Chapter 3
Concatenating Counters

CONCATENATING COUNTERS

The Am9513 counters may be concatenated in a number of different ways. These may be conceptually broken down into count up and count down concatenation. Count up concatenation will typically be used to count events with a precision greater than 16 bits. Count down concatenation is typically used to generate output frequencies of high resolution.

To simplify concatenation, the Am9513 provides an internal TC signal from the low order counter which can be selected as a count source in the high order counter's Counter Mode register. Thus, although any two counters can be concatenated with external strapping, usually adjacent counters will be used to allow use of this internal TC signal.

In count up concatenation, both the high and low order counter's Load register should be cleared to 0. The low order counter will start counting up from 0 and increment through 9999. (BCD counting is assumed throughout this discussion, although binary counting may, of course, be used). On the next source edge the low order counter will go to TC and reload 0 from the Load register. The active-going TC edge will also increment the high order counter. The counters continue counting in this manner with the high order counter incrementing each time the low order counter reaches TC. In the examples which follow, Counters 1 and 2 will be used as the low order and high order counters respectively.

In the first up concatenation configuration, shown in Figure 3-1, the counters do not use external gating and therefore will free run. The high order counter should use the TC output of the low order counter as a source. The high order counter should count on rising source edges and should be programmed for "no gating." The above requirements can be met by specifying 00 (hex) in the upper byte of the high order counter's Mode register. The low order counter should be programmed to count repetitively. The required Mode register settings for Counters 1 and 2 are shown in the figure; "don't care" bits are marked "X." Note that if the internal TC signal is used to concatenate to the upper counter, no restrictions are placed on the programming of the low order counter's Output Control field. Conversely, if external strapping is used to concatenate the counter, the low order counter should have an "Active High TC" output mode selected. Up count concatenation may also be used with either level or edge gating. For level gating, the count source may either be externally gated with external logic, or the low order counter may be programmed for level gating, as shown in Figure 3-2. In either case, the high order counter should be programmed for "no gating." Recall that while in the TC state, the counters will count all source pulses issued to them, irrespective of their gating or arming status. This can introduce counting errors when level gating is used in up count concatenation. If the gate goes inactive while the low order counter is in TC, the low order counter will count the next source edge, which drives it out of TC. The counter will then stop counting until the gate goes active again. This effectively introduces a 1 count error into the accumulated count. The maximum error that can be introduced is one extra count each time the gate is applied. This worst case error will occur only if the gate is always applied when the low order counter is in the TC state. For many applications which use the gate infrequently, this small potential error is of no significance. Applications sensitive to small count errors or applications with many gate-on, gate-off cycles should use external gating logic to inhibit source pulses.

Edge gating functions can also be used in up count concatenation. An edge gating circuit with concatenated counters should function in a logically identical manner to a single edge-gated counter. In other words, after an edge is applied to the concatenated counters, they should count until both reach TC. A new edge should be required to repeat the cycle. Direct concatenation of two counters as was done for level gating up count

