Rockwell

R6500/R6532
Timer Interrupt Precautions

PURPOSE

The R6532 is a RAM, I/O, timer (RIOT) combination device. The timer is an on-board count-down circuit and may be programmed to cause interrupts to the R6502 microprocessor. The timer interrupt is enabled by simply addressing the R6532 with the necessary selects and address bits. It is *not* necessary to load any internal registers to enable the interrupt, only to address the device.

The R6502 (or one of its 28-pin versions) microprocessor can inadvertently cause addressing of the R6532 (or other devices with a timer) during start-up or RESET operations. When $\overline{RES}$ is driven low, the R6502 and R6532 are initialized to known internal states. When $\overline{RES}$ goes high, however, the start-up procedure is initiated and the first two cycles contain arbitrary or random addresses on the bus. Table 1 illustrates the cycles immediately following $\overline{RES}$ going high.

In most cases, this is normally not a problem, since the occurrence of $\overline{RES}$ causes the R6502 $\overline{IRQ}$ interrupt to be disabled. However, if the R6532 interrupt output is connected to the $\overline{NMI}$ (Non-Maskable Interrupt) input of the R6502 and the R6532 timer interrupt is inadvertently enabled during start-up, a timer interrupt may occur before the processor has executed its system initialization procedure. The possibility of all these things happening is somewhat remote, but can be a potential problem.

DESCRIPTION

There are several solutions to this problem:

1. Use the $\overline{IRQ}$ interrupt input of the R6502, instead of $\overline{NMI}$. In this way, the $\overline{IRQ}$ interrupt is automatically disabled by $\overline{RES}$ and the R6532 timer interrupt cannot occur.

2. Use separate $\overline{RES}$ signals for the R6502 and the R6532. In order to avoid the first two cycles of the start-up (wherein the addresses are unpredictable), it is necessary to hold the $\overline{RES}$ to the 6532 low *after* the $\overline{RES}$ to the R6502 goes high. This is illustrated in Figure 1.

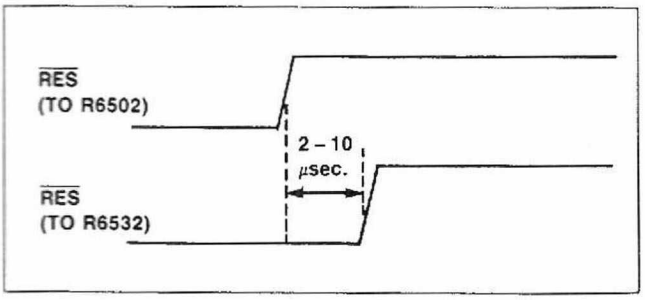

Figure 1. Delayed Reset to R6532

Table 1. Start-Up Cycle

Cycles	Address Bus	Data Bus	External Operation	Internal Operation
1	?	?	Don't Care	Hold during Reset
2	? + 1	?	Don't Care	First Start State
3	0100 + SP	?	Don't Care	Second Start State
4	0100 + SP – 1	?	Don't Care	Third Start State
5	0100 + SP – 2	?	Don't Care	Fourth Start State
6	FFFC	Start PCL	Fetch First Vector	
7	FFFD	Start PCH	Fetch Second Vector	Hold PCL
8	PCH PCL	First OP CODE	Load First OP CODE	

3. Gate the R6532 $\overline{IRQ}$ output to the R6502 $\overline{NMI}$ input with a circuit which is enabled by the initialization routine of the processor. Figure 2 shows a possible configuration, with the gate enabled only when the processor does a write operation with A15 high. Note that this scheme essentially allows disabling of the $\overline{NMI}$ in the R6502.

There are likely to be many other solutions to this problem. These ideas are intended to provide some simple ones and to provoke thought for others from the reader.

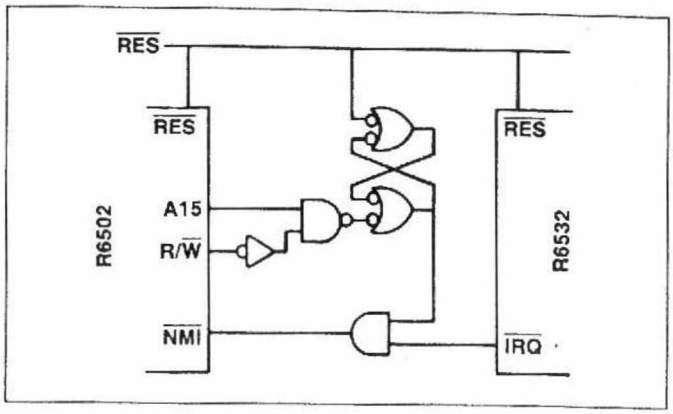

Figure 2. Scheme for Gating to the NMI Input