This document covers all programming information about Rockwell R6522 Versatile Interface Adapter (VIA) chip used in many Commodore devices. Figures and tables in this file were drawn with IBM (fixed width) font. Then, for correct view in Graphical Browser or Editor you must first install some LineDraw True-Type Font. In text mode you will need switch to original Video-BIOS character set.

This file was created from scannings by Frank Kontros <jeno@kontr.uzhgorod.ua>
Many thanks for corrections to Wolfgang Lorenz <pc64@compuserve.com>

If you have any questions, comments or suggestions concerning this file or 6522 chip, please contact me (Frank).

R6522
VERSATILE INTERFACE
ADAPTER

DESCRIPTION

The R6522 Versatile Interface Adapter (VIA) is a very flexible I/O control device. In addition, this device contains a pair of very powerful 16-bit interval timers, a serial-to-parallel/parallel-to-serial shift register and input data latching on the peripheral ports. Expanded handshaking capability allows control of bidirectional data transfers between VIA's in multiple processor systems.

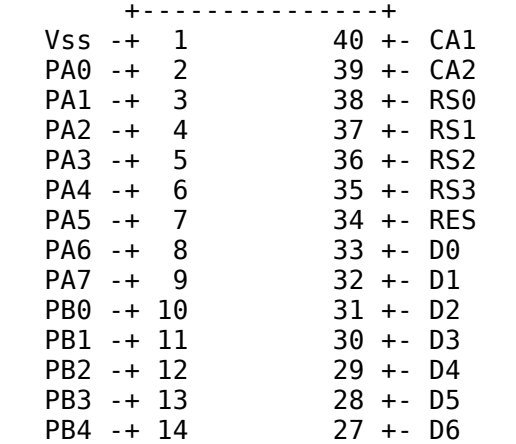
Control of peripheral devices is handled primarily through two 8-bit bidirectional ports. Each line can be programmed as either an input or an output. Several peripheral I/O lines can be controlled directly from the interval timers for generating programmable frequency square waves or for counting externally generated pulses. To facilitate control of the many powerful features of this chip, an interrupt flag register, an interrupt enable register and a pair of function control registers are provided.

FEATURES

- Two 8-bit bidirectional I/O ports
- Two 16-bit programmable timer/counters
- Serial data port
- TTL compatible
- CMOS compatible peripheral control lines
- Expanded "handshake" capability allows positive control data transfers between processor and peripheral devices
- Latched output and input registers
- 1 MHz and 2 Mhz operation
- Single +5V power supply

ORDERING INFORMATION

```
 Part Number:
 R6522 _ _ _
       | | |
       | | |
       | | |
       | | |
       | | |
       | | +------ Temperature Range
       | |         Blank = 40°C to +70°C
       | |             E = 40°C to +85°C
       | |
       | |
```



```
        | +-------- Package                    PB5 -+ 15         26 +- D7
        |              C = Ceramic             PB6 -+ 16         25 +- 02
        |              P = Plastic             PB7 -+ 17         24 +- CS1
        |                                      CB1 -+ 18         23 +- CS2
        |                                      CB2 -+ 19         22 +- R/W
        +---------- Frequency                  Vcc -+ 20         21 +- IRQ
                    No Letter = 1 MHz                +---------------+
                            A = 2 MHz              R6522 Pin Configuration
```

INTERFACE SIGNALS

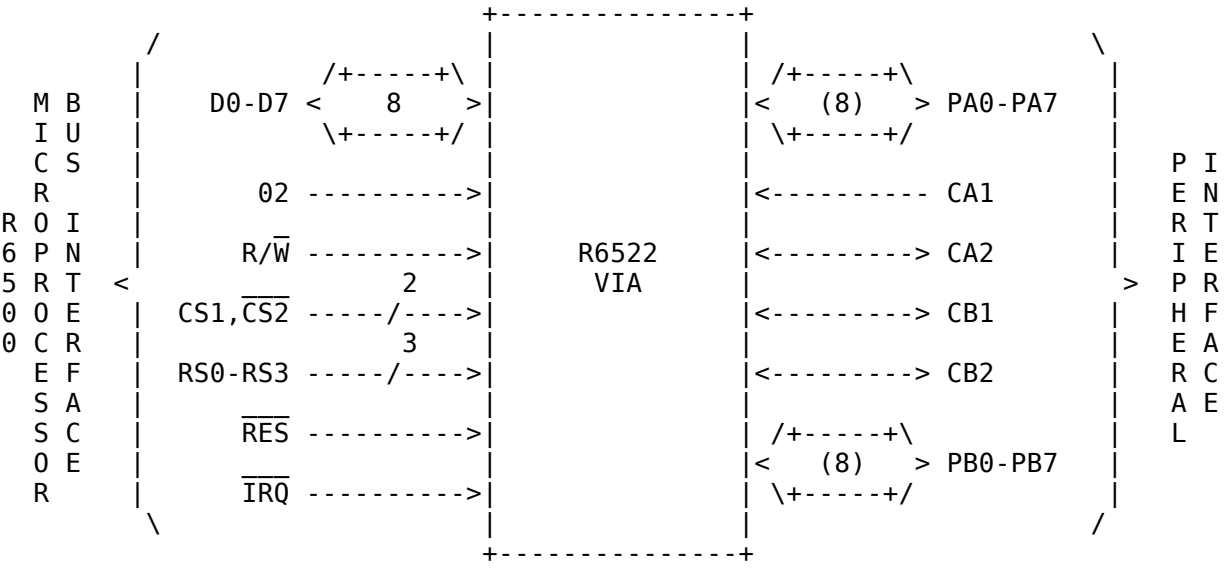


Figure 1. R6522 VIA interface Signals

RESET ($\overline{RES}$)

A low reset ($\overline{RES}$) input clears all R6522 internal registers to logic 0 (except T1 and T2 latches and counters and the Shift Register). This places all peripheral interface lines in the input state, disables the timers, shift register, etc. and disables interrupting from the chip.

INPUT CLOCK (PHASE 2)

The input clock is the system 02 clock and triggers all data transfers between processor bus and the R6522.

READ/WRITE (R/$\overline{W}$)

The direction of the data transfers between the R6522 and the system processor is controlled by the R/W line in conjunction with the CS1 and CS2 inputs. When R/W is low (write operation), and the R6522 is selected, data is transferred from the processor bus into the selected R6522 register. When R/W is high (read operation), and the R6522 is selected, data is transferred from the selected R6522 register to the processor bus.

DATA BUS (D0-D7)

The eight bidirectional data bus lines transfer data between the R6522 and the system processor bus. During read cycles, the contents of the selected R6522 register are placed on the data bus lines. During write cycles, these lines are high-impedance inputs and data is transferred from the processor bus into the selected register. When the R6522 is not selected, the data bus lines are high impedance.

CHIP SELECTS (CS1, $\overline{CS2}$)

The two chip select inputs are normally connected to processor address lines either directly or through decoding. The selected R6522 register is accessed

when CS1 is high and CS2 is low.

REGISTER SELECTS (RS0-RS3)

The coding of the four Register Select inputs select one of the 16 internal registers of the R6522, as shown in Table 1.

Table 1. R6522 Register Addressing

Reg. #	RS Coding RS3	RS2	RS1	RS0	Register Desig.	Register/Description Write (R/W = L)	Read (R/W = H)
0	0	0	0	0	ORB/IRB	Output Register B	Input Register B
1	0	0	0	1	ORA/IRA	Output Register A	Input Register A
2	0	0	1	0	DDRB	Data Direction Register B	
3	0	0	1	1	DDRA	Data Direction Register A	
4	0	1	0	0	T1C-L	T1 Low-Order Latches	T1 Low-Order Counter
5	0	1	0	1	T1C-H	T1 High-Order Counter	T1 High-Order Counter
6	0	1	1	0	T1L-L	T1 Low-Order Latches	
7	0	1	1	1	T1L-H	T1 High-Order Latches	
8	1	0	0	0	T2C-L	T2 Low-Order Latches	T2 Low-Order Counter
9	1	0	0	1	T2C-H	T2 High-Order Counter	
10	1	0	1	0	SR	Shift Register	
11	1	0	1	1	ACR	Auxiliary Control Register	
12	1	1	0	0	PCR	Peripheral Control Register	
13	1	1	0	1	IFR	Interrupt Flag Register	
14	1	1	1	0	IER	Interrupt Enable Register	
15	1	1	1	1	ORA/IRA	Output Register A°	Input Register A°

NOTE: °Same as Register 1 except no handshake.

INTERRUPT REQUEST ($\overline{IRQ}$)

The Interrupt Request output goes low whenever an internal interrupt flag is set and the corresponding interrupt enable bit is a logic 1. This output is open-drain to allow the interrupt request signal to be wire-OR'ed with other equivalent signals in the system.

PERIPHERAL PORT A (PA0-PA7)

Port A consists of eight lines which can be individually programmed to act as inputs or outputs under control of a Data Direction Register A. The polarity of output pins is controlled by an Output Register and input data may be latched into an internal register under control of the CA1 line. All of these modes of operation are controlled by the system processor through the internal control registers. These lines represents one standard TTL load in the input mode and will drive one standard TTL load in the output mode. Figure 2 illustrates the output circuit.

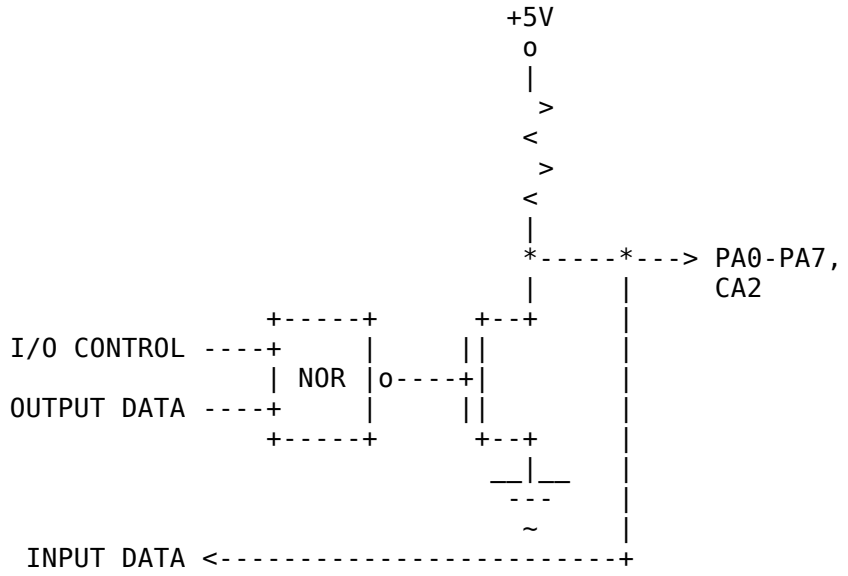


Figure 2. Port A Output Circuit

PORT A CONTROL LINES (CA1,CA2)

The two Port A control lines act as interrupt inputs or as handshake outputs. Each line controls an internal interrupt flag with a corresponding interrupt enable bit. In addition, CA1 controls the latching of data on Port A input lines. CA1 is a high-impedance input only while CA2 represents one standard TTL load in the input mode. CA2 will drive one standard TTL load in the output mode.