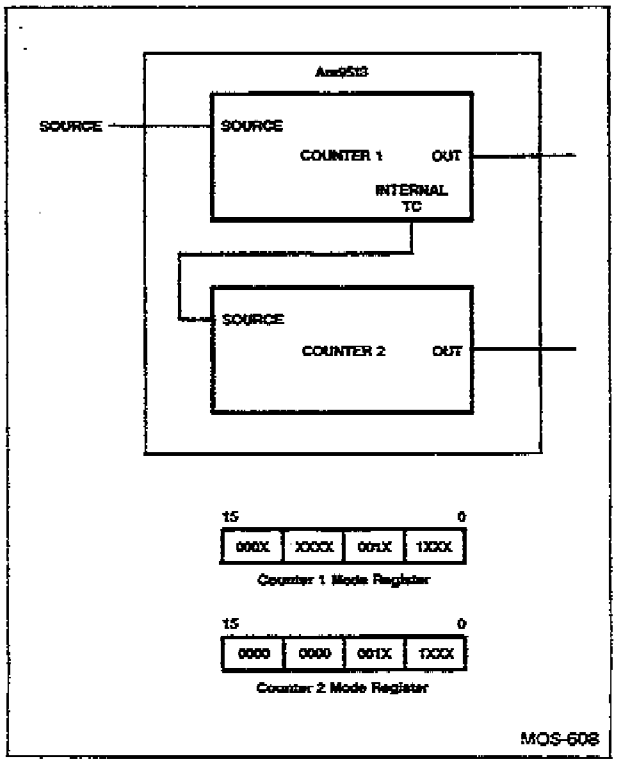

Figure 3-1. Count Up Concatenation with No Gating

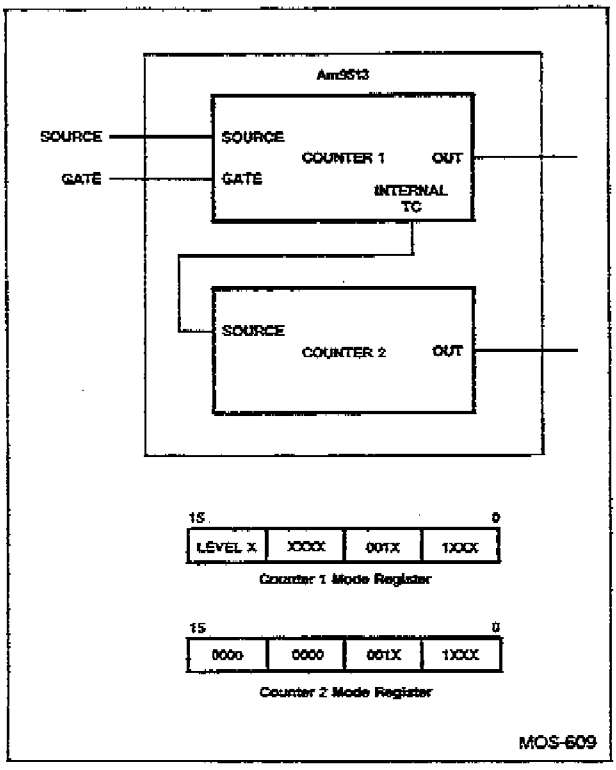

Figure 3-2. Count Up Concatenation with Level Gating

concatenation will not work. In such an arrangement, the low order counter, once triggered, will count to TC once and then stop, awaiting a new gate edge. This is unsatisfactory since we want the low order counter to continue counting until the high order counter reaches TC.

Figure 3-3 shows one method of concatenating counters for edge-triggered up counting. This method operates the counters in a similar arrangement to that used for level gating, with the requirement that each counter's output be programmed for an active-high TC pulse.

The external flip-flop is set by an external, synchronous gate signal. When both counters reach TC, the flip-flop is cleared. One potential problem exists with this scheme. Once the flip-flop clears, it will inhibit the low order counter's gate input. The low order counter will nevertheless count the next source edge, driving itself out of TC. However, the high order counter will remain in TC. When the next triggering gate edge is applied, the flip-flop will set, allowing counting to begin. When the low counter reaches its first TC, the rising TC edge will cause the high counter to leave TC. For a short period of time (the propagation delay of the high order counter from source to output), both TCs will again be active. This could potentially clear the flip-flop prematurely. To inhibit this, the source signal is added as an additional input to the NAND gate. If a relatively slow source is used with a high time greater than the total propagation delay from the source input of low order counter to the output of the high order counter, the source input on the NAND gate will inhibit clearing of the flip-flop during this transient.

The concatenation examples so far have assured that the counters are to "count repetitively," in the sense of counter mode register bit CM5. If "count once" operation is desired, in which the counters require an Arm command after each count cycle, different circuits are required.

When "no gating," "count once" operation is desired, the circuit in Figure 3-4 can be used. In this application, Counter 1 should be programmed for active-high level gating and Counter 2 should be programmed for a TC Toggled output. During counter initialization, the following set of commands should be used:

Initialize Counters' 1 and 2 Mode and Load registers
LOAD Counters 1 and 2
Clear Counter 2 output
ARM Counters 1 and 2.

The counters are now ready to count, but since Counter 2's output is low, Counter 1's gate will inhibit counting. To start counter operation, use the "Set Counter 2's output" command. The counters will then count applied source pulses until Counter 2 reaches TC and toggles its output, inhibiting Counter 1's gate. It can be seen that in this application, the "Set Counter 2's output" behaves as an ARM command. It is important that the counting rate be low enough to ensure that Counter 1's gate will not go inactive in close proximity to a source edge. High speed applications using a Counter 1 source period less than the propagation delay from Counter 1's source to Counter 2's output should use a flip-flop to synchronize Counter 2's output to Counter 1's source in order to meet timing parameters TGVEH and TEHGV in the Am9513 data sheet. High-speed applications will end the count cycle with a value slightly larger than 1 in Counter 1.

To add level gating to this "count once" feature simply involves the addition of an AND function before Counter 1's gate input. Now Counter 1 will be inhibited whenever Counter 2 toggles its output or whenever the external gate is driven low. Note that this circuit assumes the externally applied gate is synchronous to the count source; asynchronous gating signals should be synchronized with a flip-flop.

