MOS TECHNOLOGY, INC.
VALLEY FORGE CORPORATE CENTER (215) 666-7950
950 RITTENHOUSE ROAD, NORRISTOWN, PA. 19401

PRELIMINARY
DATA
SHEET
NOVEMBER 1977

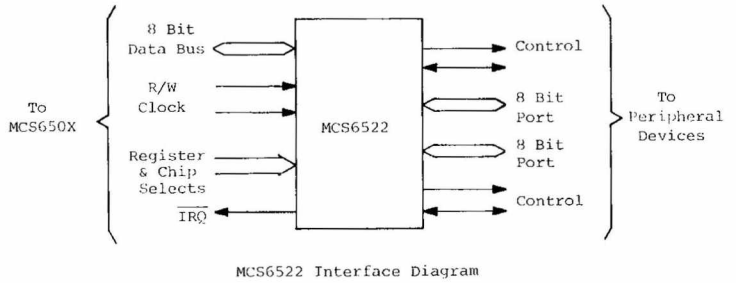
MCS6522 VERSATILE INTERFACE ADAPTER

DESCRIPTION

The MCS6522 Versatile Interface Adapter (VIA) provides all of the capability of the MCS6520. In addition, this device contains a pair of very powerful 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 bi-directional data transfers between VIA's in multiple processor systems.

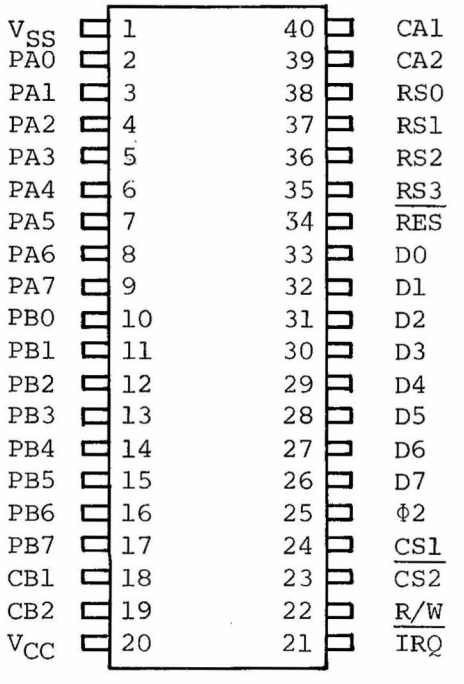
Control of peripheral devices is handled primarily through two 8-bit bi-directional ports. Each of these lines can be programmed to act as either an input or an output. Also, several peripheral I/O lines can be controlled directly from the interval timers for generating programmable-frequency square waves and for counting externally generated pulses. To facilitate control of the many powerful features of this chip, the internal registers have been organized into an interrupt flag register, an interrupt enable register and a pair of function control registers.

- ✓ Very powerful expansion of basic MCS6520 capability.
- ✓ N channel, depletion load technology, single +5V supply.
- ✓ Completely static and TTL compatible.
- ✓ CMOS compatible peripheral control lines.
- ✓ Expanded "handshake" capability allows very positive control of data transfers between processor and peripheral devices.