PORT B (PB0-PB7)

Peripheral Port B consists of eight bi-directional lines which are controlled by an output register and a data direction register in much the same manner as the Port A. In addition, the polarity of the PB7 output signal can be controlled by one of the interval timers while the second timer can be programmed to count pulses on the PB6 pin. Port B lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. In addition, they are capable of sourcing 1.0 mA at 1.5 Vdc in the output mode to allow the outputs to directly drive Darlington transistor circuits. Figure 3 is the circuit schematic.

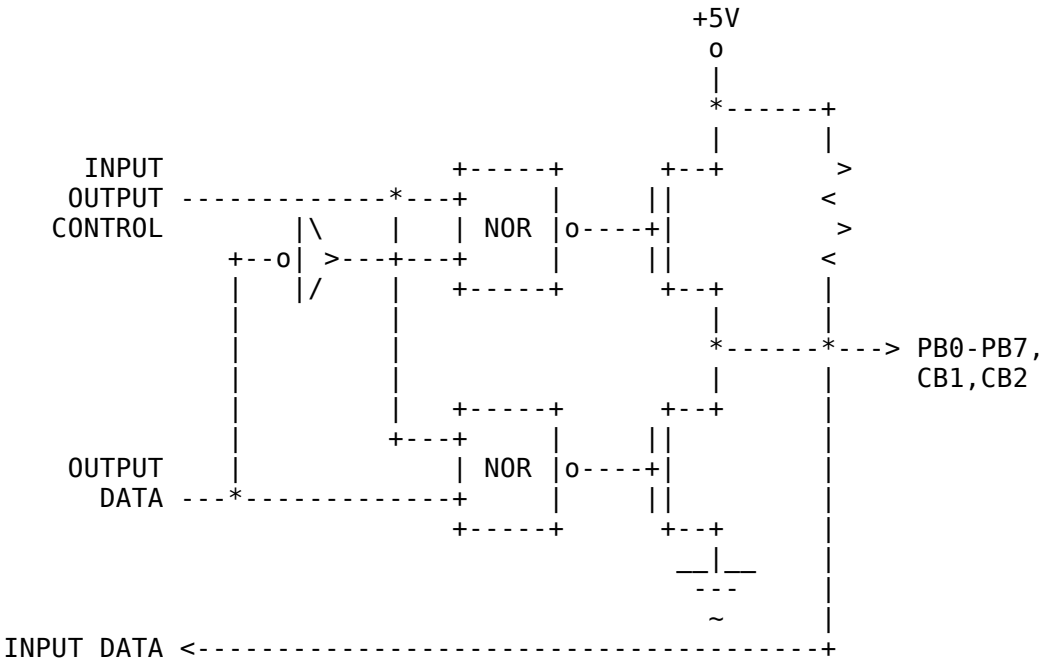


Figure 3. Port B Output Circuit

PORT B CONTROL LINES (CB1,CB2)

The Port B control lines act as interrupt inputs or as handshake outputs. As with CA1 and CA2, each line controls an interrupt flag with a corresponding interrupt enable bit. In addition, these lines act as a serial port under control of the Shift Register. These lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. CB2 can also drive a Darlington transistor circuit; however, CB1 cannot.

FUNCTIONAL DESCRIPTION

The internal organization of the R6522 VIA is illustrated in Figure 4.

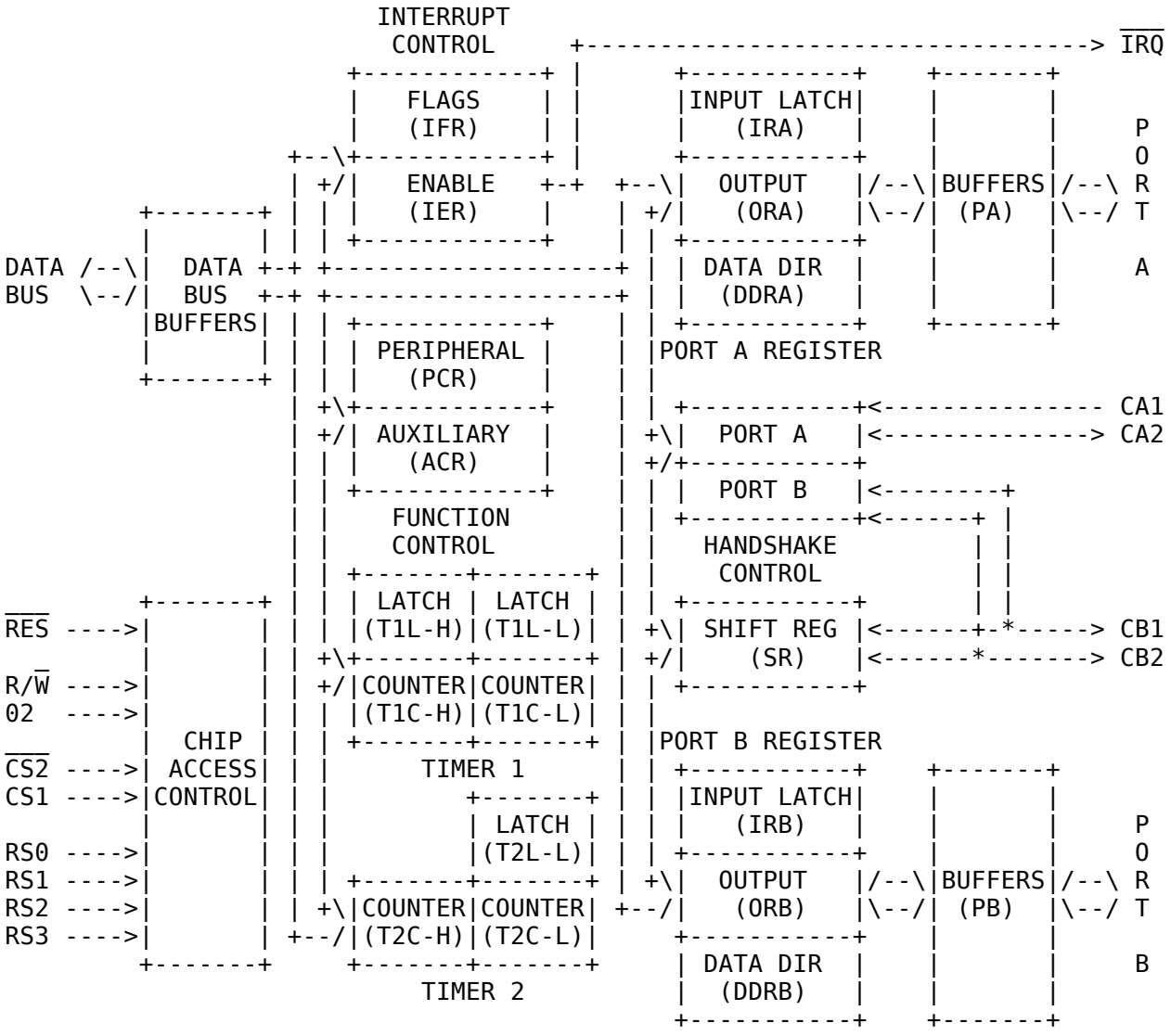


Figure 4. R6522 VIA Block Diagram

PORT A AND PORT B OPERATION

The R6522 VIA has two 8-bit bidirectional I/O ports (Port A and Port B) and each port has two associated control lines.

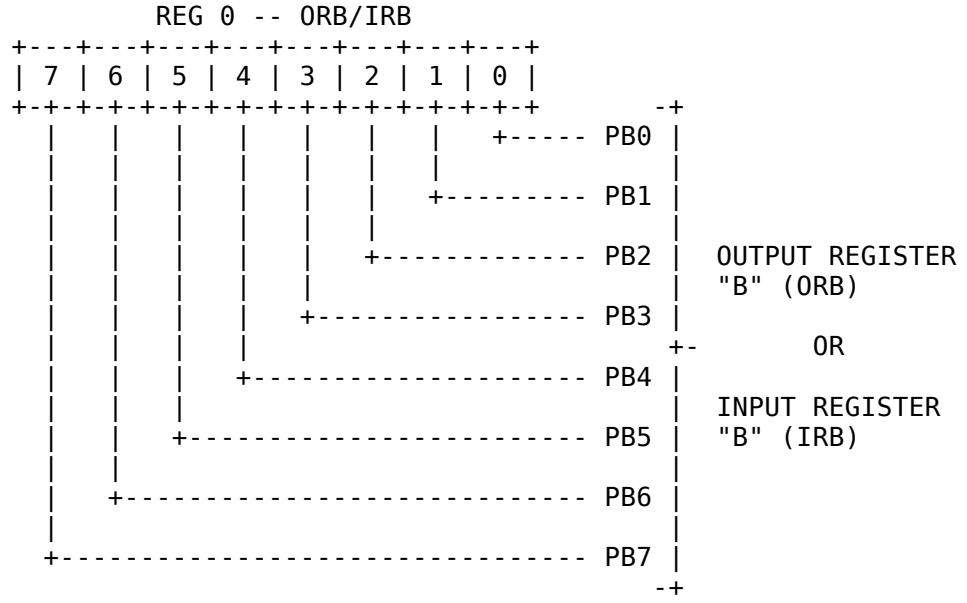
Each 8-bit peripheral port has a Data Direction Register (DDRA, DDRB) for specifying whether the peripheral pins are to act as inputs or outputs. A 0 in a bit of the Data Direction Register causes the corresponding peripheral pin to act as an input. A 1 causes the pin to act as an output.

Each peripheral pin is also controlled by a bit in the Output Register (ORA, ORB) and the Input Register (IRA, IRB). When the pin is programmed as an output, the voltage on the pin is controlled by the corresponding bit of the Output Register. A 1 in the Output Register causes the output to go high, and a 0 causes the output to go low. Data may be written into Output Register bits corresponding to pins which are programmed as inputs. In this case, however, the output signal is unaffected.