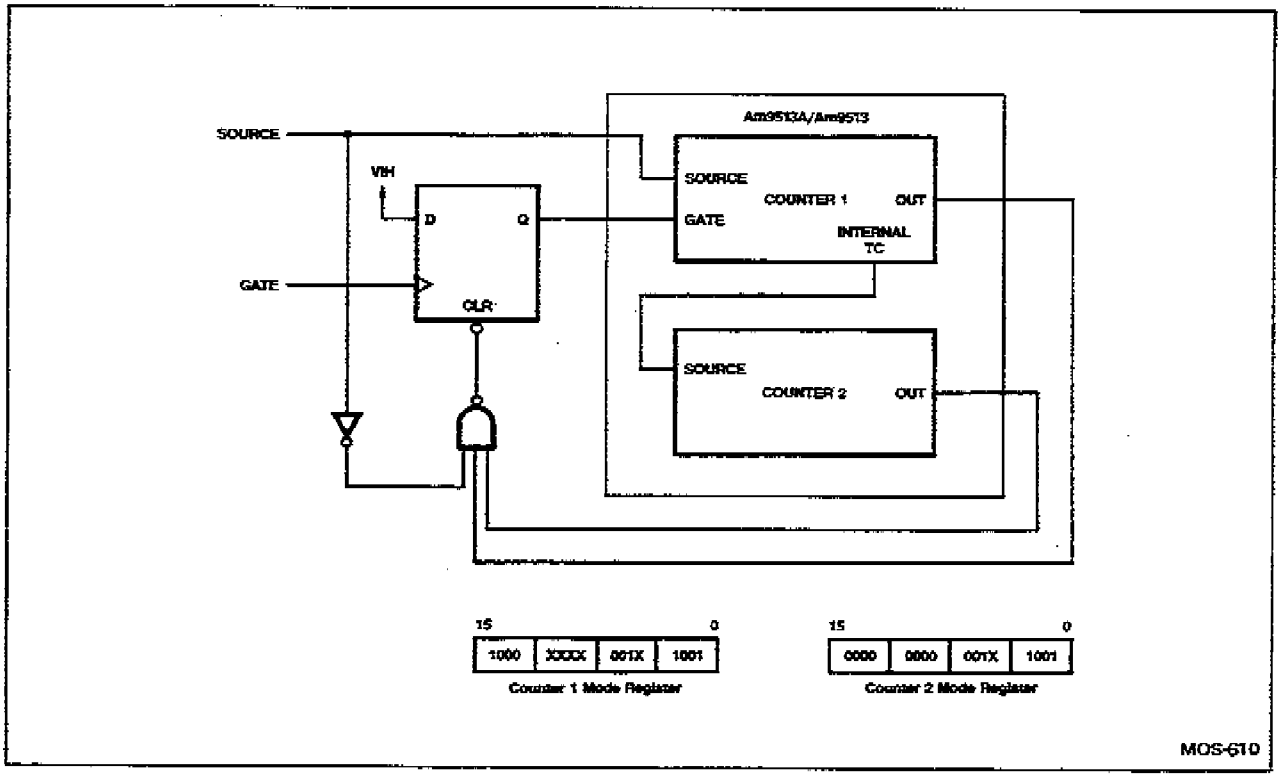

Figure 3-3. Count Up Concatenation with Edge Gating

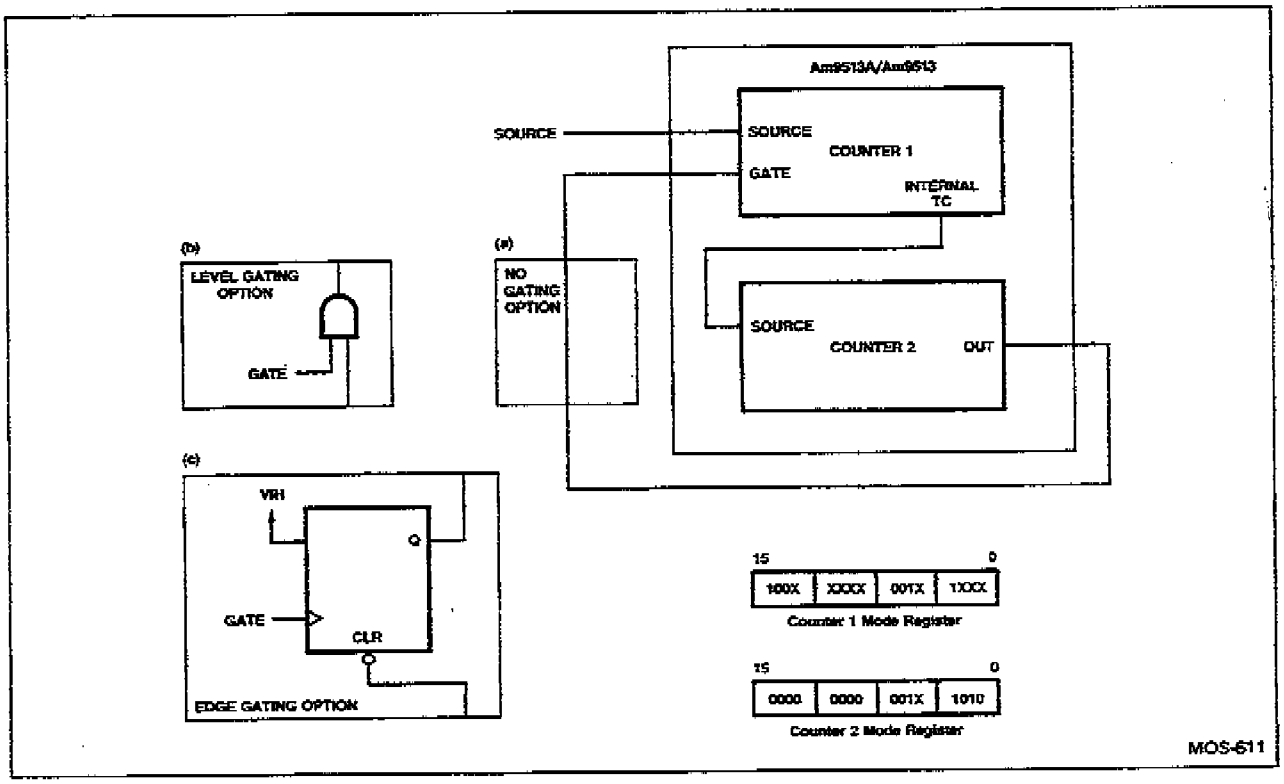

Figure 3-4. Count Up Concatenation with Count Once Feature

The final case of concatenated up counting comprises edge gating with the "count once" feature. This is achieved through a simple variation of the level gating configuration. An external gate signal sets the flip-flop and enables counting providing Counter 2's output is set. When Counter 2 reaches TC, its output will toggle (i.e., clear) and the flip-flop will clear, inhibiting further counting. To restart the counter in this configuration, the "Set Counter 2 output" command should be issued and a new gate edge should be applied in the order. As in the previous cases, the applied gate edge should be synchronous to the Counter 1 source.

