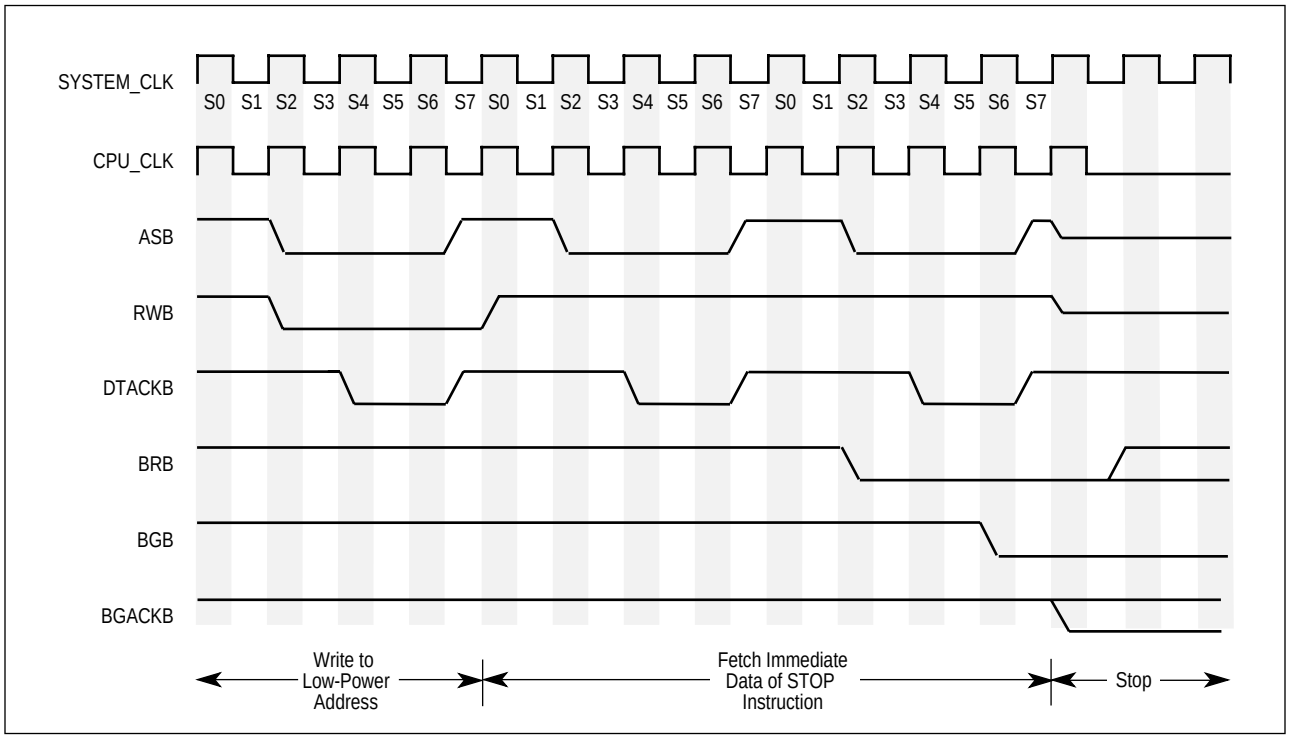

Figure 5. Clock Stop Timing with Bus Arbitration for 8-bit Data Bus

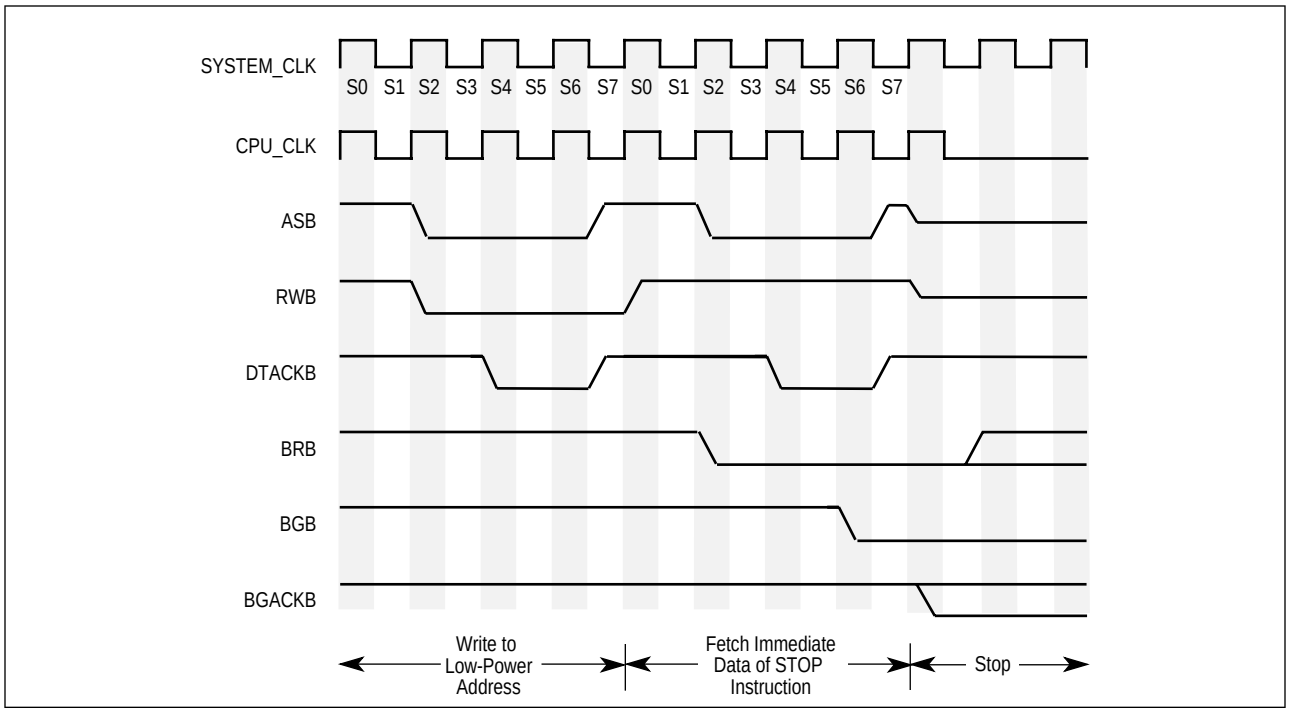

Figure 6. Clock Stop Timing with Bus Arbitration for 16-bit Data Bus

EXITING LOW-POWER MODE

Once you complete the previous steps, the SCM68000 remains in low-power mode until the clock is restarted. To restart the clock, external logic must poll IPLB2–IPLB0 to detect an interrupt higher than the one set in the **stop** instruction in step 2 of the previous section. When the correct interrupt level is received, follow these steps to bring the processor out of low-power mode:

1. Restart the system clock if it was stopped.
2. Wait for the system clock to become stable.

Assert the RESTARTB signal. This will cause the processor's clock to start on the next falling edge of the system clock. Figure 7 shows the timing for bringing the processor out of low-power mode. Both the RESTARTB and RESETB signals are subject to the asynchronous setup time that is specified in **Section 7 Electrical Characteristics** of the *EC000 Core Processor (SCM68000) User's Manual*.

Note: The processor will not recognize an interrupt until the clock has been restarted. If the external logic asserts RESTARTB with an interrupt level lower than the processor mask level, the **stop** instruction will not complete even though the clock has been restarted. To prevent glitches, the system clock must be stable before the RESTARTB signal can be asserted. An unstable clock can cause unpredictable results in the SCM68000.