MCS6522 Interface Diagram

MCS6522



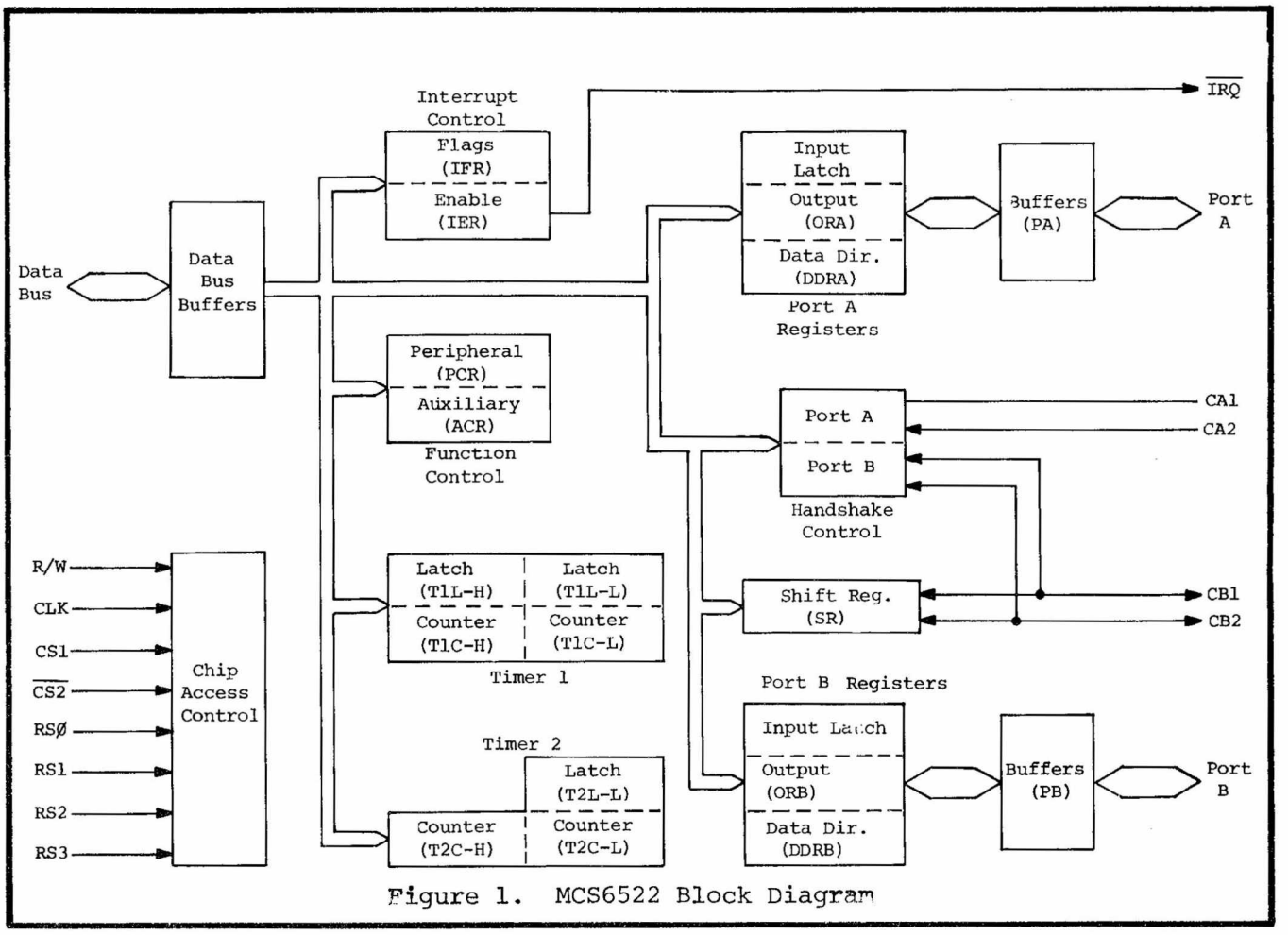


Figure 1. MCS6522 Block Diagram

PROCESSOR INTERFACE

This section contains a description of the buses and control lines which are used to interface the MCS6522 to the system processor. AC and DC parameters associated with this interface are specified on pages 21 through 24 of this document.

1. PHASE TWO CLOCK (Φ2)

Data transfers between the MCS6522 and the system processor take place only while the Phase Two Clock is high. In addition, Φ2 acts as the time base for the various timers, shift registers, etc. on the chip.

2. CHIP SELECT LINES (CS1, $\overline{CS2}$)

The two chip select inputs are normally connected to processor address lines either directly or through decoding. The selected MCS6522 register will be accessed when CS1 is high and $\overline{CS2}$ is low.