In order to analyze down concatenation, it is useful to separately analyze the sequences followed for the high order and low order counters. Figure 3-5 shows a typical count down concatenation sequence, with the high order and low order count sequences labelled. The high order counter simply decrements from some initial value L until TC is reached. (In the following discussion and figures, L and H are used to represent the Load and Hold register contents respectively; K and N are used to represent arbitrary count values.) It is then reloaded with L and repeats the sequence. Note that the high order counter, in general, will never count to 0, since TC is generated by the source edge occurring while the counter contains 1 and TC reloads the initial value L. The count sequence is thus L, (L−1), . . . , 2, 1, L, (L−1), (L−2), . . . , 2, 1, L. The low order counter starts from some initial value H and counts down to TC. This TC output will be used to decrement the high order counter by 1. The low order counter is now reloaded with 0 and counts down through (assuming BCD counting) 9999 to 1. This sequence of reloading 0 and counting down to the next

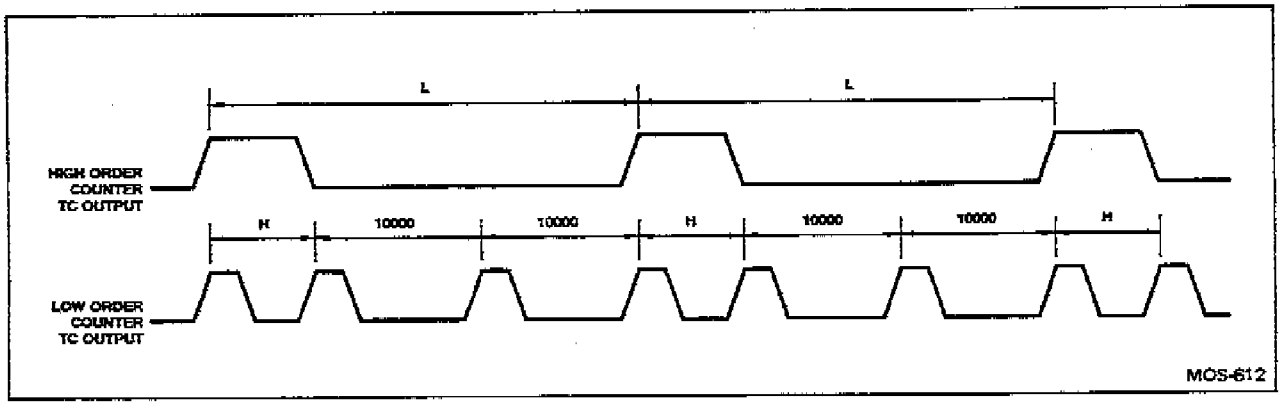

Figure 3-5. Conceptual Sequence for Count Down Concatenation

TC will be repeated by the low order counter until the high order counter has decremented to 1. On the next low order TC, the high order counter is driven to TC and reloads L. The low order counter should reload H, rather than 0, and repeat the complete count cycle. It can be seen that an important characteristic of the low order counter is that it reloads H once for each high order TC, and reloads 0 otherwise. This need for the low order counter to selectively reload 0 or H differs from up concatenation where the low order counter is always reloaded with the same value (0).

Figure 3-6 ties the above considerations together in a count repetitively, no gating, count down concatenation example. The low order counter is operated in Mode V, in which the gate is used

