

The \$03FE interrupt handler must do these things:

1. Verify that the interrupt came from the expected source.
2. Handle the interrupt as desired.
3. Clear the appropriate interrupt soft switch.
4. Return with an RTI.

Here are some things to remember if you are dealing with programs that must run in an interrupt environment:

- There is no guaranteed maximum response time for interrupts because the system may be doing a disk operation that lasts for several seconds.
- Once the built-in interrupt handler is called, it takes *at least* 150 to 200 microseconds for it to call your interrupt-handling routine. After your routine returns, it takes 40 to 140 microseconds to restore memory and return to the interrupted program.
- If memory is in the standard state when the interrupt occurs, the total overhead for interrupt processing is about 150 microseconds less than if memory is in the worst state. (The worst state is one that requires the most work to set up for: 80STORE and PAGE2 on; auxiliary memory switched in for reading and writing; bank-switched memory page 2 in the auxiliary bank switched in for reading and writing; and internal \$Cn00 ROM switched in).
- Interrupt overhead will be greater if your interrupt handler is installed through an operating system's interrupt dispatcher. The length of delay depends on the operating system, and on whether the operating system dispatches the interrupt to other routines before calling yours.

Table 6-9
BRK handler information

Information	Location
Program counter (low byte)	\$3A
Program counter (high byte)	\$3B
Encoded memory state	\$44
Accumulator	\$45
X register	\$46
Y register	\$47
Status register	\$48

Handling break instructions

The 65C02 treats a break instruction (BRK, opcode \$00) just like a hardware interrupt. After the interrupt handler sets the memory configuration, it checks to see if the interrupt was caused by a break (bit 4 of the status byte is set) and, if it was, jumps to a break-handling routine. This routine saves the state of the computer at the time of the break as shown in Table 6-9.

Finally the break routine jumps to the routine whose address is stored at \$3F0 and \$3F1.

The encoded memory state in location \$44 is interpreted as shown in Table 6-10.