Reading a peripheral port causes the contents of the Input Register (IRA, IRB) to be transferred onto the Data Bus. With input latching disabled, IRA will always reflect the levels on the PA pins. With input latching enabled, IRA will reflect the levels on the PA pins at time the latching occurred (via CA1).

The IRB register operates similar to the IRA register. However, for pins programmed as outputs there is a difference. When reading IRA, the level on the pin determines whether a 0 or a 1 is sensed. When reading IRB, however, the bit stored in the output register, ORB, is the bit sensed. Thus, for outputs which have large loading effects and which pull an output "1" down or which pull an output "0" up, reading IRA may result in reading a "0" when a "1" was actually programmed, and reading a "1" when a "0" was programmed. Reading IRB, on the other hand, will read the "1" or "0" level actually programmed, no matter what the loading on the pin.

Figures 5 through 8 illustrate the formats of the port registers. In addition, the input latching modes are selected by the Auxiliary Control Register (Figure 12).



PIN DATA DIRECTION SELECTION	WRITE	READ
DDRB = 1 OUTPUT	MPU WRITES OUTPUT LEVEL ORB	MPU READS OUTPUT REGISTER BIT, ORB PIN LEVEL HAS NO AFFECT
DDRB = 0 INPUT INPUT LATCHING DISABLED	MPU WRITES INTO ORB BUT NO AFFECT ON PIN LEVEL UNTIL DDRB CHANGED	MPU READS INPUT LEVEL ON PB PIN
DDRB = 0 INPUT INPUT LATCHING ENABLED		MPU READS IRB BIT WHICH IS THE LEVEL OF THE PB PIN AT THE TIME OF THE LAST CB1 ACTIVE TRANSITION

Figure 5. Output Register B (ORB), Input Register B (IRB)

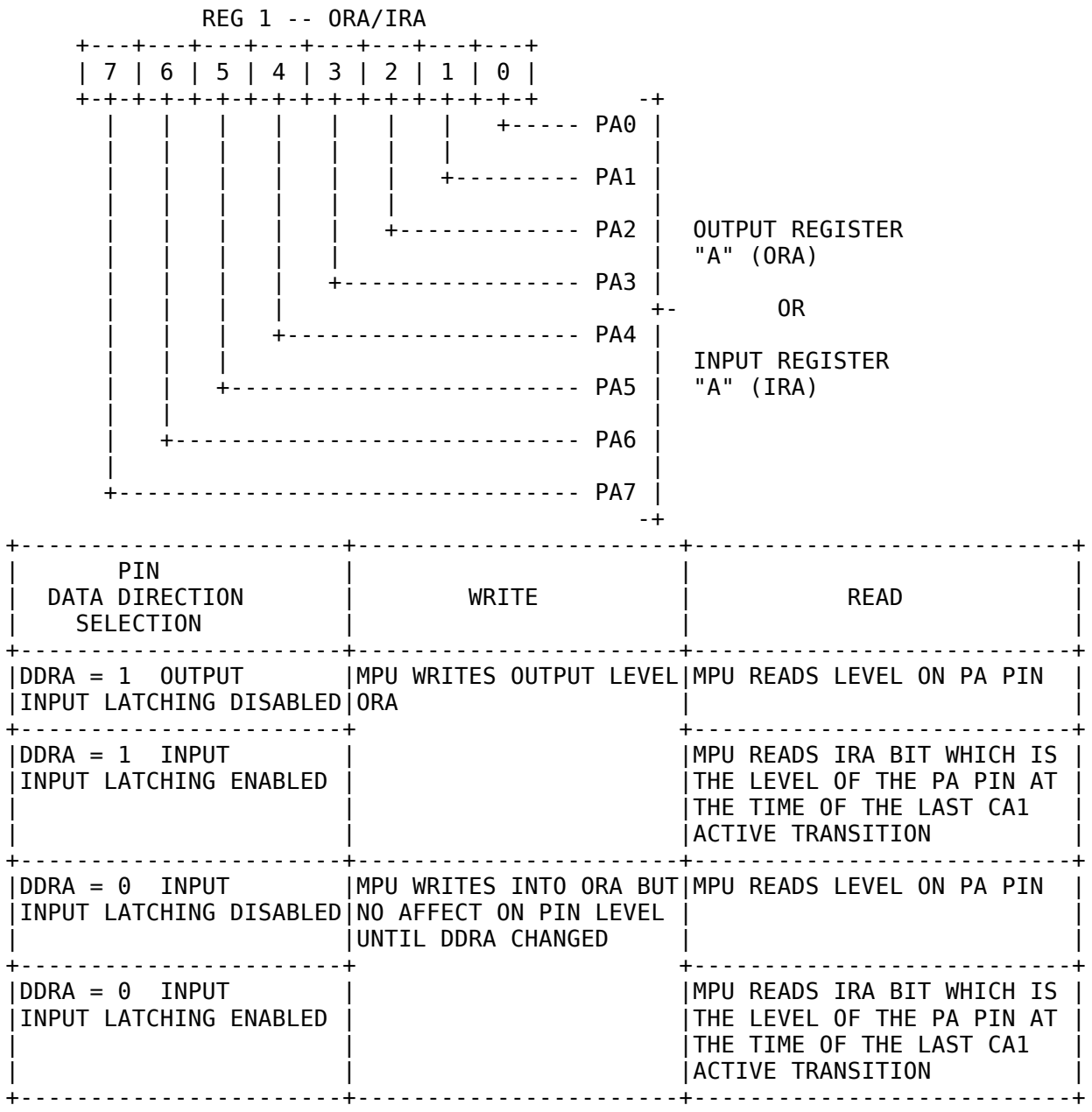


Figure 6. Output Register A (ORA), Input Register A (IRA)

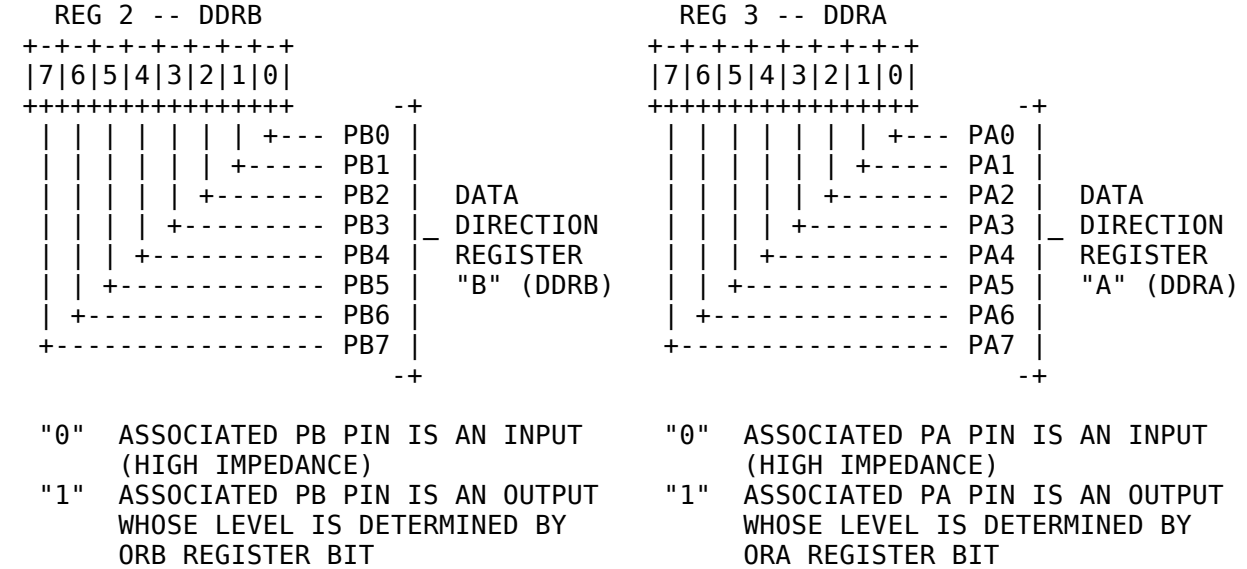


Figure 7. Data Direction Register B (DDRB)

Figure 8. Data Direction Register A (DDRA)

HANDSHAKE CONTROL OF DATA TRANSFERS

The R6522 allows positive control of data transfers between the system processor and peripheral devices through the operation of "handshake" lines. Port A lines (CA1, CA2) handshake data on both a read and a write operation while the Port B lines (CB1, CB2) handshake on a write operation only.

READ HANDSHAKE

Positive control of data transfers from peripheral devices into the system processor can be accomplished very effectively using Read Handshaking. In this case, the peripheral device must generate the equivalent of a "Data Ready" signal to the processor signifying that valid data is present on the peripheral port. This signal normally interrupts the processor, which then reads the data, causing generation of a "Data Taken" signal. The peripheral device responds by making new data available. This process continues until the data transfer is complete.

In the R6522, automatic "Read Handshaking" is possible on the Peripheral A Port only. The CA1 interrupt input pin accepts the "Data Ready" signal and CA2 generates the "Data Taken" signal. The "Data Ready" signal will set an internal flag which may interrupt the processor or which may be polled under program control. The "Data Taken" signal can either be a pulse or a level which is set low by the system processor and is cleared by the "Data Ready" signal. These options are shown in Figure 9 which illustrates the normal Read Handshake sequence.