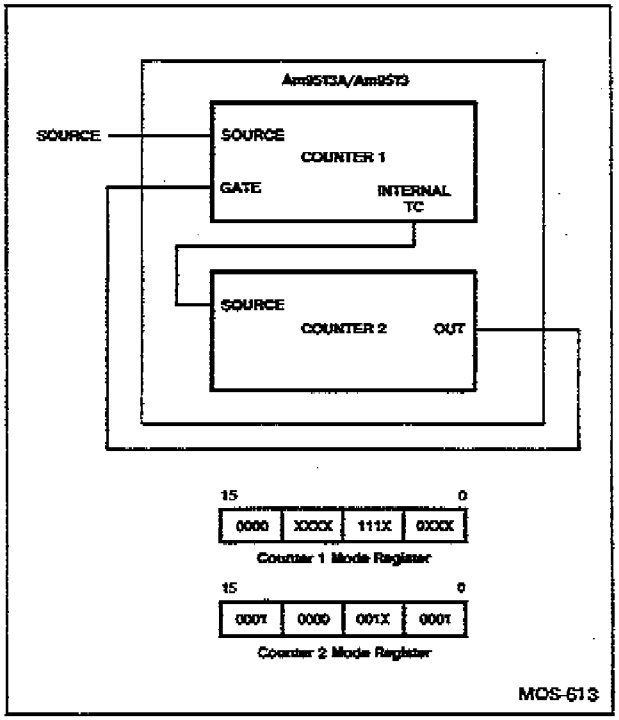

Figure 3-6. Count Down Concatenation

to select either the Load or Hold register as a reload source. The high order counter is operated in Mode D, with an active-high TC output selected in order to properly drive the low order counter's gate. In addition, the high order counter should be programmed to count on falling edges of the low order counter's internal TC output. Figure 3-7 shows timing waveforms generated by this concatenation configuration. Note that the count sequence generated never has 0 in the upper counter (disregarding the special case where L = 0). This means that the value stored in the high order counter should be biased by adding 1 in order to generate the correct divider ratio. For example, to divide by 39264178 (BCD), the high order counter's Load register should be set to 3926 + 1 = 3927 and the low order counter's Hold register should be set to 4178. The low order counter's Load register should be set to 0 to ensure proper count value rollover. Also note the unusual count sequence on the TC before the low order counter reloads from the Load register. For the above example of dividing by 39264178 the counters will count 00010002, 00010001, 00010000, 39279999, 39279998, ... , 39270002, 39270001, 39274178, 39264177, 39264176, 39264175, . . . rather than 00010002, 00010001, 00010000, 39274178, 39274177, . . . , 39270002, 39270001, 39270000, 39269999, 39269998, 39269997. . . . .

In some applications it may be desirable to level or edge gate with down concatenation. Because the low order counter uses the gate to select the reload source, the gate input cannot be used to start and stop counting in the low order counter. Accordingly, external gating logic must be used. Figure 3-8 shows the connections required for count down concatenations with level gating. Level gating is achieved by inhibiting source pulses when the gate goes inactive.

Edge gating, shown in Figure 3-9, uses an external gate signal to set an enabling flip-flop. The enabling flip-flop is cleared when both counters reach TC. The delay flip-flop ensures that one additional count occurs after both counters reach TC in order to drive the low order counter out of TC, thereby deactivating the enabling flip-flop's clear input. Note that the counters stop at an unusual point in the count sequence, ((L−1), (H−1) in Figure 3-7 or 3926 4177 for the earlier example) but this is not important

