System Engineering Notes
No. 843
SYM53C7X0 Interrupts
Rev 2.0, May 1997

The SCRIPTS processors in the SYM53C7X0 family perform most functions independently of the host microprocessor. However, certain interrupt situations must be handed by the external microprocessor. This application note explains all aspects of interrupts as they apply to the SYM53C700, SYM53C700-66, SYM53C710, and with a few modifications, the SYM53C720. The register names and addresses used in the body of the note apply to the SYM53C700, SYM53C700-66 and SYM53C710. The register differences for the SYM53C720 are discussed at the end of the note.

Polling vs. Hardware Interrupts

The external microprocessor can be informed of an interrupt condition by polling or hardware interrupts. Polling means that the microprocessor must continually loop and read a register until it detects a bit set that indicates an interrupt. This method is the fastest, but it wastes CPU time that could be used by other system tasks. The preferred method of detecting interrupts in most systems is hardware interrupts. In this case, the SYM53C7X0 will assert the Interrupt Request (IRQ/) line that will interrupt the microprocessor, causing the microprocessor to execute an interrupt service routine. A hybrid approach would use hardware for long waits, and use polling for short waits.

Registers

The five registers in the SYM53C7X0 that are used for detecting or defining interrupts are the ISTAT (register 0x21), the SSTAT0 (register 0x0D), the DSTAT (register 0x0C), the SIEN (register 0x03), and the DIEN (register 0x39).

The ISTAT is the only register that can be accessed as a slave during SCRIPTS operation, therefore it is the register that is polled when polled interrupts are used. It is also the first register that should be read when the IRQ/ pin has been asserted in association with a hardware interrupt. If the SIP bit in the ISTAT register is set, then a SCSI-type interrupt has occurred and the SSTAT0 register should be read. If the DIP bit in the ISTAT register is set, then a DMA-type interrupt has occurred and the DSTAT register should be read. SCSI-type and DMA-type interrupts may occur simultaneously, so in some cases both SIP and DIP may be set.

The SSTAT0 register contains the SCSI-type interrupt bits. Reading this register will determine which condition or conditions caused the SCSI-type interrupt, and will clear that SCSI interrupt condition. If the SYM53C7X0 is receiving data from the SCSI bus and a fatal interrupt condition occurs, the SYM53C7X0 will attempt to send the contents of the DMA FIFO to memory before generating the interrupt. If the SYM53C7X0 is sending data to the SCSI bus and a fatal SCSI interrupt occurs, data could be left in the DMA FIFO. Because of this the DFE bit in DSTAT should be checked. If this bit is clear, set the CLF (Clear DMA and SCSI FIFOs) bit before continuing.

The DSTAT register contains the DMA-type interrupt bits. Reading this register will determine which condition or conditions caused the DMA-type interrupt, and will clear that DMA interrupt condition. Bit 7 in DSTAT, DFE (DMA FIFO Empty), is purly a status bit; it will not generate an interrupt under any circumstances and will not be cleared when read. DMA interrupts will flush neither the DMA not SCSI FIFOs before generating the interrupt, so the DFE in the DSTAT register should be checked after any DMA interrupt. If the DFE bit is clear, then the FIFOs must be cleared by setting the CLF (Clear DMA and SCSI FIFOs) bit, or flushed by setting the FLF (Flush DMA FIFO) bit. The CLF bit is bit 6 in the DFIFO register on the SYM53C700 and SYM53C700-66, bit 2 in CTEST8 in the SYM53C710, and bit 2 in CTEST3 in the SYM53C720. The FLF bit is bit 7 in the DFIFO register in the SYM53C700 and SYM53C700-66, bit 3 in CTEST8 in the SYM53C710, and bit 3 in CTEST3 in the SYM53C720.

The SIEN register is the interrupt enable register for the SCSI interrupts in SSTAT0.

The DIEN register is the interrupt enable register for DMA interrupts in DSTAT.