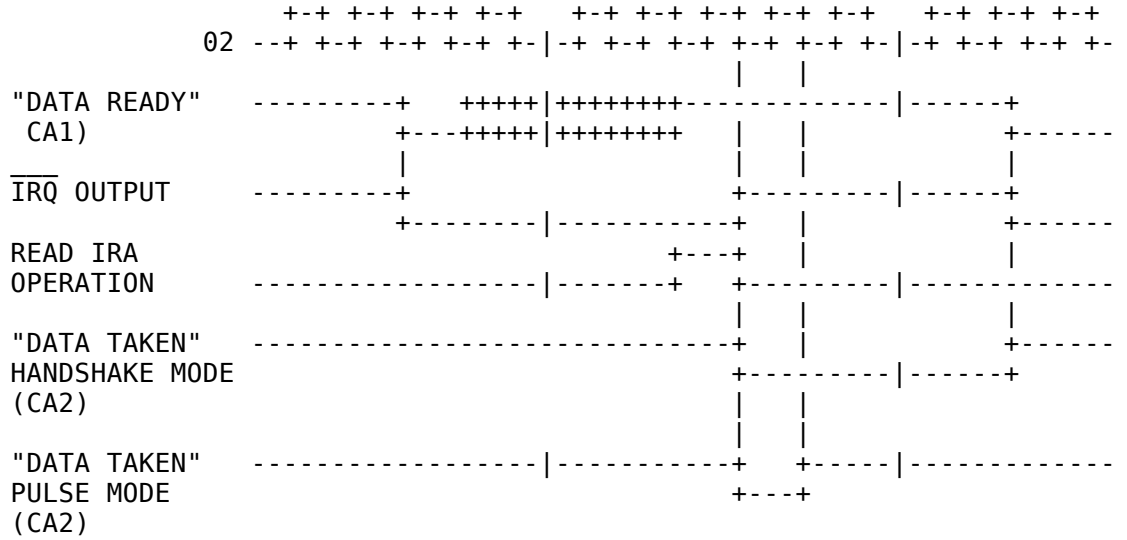


Figure 9. Read Handshake Timing (Port A, Only)

WRITE HANDSHAKE

The sequence of operations which allows handshaking data from the system processor to a peripheral device is very similar to that described for Read Handshaking. However, for Write Handshaking, the R6522 generates the "Data Ready" signal and the peripheral device must respond with the "Data Taken" signal. This can be accomplished on both the PA port and the PB port on the R6522. CA2 or CB2 act as a "Data Ready" output in either the handshake mode or pulse mode and CA1 or CB1 accept the "Data Taken" signal from the peripheral device, setting the interrupt flag and cleaning the "Data Ready" output. This sequence is shown in Figure 10.

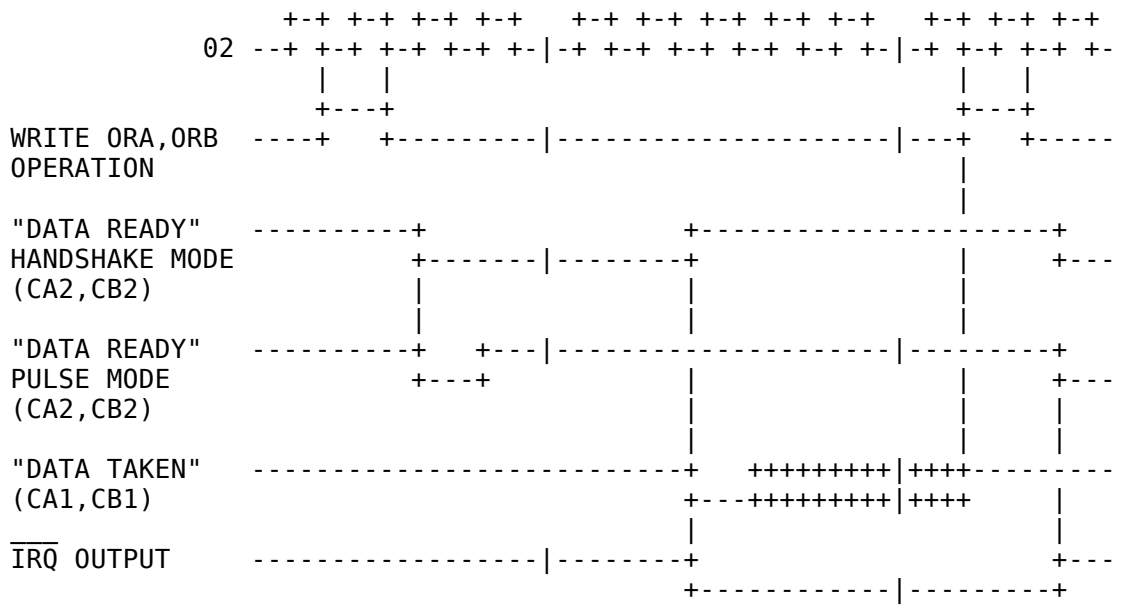


Figure 10. Write Handshake Timing

Selection of operating modes for CA1, CA2, CB1 and CB2 is accomplished by the Peripheral Control Register (Figure 11).

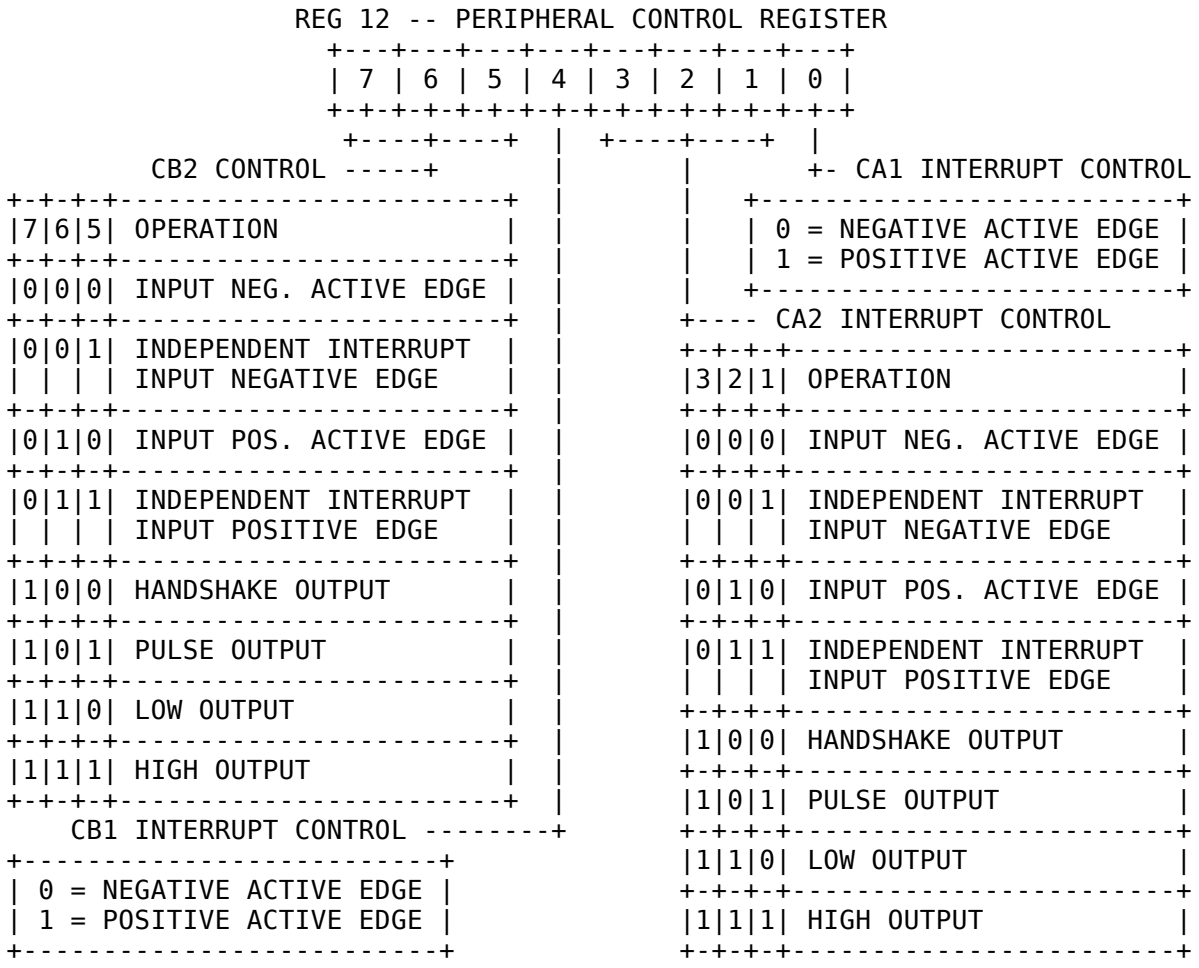


Figure 11. Peripheral Control Register (PCR)

COUNTERS/TIMERS

There are two independent 16-bit-counter/timers (called Timer 1 and Timer 2) in the R6522. Each timer is controlled by writing bits into the Auxiliary Control Register (ACR) to select this mode of operation (Figure 12).

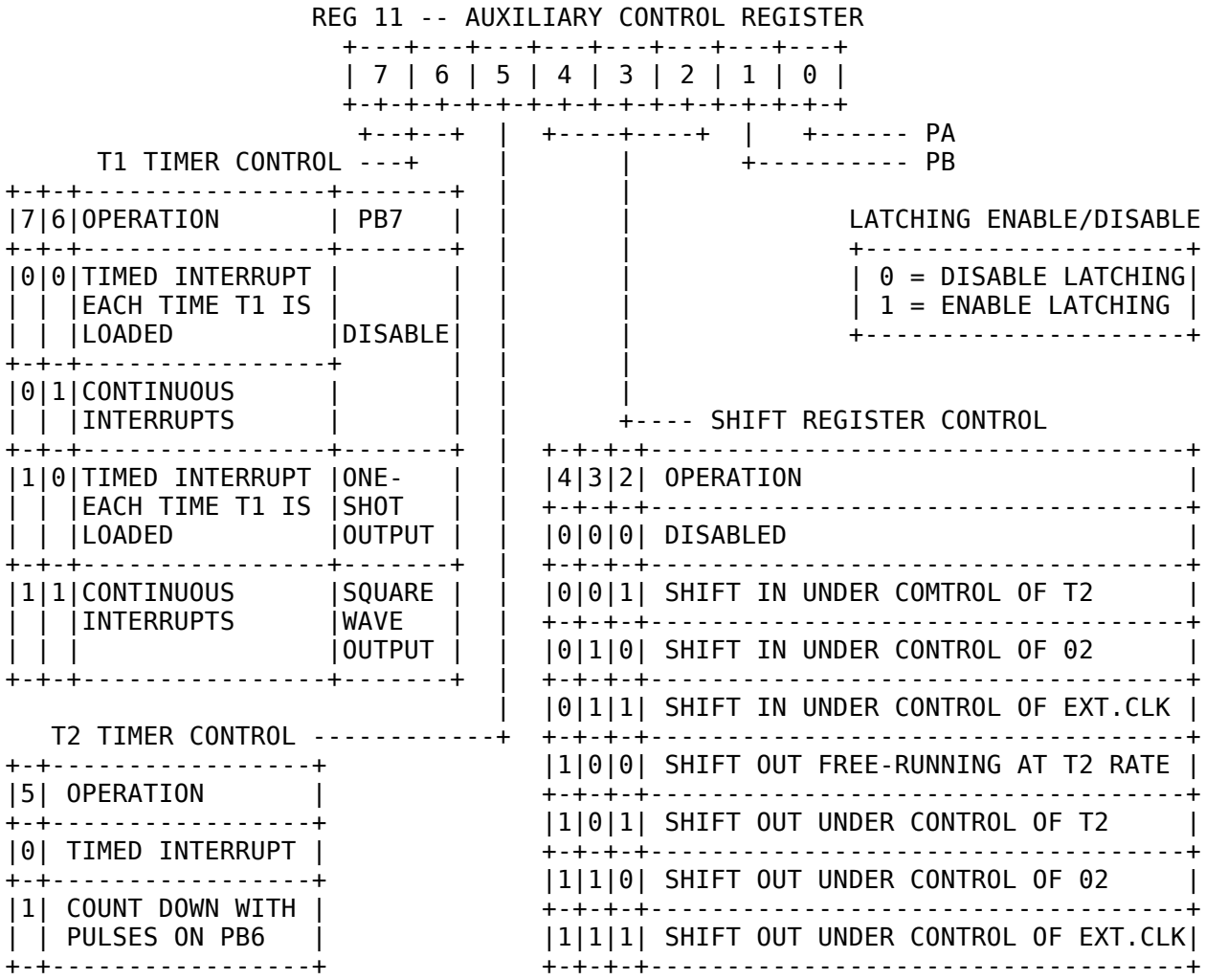
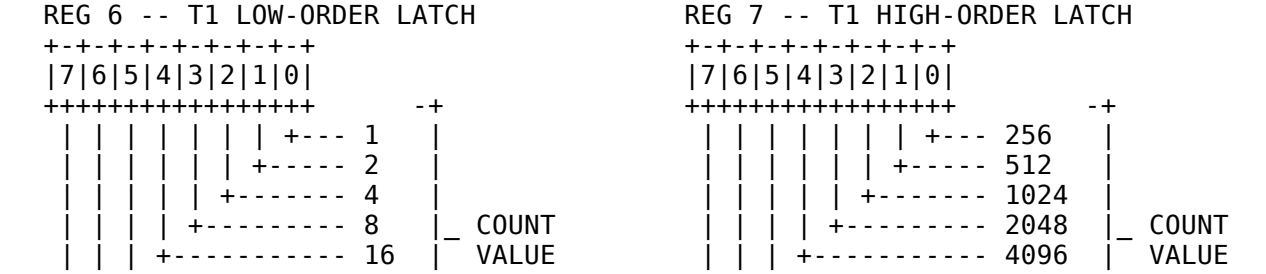