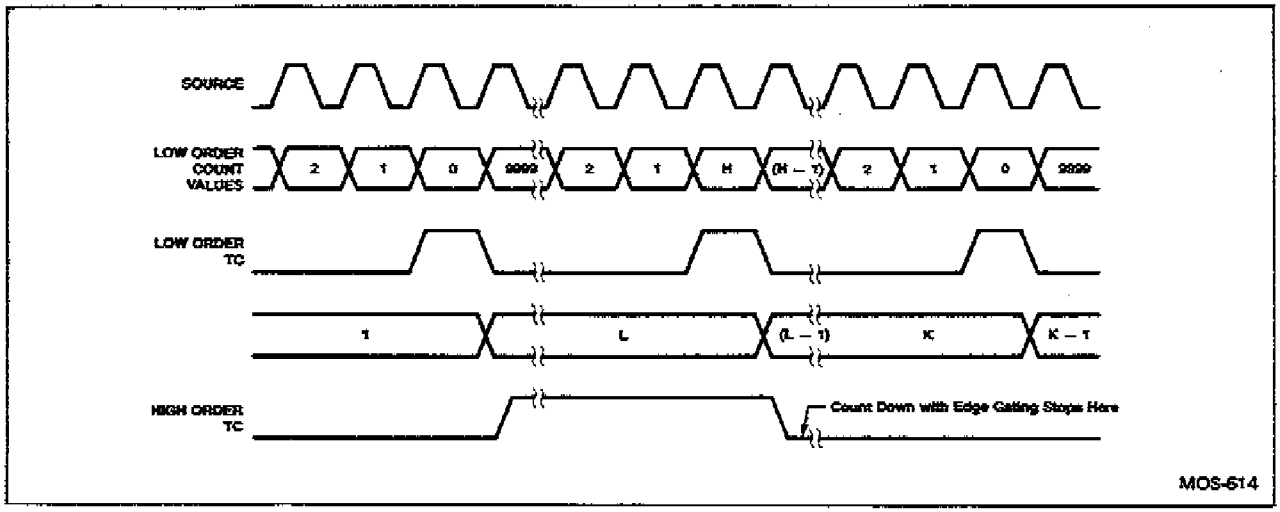

Figure 3-7. Timing Waveforms for Am9513 Count Down Concatenation

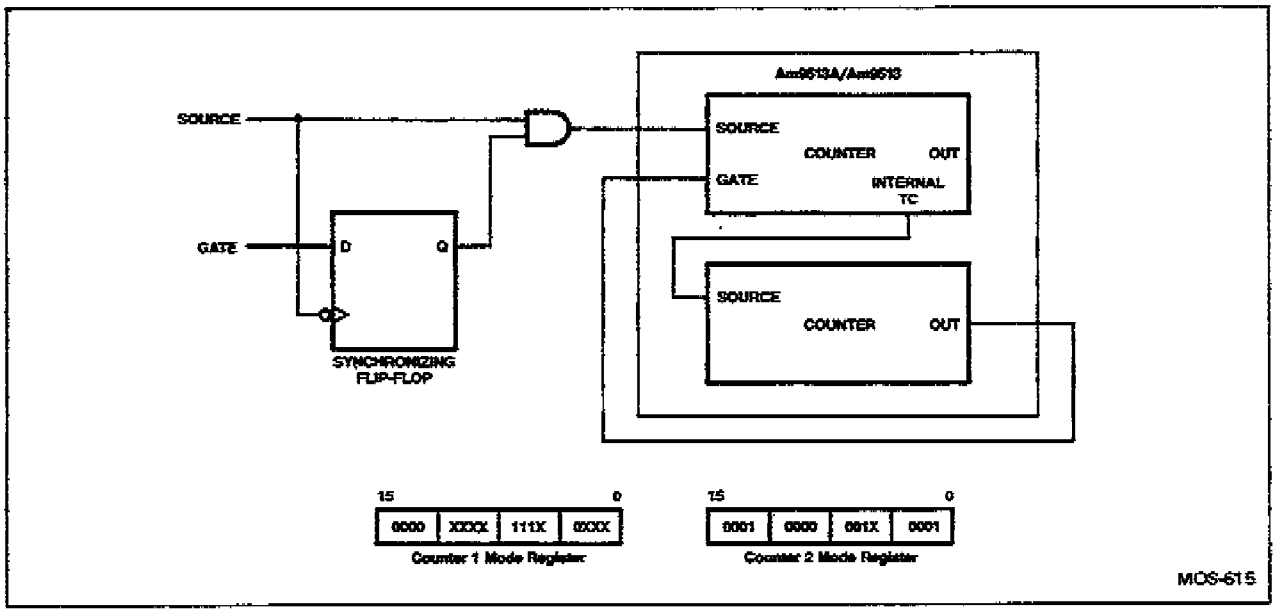

Figure 3-8. Count Down Concatenation with Level Gating

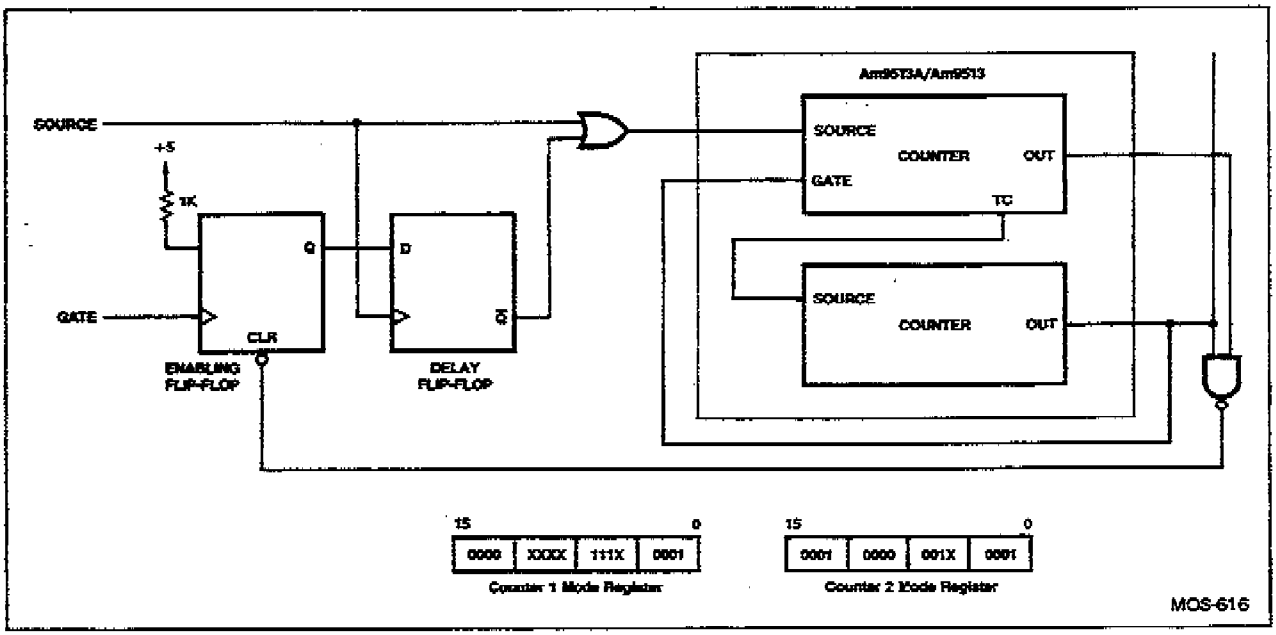

Figure 3-9. Count Down Concatenation with Edge Gating

since the timeout duration remains constant (at (L−1), H for Figure 3-7 and 3926 4178 for the earlier example). To ensure that the counters' first timing cycle has the same timeout duration as subsequent timing cycles, it is important that the high and low order counters be initialized to (L−1) and (H−1) respectively prior to the first timing cycle. Note that if the Counter 1 source period is less than the propagation delay from Counter 1's source through Counter 2's output, through the two flip-flops to the OR gate, then the low order counter's contents at the end of a count cycle may be offset by a few counts. In such cases, the value used to initialize the counters should be similarly offset.