3. REGISTER SELECT LINES (RS0, RS1, RS2, RS3)

The four Register select lines are normally connected to the processor address bus lines to allow the processor to select the internal MCS6522 register which is to be accessed. The sixteen possible combinations access the registers as follows:

RS3	RS2	RS1	RS0	Register	Remarks	RS3	RS2	RS1	RS0	Register	Remarks
L	L	L	L	ORB		H	L	L	L	T2L-L T2C-L	Write Latch Read Counter
L	L	L	H	ORA	Controls Handshake	H	L	L	H	T2C-H	Triggers T2L-L/ T2C-L Transfer
L	L	H	L	DDRB		H	L	H	L	SR	
L	L	H	H	DDRA		H	L	H	H	ACR	
L	H	L	L	T1L-L T1C-L	Write Latch Read Counter	H	H	L	L	PCR	
L	H	L	H	T1C-H	Trigger T1L-L/ T1C-L Transf.	H	H	L	H	IFR	
L	H	H	L	T1L-L		H	H	H	L	IER	
L	H	H	H	T1L-H		H	H	H	H	ORA	No Effect on Handshake

Note: L = 0.4V DC, H = 2.4V DC.

4. READ/WRITE LINE (R/W)

The direction of data transfers between the MCS6522 and the system processor is controlled by the R/W line. If R/W is low, data will be transferred out of the processor into the selected MCS6522 register (write operation). If R/W is high and the chip is selected, data will be transferred out of the MCS6522 (read operation).

5. DATA BUS (DB0 - DB7)

The 8 bi-directional data bus lines are used to transfer data between the MCS6522 and the system processor. The internal drivers will remain in the high-impedance state except when the chip is selected (CS1 = 1, $\overline{CS2}$ = 0), Read/Write is high and the Phase Two Clock is high. At this time, the contents of the selected register are placed on the data bus. When the chip is selected, with Read/Write low and $\phi 2$ = 1, the data on the data bus will be transferred into the selected MCS6522 register.

6. RESET ($\overline{RES}$)

The Reset input clears all internal registers to logic 0 (except T1, T2 and SR). This places all peripheral interface lines in the input state, disables the timers, shift register, etc. and disables interrupting from the chip.

7. 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 INTERFACE

This section contains a brief description of the buses and control lines which are used to drive peripheral devices under control of the internal MCS6522 registers.

1. PERIPHERAL A PORT (PA0 - PA7)

The Peripheral A port consists of 8 lines which can be individually programmed to act as an input or an output under control of a Data Direction Register. The polarity of output pins is controlled by an Output Register and input data can 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 represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode.

2. PERIPHERAL A CONTROL LINES (CA1, CA2)

The two peripheral 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 Peripheral A Port input lines. The various modes of operation are controlled by the system processor through the internal control registers. 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.

3. PERIPHERAL B PORT (PB0 - PB7)

The Peripheral B port consists of 8 bi-directional lines which are controlled by an output register and a data direction register in much the same manner as the PA port. 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. These 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 30 ma at 1.5 VDC in the output mode to allow the outputs to directly drive Darlington transistor switches.

4. PERIPHERAL B CONTROL LINES (CB1, CB2)

The Peripheral 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. 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 switches.

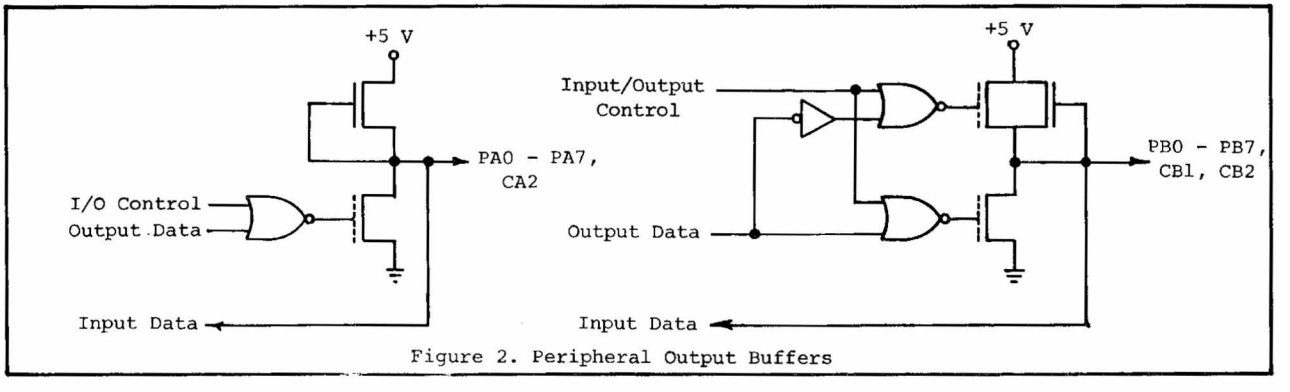


Figure 2. Peripheral Output Buffers

MCS6522 OPERATION

This section contains a discussion of the various blocks of logic shown in Figure 1. In addition, the internal operation of the MCS6522 is described in detail.