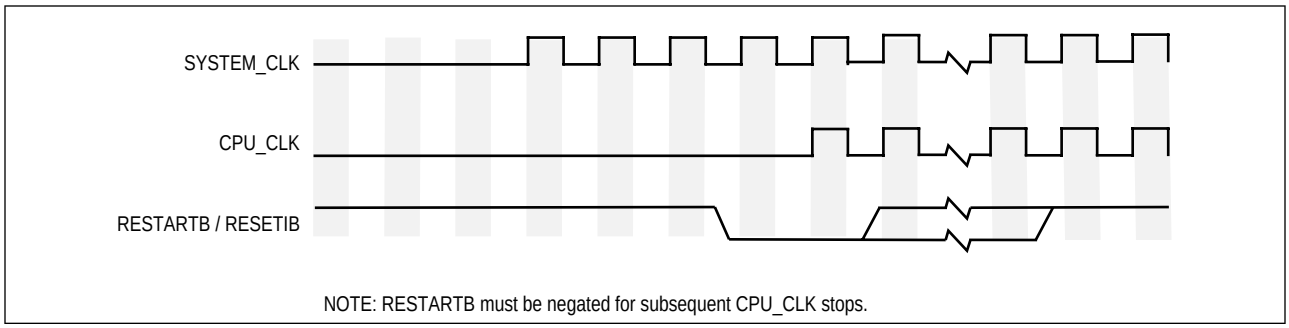

Figure 7. Clock-Start Timing

3. If the SCM68000 was placed into a three-state condition, the BGACKB signal (used for 3-wire bus arbitration) or the BRB signal (used for 2-wire bus arbitration) must be negated before the processor can execute instructions.

EXAMPLE TRAP ROUTINE

```
TRAP_xMOVE.B #0,$low_power_address/* Write that causes ADDRESS_MATCH to
assert */
STOP #$2000/* STOP instruction with desired
interrupt mask */
RTE/* Return from the exception */
```

The first instruction (**move.b** #0,$low_power_address) writes a byte to the low-power address that notifies the external circuitry to begin the sequence that will stop the processor's clock. The second instruction (**stop** #$2000) is the **stop** instruction that loads the status register with the immediate data that allows you to set the interrupt that will make the processor exit low-power mode. The final instruction (**rte**) signals the processor to return from the exception and resume normal processing.

Literature Distribution Centers:
USA/EUROPE: Motorola Literature Distribution; P.O. Box 20912, Arizona 85036.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan.
ASIA-PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.

SEMICONDUCTOR PRODUCT INFORMATION