Figure 12. Auxiliary Control Register (ACR)

Timer 1 Operation

Interval Timer T1 consists of two 8-bit latches (Figure 13) and a 16-bit counter (Figure 14). The latches store data which is to be loaded into the counter. After loading, the counter decrements at 02 clock rate. Upon reaching zero, an interrupt flag is set, and IRQ goes low if the T1 interrupt is enabled. Timer 1 then disables any further interrupts or automatically transfers the contents of the latches into the counter and continues to decrement. In addition, the timer may be programmed to invert the output signal on a peripheral pin (PB7) each time it "times-out". Each of these modes is discussed separately below.

Note that the processor does not write directly into the low-order counter (T1C-L). Instead, this half of the counter is loaded automatically from the low order latch (T1L-L) when the processor writes into the high order counter (T1C-H). In fact, it may not be necessary to write to the low order counter in some applications since the timing operation is triggered by writing to the high order latch.



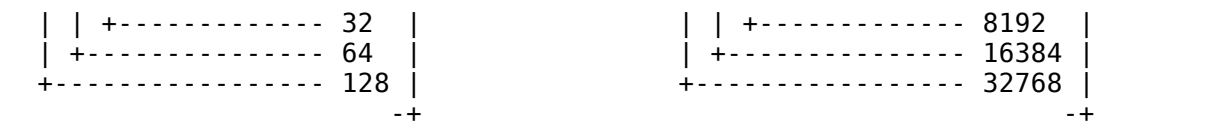


Figure 13. Timer 1 (T1) Latch Registers

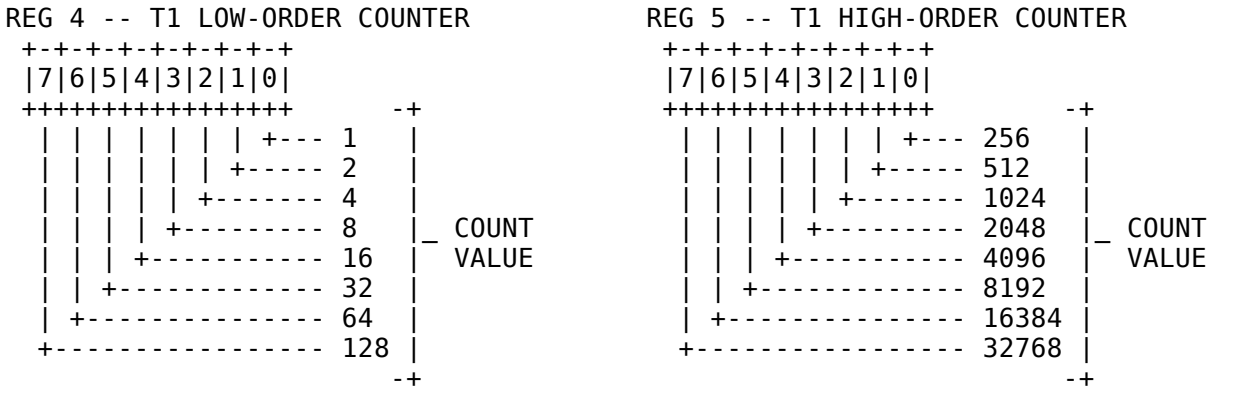


Figure 14. Timer 1 (T1) Counter Registers

Timer 1 One-Shot Mode

The Timer 1 one-shot mode generates a single interrupt for each timer load operation. As with any interval timer, the delay between the "write T1C-H" operation and generation of the processor interrupt is a direct function of the data loaded into the timing counter. In addition to generating a single interrupt, Timer 1 can be programmed to produce a single negative pulse on the PB7 peripheral pin. With the output enabled (ACR7=1) a "write T1C-H" operation will cause PB7 to go low. PB7 will return high when Timer 1 times out. The result is a single programmable width pulse.

T1 interrupt flag will be set, the $\overline{IRQ}$ pin will go low (interrupt enabled), and the signal on PB7 will go high. At this time the counter will continue to decrement at system clock rate. This allows the system processor to read the contents of the counter to determine the time since interrupt. However, the T1 interrupt flag cannot be set again unless it has been cleared as described in this specification.

Timing for the R6522 interval timer one-shot mode is shown in Figure 15.

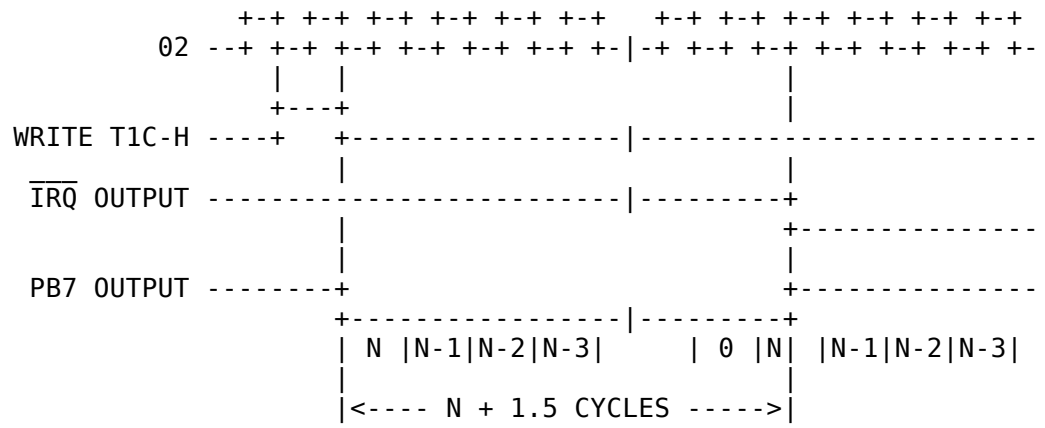


Figure 15. Timer 1 One-Shot Mode Timing

In the one-shot mode, writing into the T1L-H has no effect on the operation of Timer 1. However, it will be necessary to assure that the low order latch contains the proper data before initiating the count-down with a "write T1C-H" operation. When the processor writes into the high order counter (T1C-H), the T1 interrupt flag will be cleared, the contents of the low order latch will be transferred into the low order counter, and the timer will begin to decrement at system clock rate. If the PB7 output is enabled, this signal will go low on the 02 following the write operation. When the counter reaches zero, the T1 interrupt flag will be set, the IRQ pin will go low (interrupt enabled), and the signal on PB7 will go high. At this time the counter will continue to decrement at system clock rate. This allows the system processor to read the contents of the counter to determine the time since interrupt. However, the T1 interrupt flag cannot be set again unless it has been cleared as described in this specification.

Timer 1 Free-Run Mode

The most important advantage associated with the latches in T1 is the ability to produce a continuous series of evenly spaced interrupts and the ability to produce a square wave on PB7 whose frequency is not affected by variations in the processor interrupt response time. This is accomplished in the "free-running" mode.

In the free-running mode, the interrupt flag is set and the signal on PB7 is inverted each time the counter reaches zero. However, instead of continuing to decrement from zero after a time-out, the timer automatically transfers the contents of the latch into the counter (16 bits) and continues to decrement from there. The interrupt flag can be cleared by writing T1C-H, by reading T1C-L or by writing directly into the flag as described later. However, it is not necessary to rewrite the timer to enable setting the interrupt flag on the next time-out.

All interval timers in the R6522 are "re-triggerable". Rewriting the counter will always re-initialize the time-out period. In fact, the time-out can be prevented completely if the processor continues to rewrite the timer before it reaches zero. Timer 1 will operate in this manner if the processor writes into the high order counter (T1C-H). However, by loading the latches only, the processor can access the timer during each down-counting operation without affecting the time-out in process. Instead, the data loaded into the latches will determine the length of the next time-out period. This capability is particularly valuable in the free-running mode with the output enabled. In this mode, the signal on PB7 is inverted and the interrupt flag is set with each time-out. By responding to the interrupts with new data for the latches, the processor can determine the period of the next half cycle during each half cycle of the output signal on PB7. In this manner, very complex waveforms can be generated.

A precaution to take in the use of PB7 as the timer output concerns the Data Direction Register contents for PB7. Both DDRB bit 7 and ACR bit 7

must be 1 for PB7 to function as the timer output. If one is 1 and the other is 0, then PB7 functions as a normal output pin, controlled by ORB bit 7.