A. DATA BUS BUFFERS (DB), PERIPHERAL A BUFFERS (PA), PERIPHERAL B BUFFERS (PB)

The characteristics of the buffers which provide the required voltage and current drive capability were discussed in the previous section. AC and DC parameters for these buffers are specified on pages 21 through 24 of this document.

B. CHIP ACCESS CONTROL

The Chip Access Control contains the necessary logic to detect the chip select condition and to decode the Register Select inputs to allow accessing the desired internal register. In addition, the R/W and $\Phi 2$ signals are utilized to control the direction and timing of data transfers. When writing into the MCS6522, data is first latched into a data input register during $\Phi 2$. Data is then transferred into the desired internal register during $\Phi 2 \cdot$ Chip Select. This allows the peripheral I/O line to change without "glitching." When the processor reads the MCS6522, data is transferred from the desired internal register directly onto the Data Bus during $\Phi 2$.

C. PORT A REGISTERS, PORT B REGISTERS

Three registers are used in accessing each of the 8-bit peripheral ports. Each 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 an Input Register (IRA, IRB). When the pin is programmed to act 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 pin to go high, and a 0 causes the pin to go low. Data can be written into Output Register bits corresponding to pins which are programmed to act as inputs; however, the pin will be 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 data on the PA pins. With input latching enabled, IRA will reflect the contents of the Port A prior to setting the CA1 Interrupt Flag (IFR1) by an active transition on CA1.

The IRB register operates in a similar manner. However, for output pins, the corresponding IRB bit will reflect the contents of the Output Register bit instead of the actual pin. This allows proper data to be read into the processor if the output pin is not allowed to go to full voltage. With input latching enabled on Port B, setting CB1 interrupt flag will cause IRB to latch this combination of input data and ORB data until the interrupt flag is cleared.

D. HANDSHAKE CONTROL

The MCS6522 allows very 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.