Fatal vs. Non-Fatal Interrupts

A fatal interrupt, as the name implies, always causes SCRIPTS to stop running. A non-fatal interrupt will cause SCRIPTS to stop running only if it is not masked. Masking will be discussed later in this engineering note. All DMA interrupts (indicated by the DIP bit in ITSAT and one or more bits in DSTAT being set) are fatal.

Some SCSI interrupts (indicated by the SIP bit in the ISTAT and one or more bits in SSTAT0 being set) are non-fatal. When the chip is operating in Initiator mode, only the CMP (Function Complete) and SEL(Selected or Reselected) interrupts are non-fatal. When operating in Target mode CMP, SEL, and M/A (Target mode: ATN/ active) are non-fatal. Refer to the description for the DHP (Disable Halt on a Parity Error or ATN/ active (Target Mode Only)) bit in the SXFER register to configure the chip’s behavior when the ATN/ interrupt is enabled during Target mode operation.

The reason for non-fatal interrupts is to prevent SCRIPTS from stopping when an interrupt occurs that does not require service from the CPU. This prevents an interrupt when arbitration is complete (CMP set), when the SYM53C7X0 has been selected or reselected (SEL set), or when the initiator has asserted ATN (target mode: ATN/ active). These interrupts are not needed for events that occur during high-level SCRIPTS operation.

Masking

Masking an interrupt means disabling or ignoring that interrupt. Interrupts can be masked by clearing bits in the SIEN (for SCSI interrupts) register or DIEN (for DMA interrupts) register. How the chip will respond to masked interrupts depends on: whether polling or hardware interrupts are being used; whether the interrupt is fatal or non-fatal; and whether the chip is operating in Initiator or Target mode.

If a non-fatal interrupt is masked and that condition occurs, SCRIPTS will not stop, the appropriate bit in the SSTAT0 will still be set, the SIP bit in the ISTAT will not be set, and the IRQ/ pin will not be asserted. See the section on non-fatal vs. fatal interrupts for a list of the non-fatal interrupts.

If a fatal interrupt is masked and that condition occurs, then SCRIPTS will still stop, the appropriate bit in the DSTAT or SSTAT0 register will be set, the SIP or DIP bits in the ISTAT will be set, and the IRQ/ pin will not be asserted.

When the chip is initialized, enable all fatal interrupts if you are using hardware interrupts. If a fatal interrupt is disabled and that interrupt condition occurs, SCRIPTS will halt and the system will never know it unless it times out and checks the ISTAT after a certain period of inactivity.

If you are polling the ISTAT instead of using hardware interrupts, then masking a fatal interrupt will make no difference since the SIP and DIP bits in the ISTAT inform the system of interrupts, not the IRQ/ pin.

Masking an interrupt after IRQ/ is asserted will not cause IRQ/ to be deasserted.

Stacked Interrupts

The SYM53C7X0 has the ability to stack interrupts if they occur one after the other. If the SIP or DIP bits in the ISTAT register are set (first level), then there is already at least one pending interrupt and any future interrupts will be stacked in extra registers behind the SSTAT0 and DSTAT registers (second level). When two interrupts have occurred and the two levels of the stack are full, any further interrupts will set additional bits in the extra registers behind SSTAT0 and DSTAT. When the first level of interrupts are cleared, all the interrupts that came in afterward will move into the SSTAT0 and DSTAT. After the first interrupt is cleared by reading the appropriate register, the IRQ/ pin will be deasserted for a set time as published in the product Data Manual; the stacked interrupt(s) will move into the SSTAT0 or DSTAT; and the IRQ/ pin will be asserted once again.

Since a masked non-fatal interrupt will not set the SIP or DIP bits, interrupt stacking will not occur as a result of a masked, non-fatal interrupt. A masked, non-fatal interrupt will still post the interrupt in SSTAT0 but will not assert the IRQ/ pin. Since no interrupt is generated, future interrupts will move right into the SSTAT0 instead of being stacked behind another interrupt. When another condition occurs that generates an interrupt, the bit corresponding to the earlier masked non-fatal interrupt will still be set.