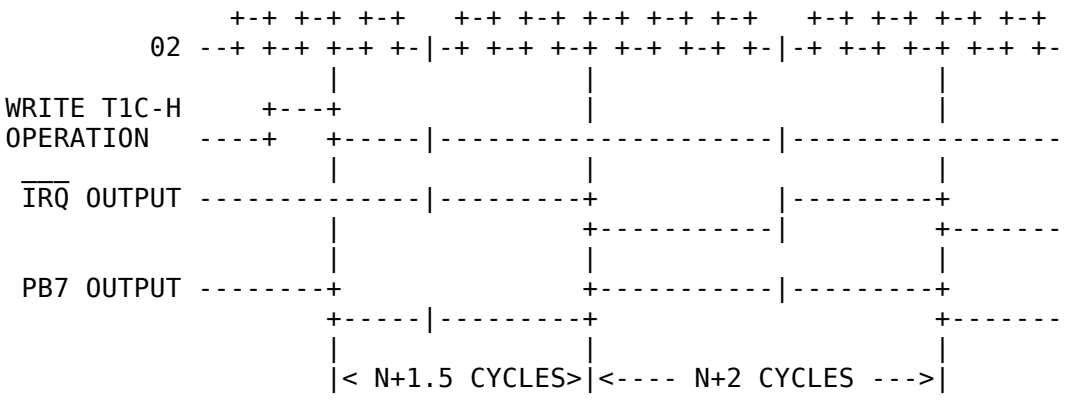


Figure 16. Timer 1 Free-Run Mode Timing

Timer 2 Operation

Timer 2 operates as an interval timer (in the "one-shot" mode only), or as a counter for counting negative pulses on the PB6 peripheral pin. A single control bit in the Auxiliary Control Register selects between these two modes. This timer is comprised of a "write-only" low-order latch (T2L-L), a "read-only" low-order counter (T2C-L) and a read/write high order counter (T2C-H). The counter registers act as a 16-bit counter which decrements at 02 rate. Figure 17 illustrates the T2 Latch/Counter Registers.

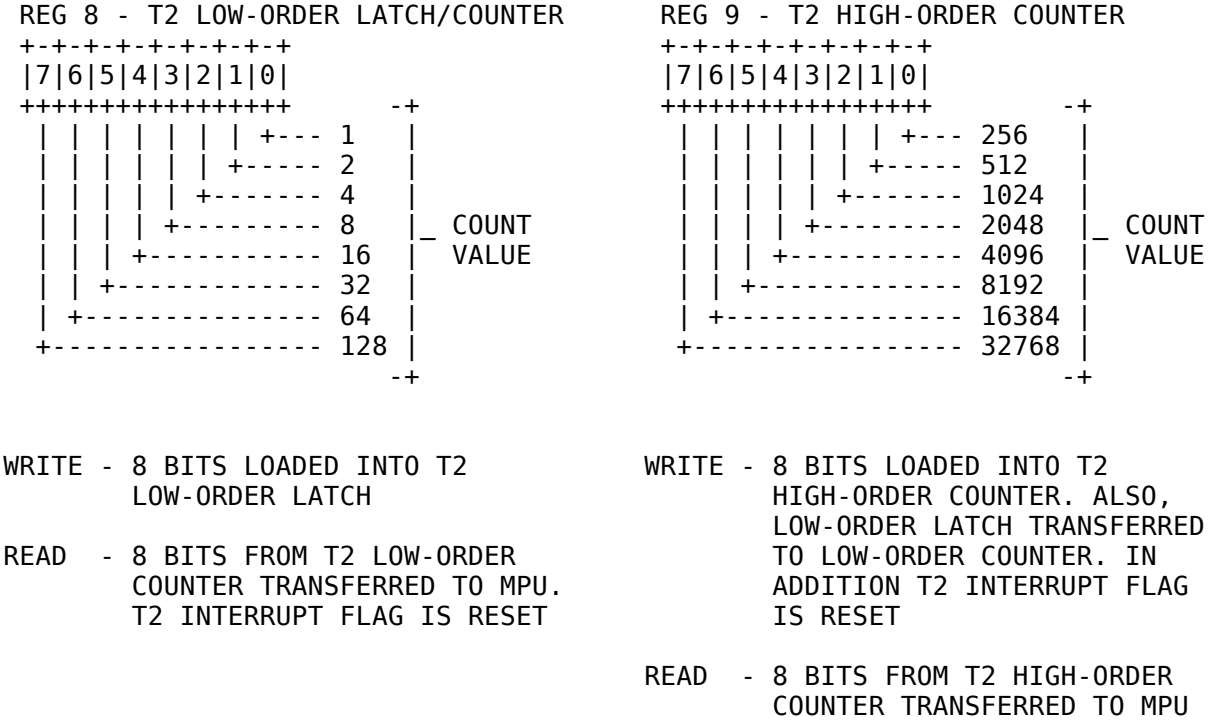


Figure 17. Timer 2 (T2) Latch/Counter Registers

Timer 2 One-Shot Mode

As an interval timer, T2 operates in the "one-shot" mode similar to Timer 1. In this mode, T2 provides a single interrupt for each "write T2C-H" operation. After timing out, the counter will continue to decrement. However, setting of the interrupt flag is disabled after initial time-out so that it will not be set by the counter decrementing again through zero. The processor must rewrite T2C-H to enable setting of the interrupt flag. The interrupt flag is cleared by reading T2C-L or by writing T2C-H. Timing

for this operation is shown in Figure 18.

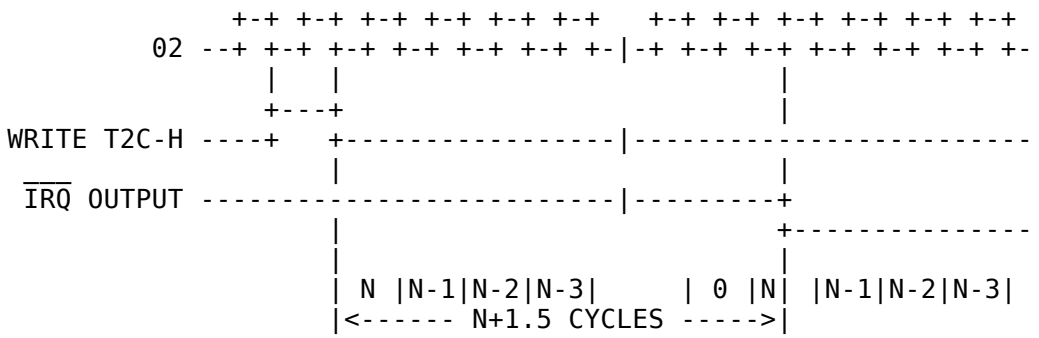


Figure 18. Timer 2 One-Shot Mode Timing

Timer 2 Pulse Counting Mode

In the pulse counting mode, T2 counts a predetermined number of negative-going pulses on PB6. This is accomplished by first loading a number into Timer 2. Writing into T2C-H clears the interrupt flag and allows the counter to decrement each time a pulse is applied to PB6. The interrupt flag is set when T2 counts down past zero. The counter will then continue to decrement with each pulse on PB6. However, it is necessary to rewrite T2C-H to allow the interrupt flag to set on a subsequent time-out. Timing for this mode is shown in Figure 19. The pulse must be low on the leading edge of 02.

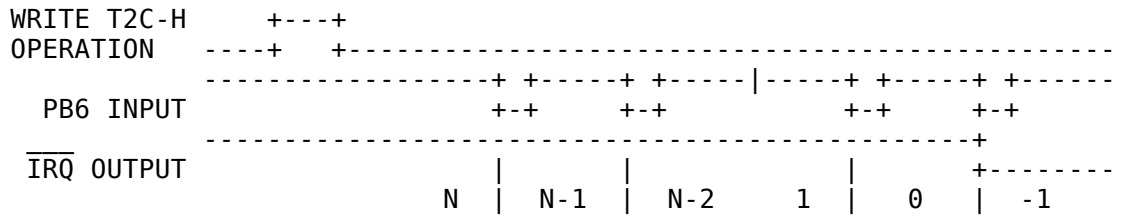
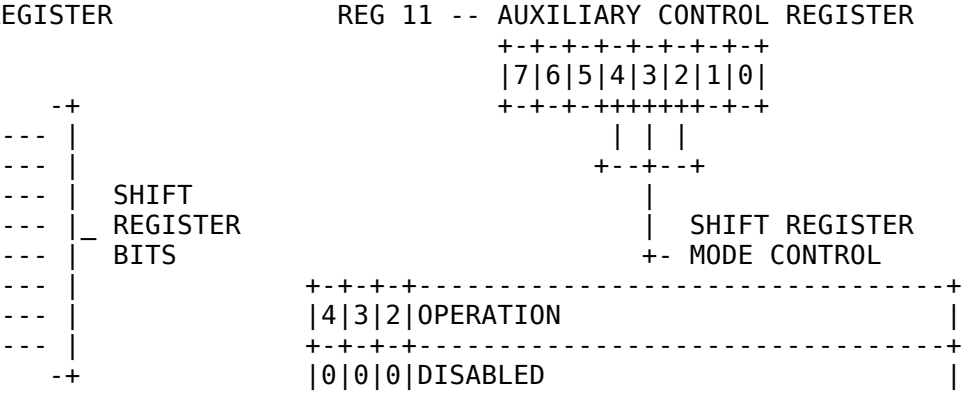


Figure 19. Timer 2 Pulse Counting Mode

SHIFT REGISTER OPERATION

The Shift Register (SR) performs serial data transfers into and out of the CB2 pin under control of an internal modulo-8 counter. Shift pulses can be applied to the CB1 pin from an external source or, with the proper mode selection, shift pulses generated internally will appear on the CB1 pin for controlling external devices.

The control bits which select the various shift register operating modes are located in the Auxiliary Control Register. Figure 20 illustrates the configuration of the SR data bits and Figure 21 shows the SR control bits of the ACR.



	0	0	1	SHIFT IN UNDER CONTROL OF T2
	0	1	0	SHIFT IN UNDER CONTROL OF 02
	0	1	1	SHIFT IN UNDER CONT. OF EXT.CLK
	1	0	0	SHIFT OUT FREE RUNNING AT T2 RATE
	1	0	1	SHIFT OUT UNDER CONTROL OF T2
	1	1	0	SHIFT OUT UNDER CONTROL OF 02
	1	1	1	SHIFT OUT UNDER CONT. OF EXT.CLK

NOTES

1 WHEN SHIFTING OUT BIT 7 IS THE FIRST BIT OUT AND SIMULTANEOUSLY IS ROTATED BACK INTO BIT 0

2 WHEN SHIFTING IN BITS INITIALLY ENTER BIT 0 AND ARE SHIFTED TOWARDS BIT 7

Figure 20. Shift registers

Figure 21. Shift Register Modes

SR Mode 0 -- Disabled

Mode 0 disables the Shift Register. In this mode the microprocessor can write or read the SR and the SR will shift on each CB1 positive edge shifting in the value on CB2. In this mode the SR Interrupt Flag is disabled (held to a logic 0).