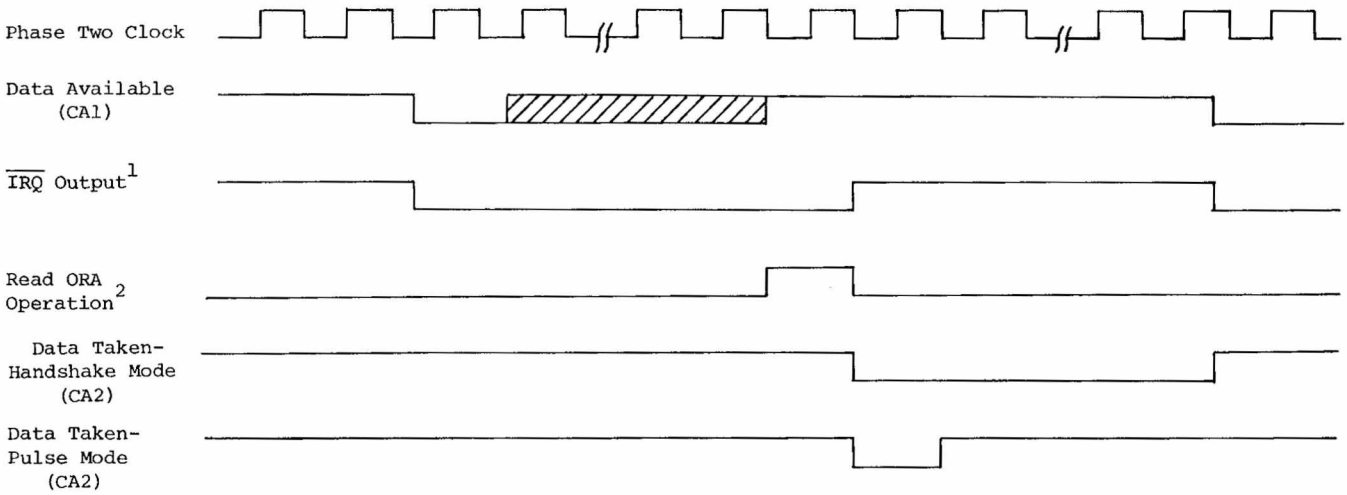
Read Handshake

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 "Data Ready" to signal the processor 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 MCS6522, 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 can be polled under software control. The Data Taken signal can be either a pulse or a DC level which is set low by the system processor and is cleared by the Data Ready signal. These options are shown in Figure 3 which illustrates the normal Read Handshaking sequence.

Write Handshake

The sequence of operations which allows handshaking data from the system processor to a peripheral device is very similar to that described in Section A for Read Handshaking. However, for "Write" handshaking, the processor must generate the "Data Ready" signal (through the MCS6522) 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 MCS6522. CA2 or CB2 acts as a Data Ready output in either the DC level or pulse mode and CA1 or CB1 accepts the "Data Taken" signal from the peripheral device, setting the interrupt flag and clearing the "Data Ready" output. This sequence is shown in Figure 4.

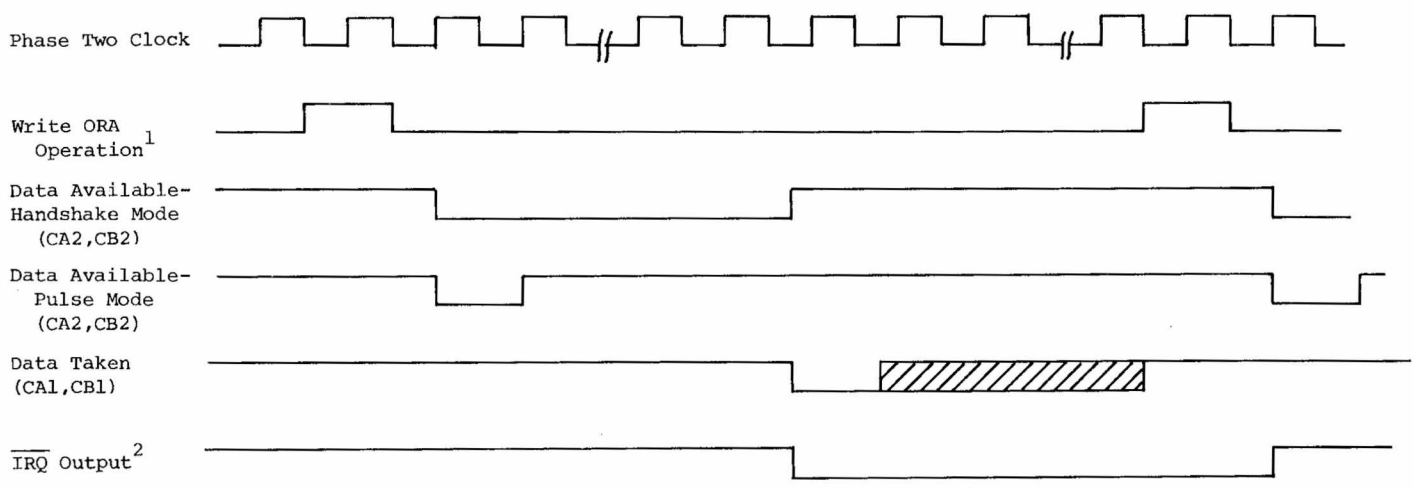


Notes:

1. Signals "data available" to the system processor.

2. R/W = 1, $\overline{CS2}$ = 0, CS1 = 1, RS3 = 0, RS2 = 0, RS1 = 0, RS0 = 1.

Figure 3. Read Handshake Timing Sequence



Notes:

1. R/W = 0, $\overline{CS2}$ = 0, CS1 = 1, RS3 = 0, RS2 = 0, RS1 = 0, RS0 = 1.

2. Signals "data taken" to the system processor.

Figure 4. Write Handshake Timing Sequence

C. TIMER 1

Introduction

Interval Timer T1 consists of two 8-bit latches and a 16-bit counter. The latches are used to store data which is to be loaded into the counter. After loading, the counter decrements at system clock rate,i.e., under control of the clock applied to the Phase Two input pin. Upon reaching zero, an interrupt flag will be set, and $\overline{IRQ}$ will go low. The timer will then disable any further interrupts, or will automatically transfer the contents of the latches into the counter and will continue to decrement. In addition, the timer can be instructed to invert the output signal on a peripheral pin each time it "times-out". Each of these modes is discussed seperately below.