The previous count down concatenation examples have assumed the counters are to count repetitively. To add count once capability to a count down configuration, the high order counter should be programmed to generate a TC Toggled output waveform. This output should be used to gate source pulses through an AND gate into the low order counter, as shown in Figure 3-10. The count cycle will now appear as shown in Figure 3-11. Note that when the counters stop, the high order counter's output will be low and the low order counter's contents will be 9999. To reset the counters for another timing cycle, a LOAD command should be issued to the low order counter, which will

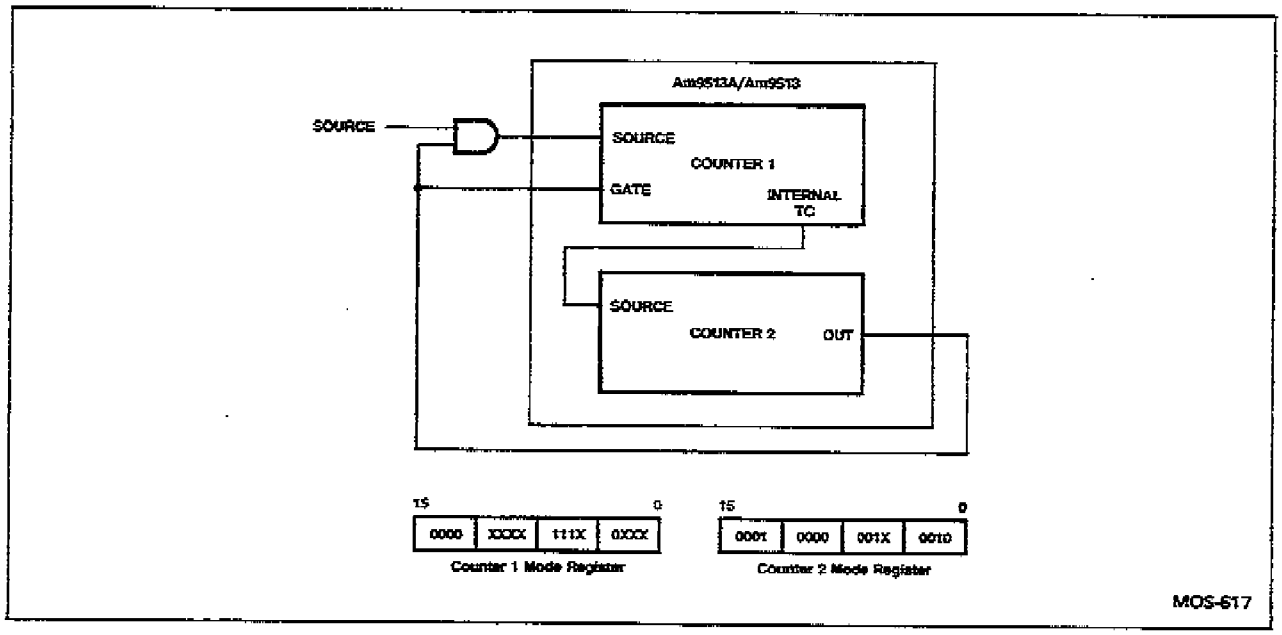

Figure 3-10. Count Down Concatenation with Count Once Feature

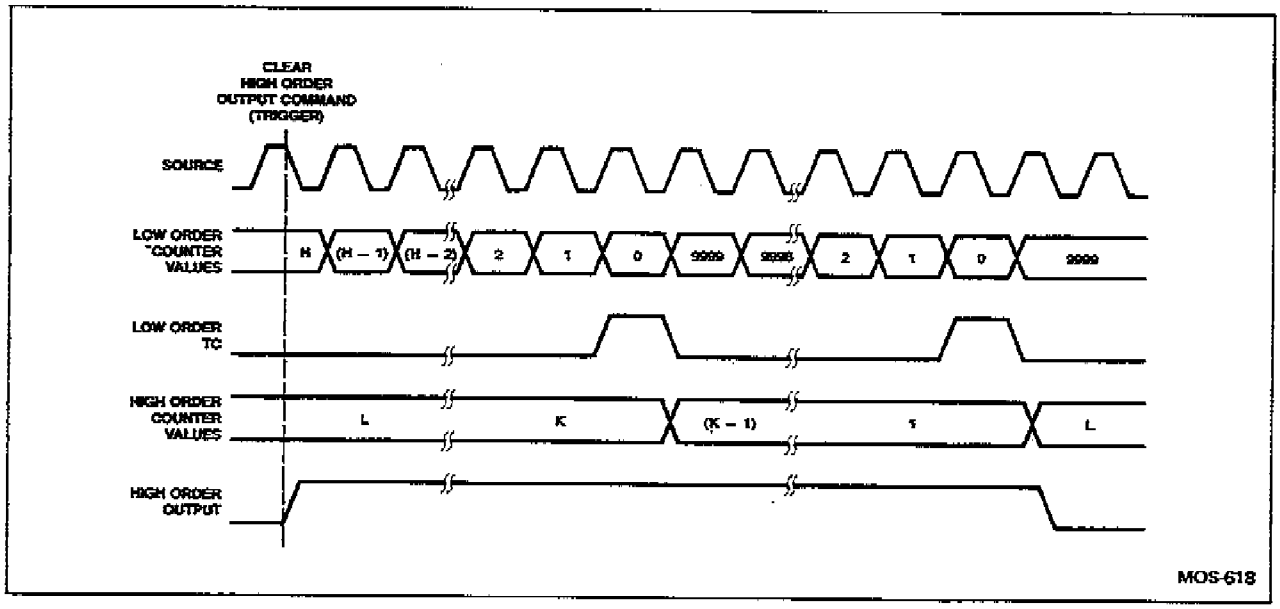

Figure 3-11. Timing Waveforms for Count Down Concatenation with Count Once Feature

reload from the Hold register. The output of the high order counter can now be set to enable counting. Level gating can be added to count-once, count down concatenation by using a 3-input AND gate and driving the third input with an external level gate signal. Count-once, count down concatenation with edge gating can be achieved with the circuit shown in Figure 3-12. The flip-flop is set by an external synchronous gate edge; it is cleared at the end of the count cycle when Counter 2's TC Toggled output goes low.