SR Mode 1 -- Shift in Under Control of T2

In mode 1, the shifting rate is controlled by the low order 8 bits of T2 (Figure 22). Shift pulses are generated on the CB1 pin to control shifting in external devices. The time between transitions of this output clock is a function of the system clock period and the contents of the low order T2 latch (N).

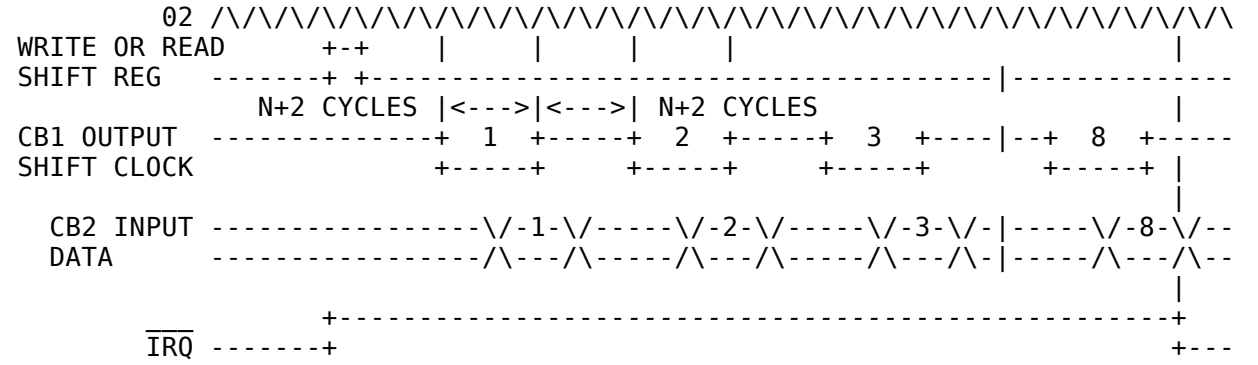


Figure 22. SR Mode 1 -- Shift In Under T2 Control

The shifting operation is triggered by the read or write of the SR if the SR flag is set in the IFR. Otherwise the first shift will occur at the next time-out of T2 after a read or write of the SR. Data is shifted first into the low order bit of SR and is then shifted into the next higher order bit of the shift register on the negative-going edge of each clock pulse. The input data should change before the positive-going edge of the CB1 clock pulse. This data is shifted into shift register during the 02 clock cycle following the positive-going edge of the CB1 clock pulse. After 8 CB1 clock pulses, the shift register interrupt flag will set and IRQ will go low.

SR Mode 2 -- Shift in Under 02 Control

In mode 2, the shift rate is a direct function of the system clock frequency (Figure 23). CB1 becomes an output which generates shift pulses for controlling external devices. Timer 2 operates as an independent interval timer and has no effect on SR. The shifting operation is triggered by reading or writing the Shift Register. Data is shifted, first into bit 0 and is then shifted into the next higher order bit of the shift register on the trailing edge of each 02 clock pulse. After 8 clock pulses, the shift register interrupt flag will be set, and the output clock pulses on CB1 will stop.

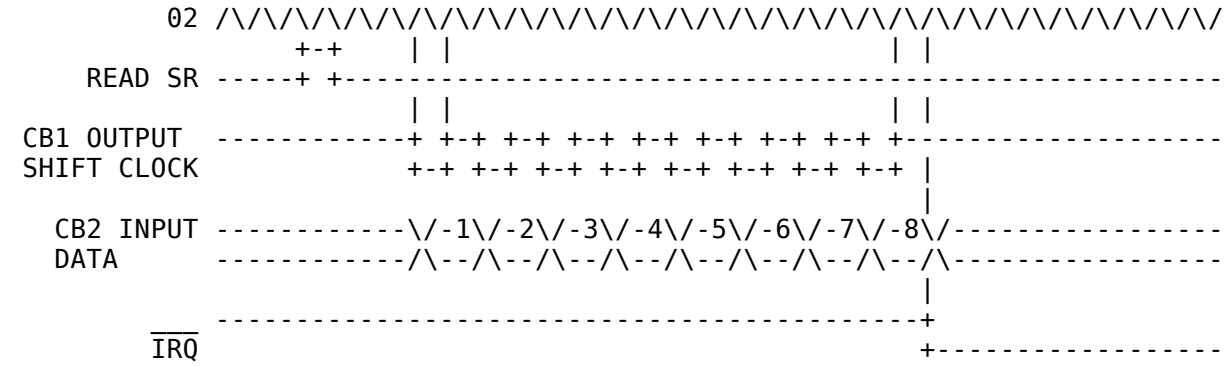


Figure 23. SR Mode 2 -- Shift In Under 02 Control

SR Mode 3 -- Shift in Under CB1 Control

In mode 3, external pin CB1 becomes an input (Figure 24). This allows an external device to load the shift register at its own pace. The shift register counter will interrupt the processor each time 8 bits have been shifted in. However, the shift register counter does not stop the shifting operation; it acts simply as a pulse counter. Reading or writing the Shift Register resets the Interrupt Flag and initializes the SR counter to count another 8 pulses.

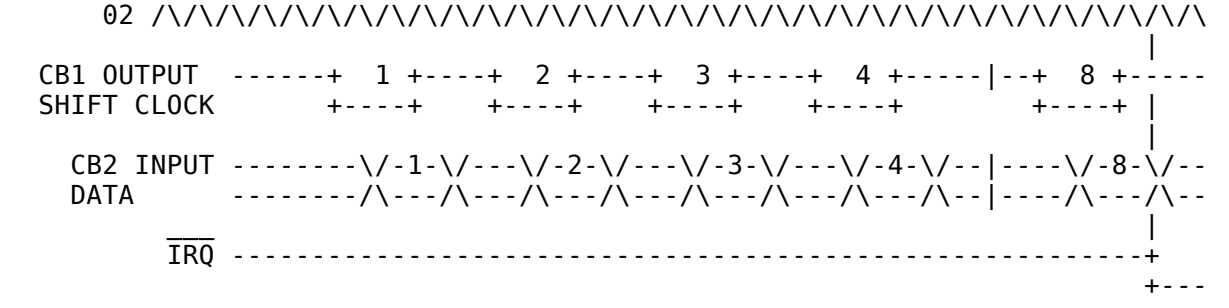


Figure 24. SR Mode 3 -- Shift In Under CB1 Control

Note that the data is shifted during the first system clock cycle following the positive-going edge of the CB1 shift pulse. For this reason, data must be held stable during the first full cycle following CB1 going high.

SR Mode 4 -- Shift Out Under T2 Control (Free-Run)

Mode 4 is very similar to mode 5 in which the shifting rate is set by T2. However, in mode 4 the SR Counter does not stop the shifting operation (Figure 25). Since the Shift Register bit 7 (SR7) is recirculated back into bit 0, the 8 bits loaded into the Shift Register will be clocked onto CB2 repetitively. In this mode the Shift Register Counter is disabled.

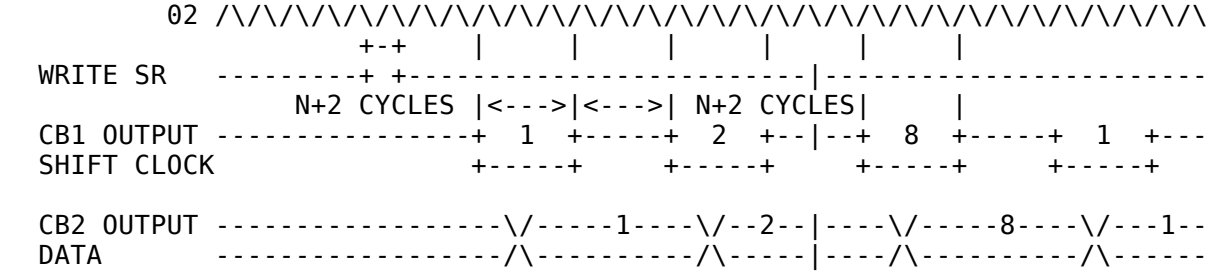


Figure 25. SR Mode 4 -- Shift Out Under T2 Control (Free-Run)

SR Mode 5 -- Shift Out Under T2 Control

In mode 5, the shift rate is controlled by T2 (as in mode 4). The shifting operation is triggered by the read or write of the SR if the SR flag is set in the IFR (Figure 26). Otherwise the first shift will occur at the next time-out of T2 after a read or write of the SR. However, with each read or write of the shift register the SR Counter is reset and 8 bits are shifted onto CB2. At the same time, 8 shift pulses are generated on CB1 to control shifting in external devices. After the 8 shift pulses, the shifting is disabled, the SR Interrupt Flag is set and CB2 remains at the last data level.

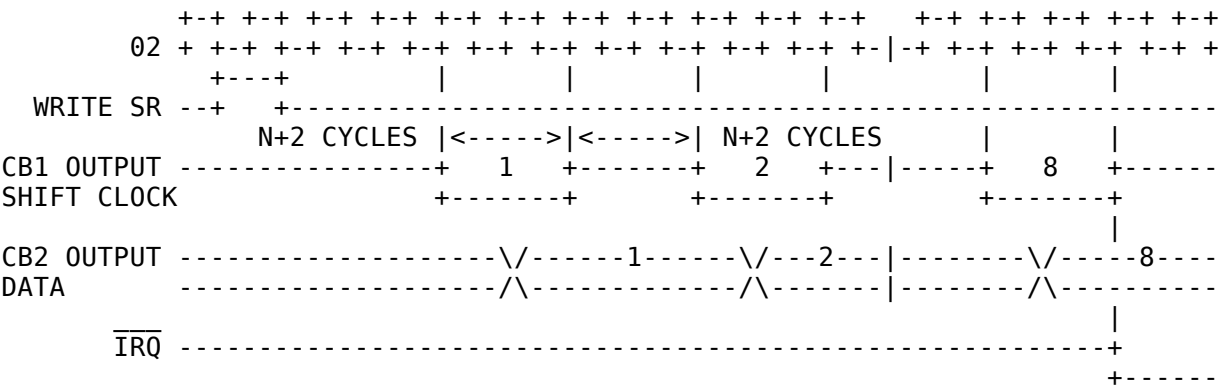


Figure 26. SR Mode 5 -- Shift Out Under T2 Control

SR Mode 6 -- Shift Out Under 02 Control

In mode 6, the shift rate is controlled by the 02 system clock (Figure 27).