A related situation to interrupt stacking is when two interrupts occur simultaneously. Since stacking does not occur until the SIP or DIP bits are set, there is a small timing window in which multiple interrupts can occur but will not be stacked. These could be multiple SCSI interrupts (SIP set), multiple DMA interrupts (DIP set), or multiple SCSI and multiple DMA interrupts (both SIP and DIP set).

As previously mentioned, DMA interrupts will not attempt to flush the FIFOs before generating the interrupt. It is important to set either the CLF (Clear DMA and SCSI FIFOs) bit or the FLF (Flush DMA FIFO) bit if a DMA interrupt occurs and the DFE (DMA FIFO Empty) bit is not set. This is because any future SCSI interrupts will not be posted until the DMA FIFO is clear of data. These ‘locked out’ SCSI interrupts will be posted as soon as the DMA FIFO is empty.

Halting in an Orderly Fashion

When an interrupt occurs, the SYM53C7X0 will attempt to halt in an orderly fashion.

- If in the middle of an instruction fetch, the fetch will be completed, except in the case of a Bus Fault or Watchdog Timeout. Execution will not begin, but the DSP will point to the next instruction since it is updated when the current Script is fetched.

- If the DMA direction is a write to memory and a SCSI interrupt occurs, the SYM53C7X0 will attempt to flush the DMA FIFO to memory before halting. Under any other circumstances only the current cycle will be completed before halting, so the DFE bit in DSTAT should be checked to see if any data remains in the DMA FIFO.
- SCSI REQ/ACK handshakes that have begun will be completed before halting.
- The SYM53C7X0 will attempt to clean up any outstanding synchronous offset before halting.
- In the case of Transfer Control Instructions, once instruction execution begins it will continue to completion before halting.
- If the instruction is a JUMP/CALL WHEN <phase>, the DSP will be updated to the transfer address before halting.
- All other instructions may half before completion.

Sample Interrupt Service Routine

The following is a sample of an interrupt service for the SYM53C7X0. It can be repeated if polling is used, or should be called when the IRQ/ pin is asserted if hardware interrupts are used.

1. Read ISTAT
2. If only the SIP bit is set, read SSTAT0 to clear the SCSI interrupt condition and get the SCSI interrupt status. The bits in the SSTAT0 tell which SCSI interrupt(s) occurred and determine what action is required to service the interrupt(s).
3. If only the DIP bit is set, read the DSTAT to clear the interrupt condition and get the DMA interrupt status. The bits in the DSTAT will tell which DMA interrupt(s) occurred and determine what action is required to service the interrupt(s).
4. If both the SIP and DIP bits are set, read SSTAT0 and DSTAT as a word to clear the SCSI and DMA interrupt condition and get the interrupt status. If using 8-bit reads of the SSTAT0 and DSTAT registers to clear interrupts, insert 10 CLKs on the SYM53C700 and SYM53C700-66, and 12 CLKs on the SYM53C710 between the consecutive reads to ensure that the interrupts clear properly. Both the SCSI and DMA interrupt conditions should be handled before leaving the ISR. It is recommended that the DMA interrupt be serviced before the SCSI interrupt because a serious DMA interrupt condition could influence how the SCSI interrupt is acted upon.
5. When using polled interrupts, go back to step 1 before leaving the interrupt service routine, in case any stacked interrupts moved in when the first interrupt was cleared. When using hardware interrupts, the IRQ/ pin will be asserted again if there are any stacked interrupts. This should cause the system to re-enter the interrupt service routine.

Register Differences for the SYM53C720

Interrupts in the SYM53C720 work the same way as described above except that the SCSI registers are expanded and the names have changed. The SYM53C720 replaces the SSTATO register with two registers, the SIST0 and SIST1, and replaces the SIEN with the SIEN0 and SIEN1. When the above routines refer to reading SSTAT0, both the SIST0 and SIST1 registers on the SYM53C720 should be read. Refer to the SYM53C720 Data Manual for more information.