The concatenation examples presented so far have used two counters to create a 32-bit effective count length. These configurations can be extrapolated to concatenate 3 or more counters to any desired length. Other concatenation variations adventuresome users may wish to investigate are those that use the Alarm registers on Counter 1 and 2 to generate unusual count sequences. Since these Alarm register configurations usually add much complexity for only a limited increase in functionality they are not discussed in this manual.

Saving Concatenated Count Values

The contents of concatenated counters may be read by issuing a SAVE command to the appropriate counters, which will transfer the current counter contents into the counter's Hold registers.

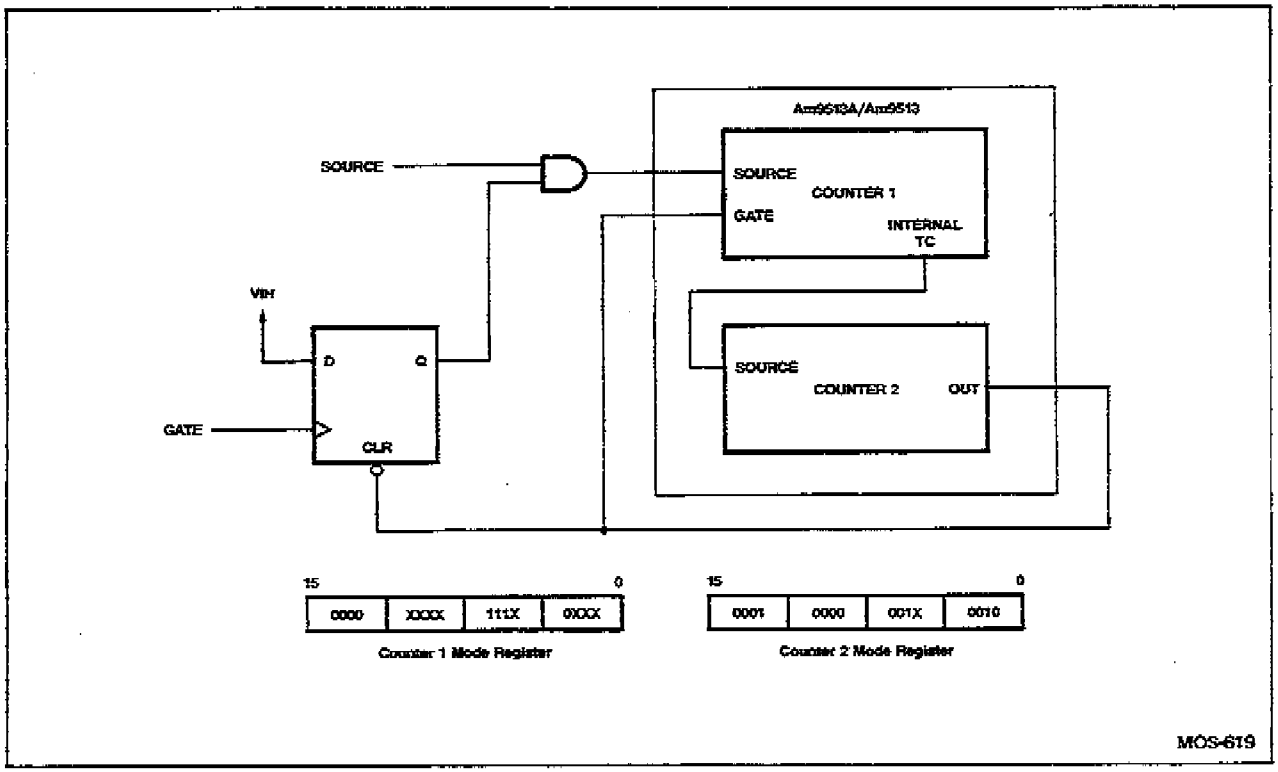

Figure 3-12. Count Down Concatenation with Edge Gating and Count Once Feature

(Since in count down concatenation the Hold register is used to generate the count sequence, in many such applications it may not be feasible to save the low-order counter.) Because the count ripples between concatenated counters, the possibility exists that a SAVE command will be issued after the low order counter increments/decrements but before the carry/borrow ripples through to the high order counter, resulting in an incorrect value being saved in the high order counter's Hold register. The user can protect against this by examining the contents of the low order counter's Hold register immediately after issuing the SAVE command. If the Hold register is equal to the value that would have been expected immediately following generation of a carry/borrow signal, this indicates that the high order value saved is suspect. A new SAVE command should therefore be issued to the high order counter to save a correct count. By the time the low order Hold register contents are read and tested, and a new SAVE command is issued, the high order counter's contents will be stable. The "Time-of-Day" chapter discusses these considerations with respect to Time-of-Day accumulation, and includes a representative software listing.