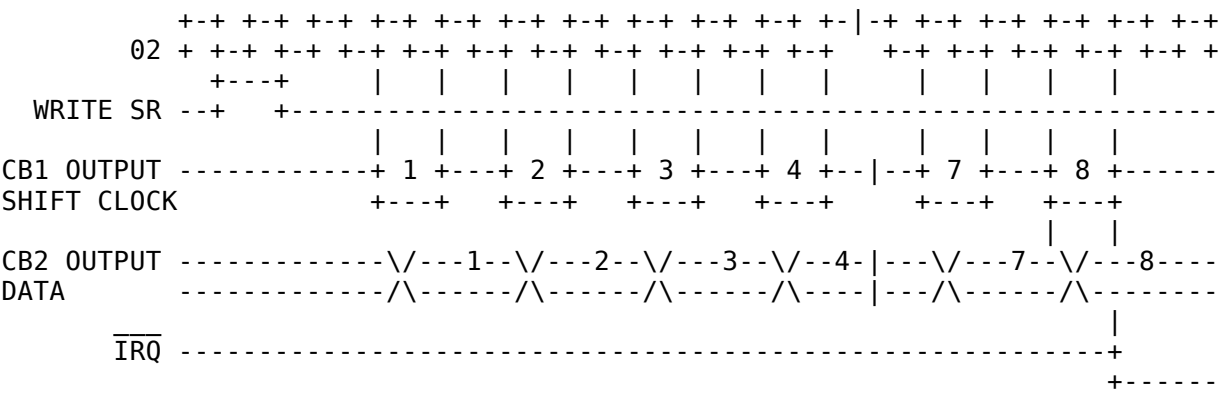
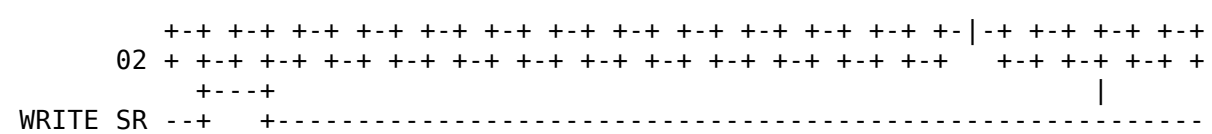


Figure 27. SR Mode 6 -- Shift Out Under 02 Control

SR Mode 7 -- Shift Out Under CB1 Control

In mode 7, shifting is controlled by pulses applied to the CB1 pin by an external device (Figure 28). The SR counter sets the SR Interrupt Flag each time it counts 8 pulses but it does not disable the shifting function. Each time the microprocessor, writes or reads the shift register, the SR Interrupt Flag is reset and the SR counter is initialized to begin counting the next 8 shift pulses on pin CB1. After 8 shift pulses, the Interrupt Flag is set. The microprocessor can then load the shift register with the next byte of data.



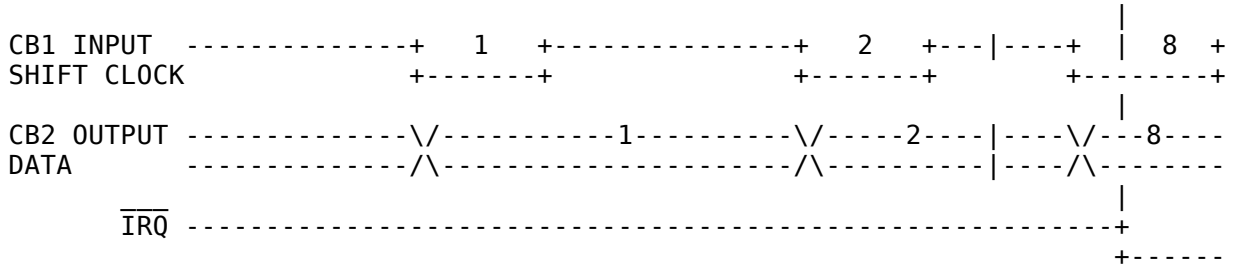


Figure 28. SR Mode 7 -- Shift Out Under CB1 Control

INTERRUPT OPERATION

Controlling interrupts within the R6522 involves three principal operations. These are flagging the interrupts, enabling interrupts and signaling to the processor that an active interrupt exists within the chip. Interrupt flags are set in the Interrupt Flag Register (IFR) by conditions detected within the R6522 or on inputs to the R6522. These flags normally remain set until the interrupt has been serviced. To determine the source of an interrupt, the microprocessor must examine these flags in order, from highest to lowest priority.

Associated with each interrupt flag is an interrupt enable bit in the Interrupt Enable Register (IER). This can be set or cleared by the processor to enable interrupting the processor from the corresponding interrupt flag. If an interrupt flag is set to a logic 1 by an interrupting condition, and the corresponding interrupt enable bit is set to a 1, the Interrupt Request (IRQ) output will go low. IRQ is an "open-collector" output which can be "wire-OR'ed" with other devices in the system to interrupt the processor.

Interrupt Flag Register (IFR)

In the R6522, all the interrupt flags are contained in one register, i.e., the IFR (Figure 29). In addition, bit 7 of this register will be read as a logic 1 when an interrupt exists within the chip. This allows very convenient polling of several devices within a system to locate the source of an interrupt.

REG 13 -- INTERRUPT FLAG REGISTER

7	6	5	4	3	2	1	0	SET BY	CLEARED BY
							CA2	CA2 ACTIVE EDGE	READ OR WRITE REG 1 (ORA)°
							CA1	CA1 ACTIVE EDGE	READ OR WRITE REG 1 (ORA)
							SHIFT REG	COMPLETE 8 SHIFTS	READ OR WRITE SHIFT REG
							CB2	CB2 ACTIVE EDGE	READ OR WRITE ORB°
							CB1	CB1 ACTIVE EDGE	READ OR WRITE ORB
							TIMER 2	TIME-OUT OF T2	READ T2 LOW OR WRITE T2 HIGH
							TIMER 1	TIME-OUT OF T1	READ T1 LOW OR WRITE T1 HIGH
							IRQ	ANY ENABLED INTERRUPT	CLEAR ALL INTERRUPTS

° IF THE CA2/CB2 CONTROL IN THE PCR IS SELECTED AS "INDEPENDENT" INTERRUPT INPUT, THEN READING OR WRITING THE OUTPUT REGISTER ORA/ORB WILL NOT CLEAR THE FLAG BIT. INSTEAD, THE BIT MUST BE CLEARED BY WRITING INTO THE IFR, AS DESCRIBED PREVIOUSLY.

Figure 29. Interrupt Flag Register (IFR)

The Interrupt Flag Register (IFR) may be read directly by the processor. In addition, individual flag bits may be cleared by writing a "1" into the appropriate bit of the IFR. When the proper chip select and register signals are applied to the chip, the contents of this register are placed on the data bus. Bit 7 indicates the status of the IRQ output. This bit corresponds to the logic function: IRQ = IFR6xIER6 + IFR5xIER5 + IFR4xIER4 + + IFR3xIER3 + IFR2xIER2 + IFR1xIER1 + IFR0xIER0

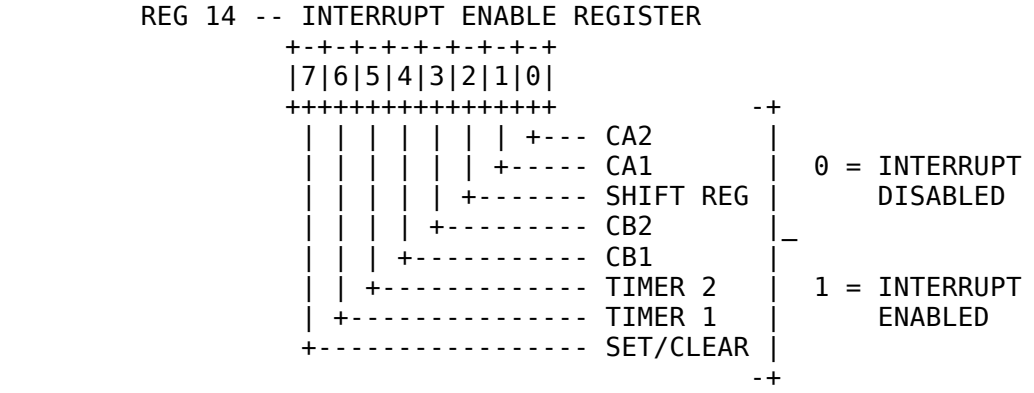
Note:

x = logic AND, + = logic OR

The IFR bit 7 is not a flag. Therefore, this bit is not directly cleared by writing a logic 1 into it. It can only be cleared by clearing all the flags in the register or by disabling all the active interrupts as discussed in next section.

Interrupt Enable Register (IER)

For each interrupt flag in IFR, there is a corresponding bit in the Interrupt Enable Register (IER) (Figure 30). Individual bits in the IER can be set or cleared to facilitate controlling individual interrupts without affecting others. This is accomplished by writing to the IER after bit 7 set or cleared to, in turn, set or clear selected enable bits. If bit 7 of the data placed on the system data bus during this write operation is a 0, each 1 in bits 6 through 0 clears the corresponding bit in the IER. For each zero in bits 6 through 0, the corresponding bit is unaffected.



NOTES:
1 IF BIT 7 IS A "0", THEN EACH "1" IN BITS 0-6 DISABLES THE CORRESPONDING INTERRUPT.
2 IF BIT 7 IS A "1", THEN EACH "1" IN BITS 0-6 ENABLES THE CORRESPONDING INTERRUPT.
3 IF A READ OF THIS REGISTER IS DONE, BIT 7 WILL BE "1" AND ALL OTHER BITS WILL REFLECT THEIR ENABLE/DISABLE STATE.

Figure 30. Interrupt Enable Register (IER)

Selected bits in the IER can be set by writing to the IER with bit 7 in the data word set to a logic 1. In this case, each 1 in bits 6 through 0 will set the corresponding bit. For each zero, the corresponding bit will be unaffected. This individual control of the setting and clearing operations allows very convenient control of the interrupts during system operation.

In addition to setting and clearing IER bits, the contents of this register can be read at any time. Bit 7 will be read as a logic 1, however.