Writing the Timer 1 Registers

The operations which take place when writing to each of the four T1 addresses are as follows:

RS3	RS2	RS1	RS0	Operation (R/W = L)
L	H	L	L	Write into low order latch.
L	H	L	H	Write into high order latch. Write into high order counter. Transfer low order latch into low order counter. Reset T1 interrupt flag.
L	H	H	L	Write low order latch.
L	H	H	H	Write high order latch. Reset T1 interrupt flag.

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 when the processor writes into the high order counter. 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 counter.

The second set of addresses allows the processor to write into the latch register without affecting the count-down in progress.This is discussed in detail below.

Reading the Timer 1 Registers

For reading the Timer 1 registers, the four addresses relate directly to the four registers as follows:

RS3	RS2	RS1	RS0	Operation (R/W = H)
L	H	L	L	Read T1 low order counter. Reset T1 interrupt flag.
L	H	L	H	Read T1 high order counter.
L	H	H	L	Read T1 low order latch.
L	H	H	H	Read T1 high order latch.

Timer 1 Operating Modes

Two bits are provided in the Auxiliary Control Register to allow selection of the T1 operating modes. These bits and the four possible modes are as follows:

ACR7 Output Enable	ACR6 "Free-Run" Enable	Mode
0	0	Generate a single time-out interrupt each time T1 is loaded. PB7 disabled.
0	1	Generate continuous interrupts.PB7 disabled.
1	0	Generate a single interrupt and an output pulse on PB7 for each T1 load operation.
1	1	Generate continuous interrupts and a square wave output on PB7.

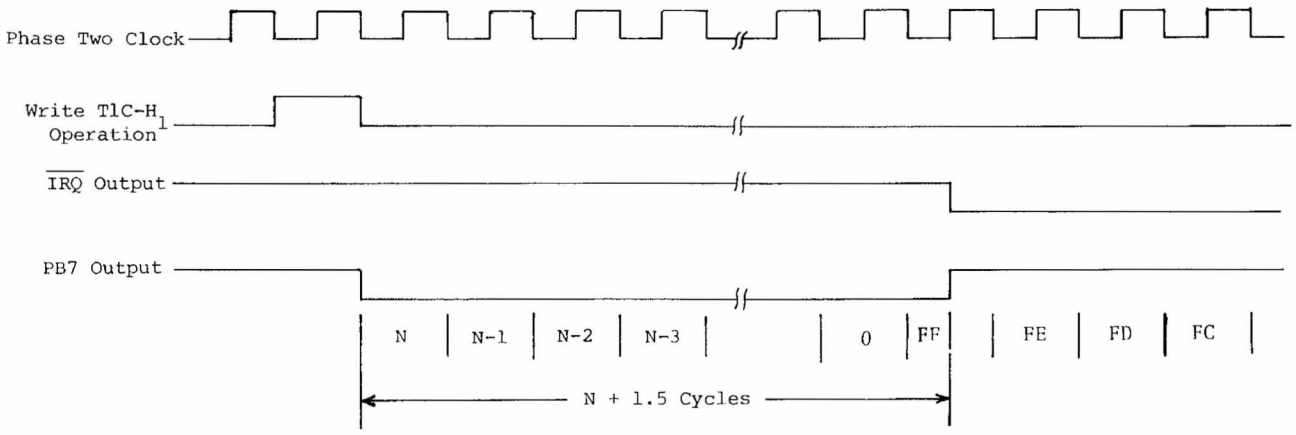
Timer 1 One-Shot Mode

The interval timer one-shot mode allows generation of 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.

> NOTE
> PB7 will act as an output if DDRB7 = 1 or if ACR7 = 1. However, if both DDRB7 and ACR7 are logic 1, PB7 will be controlled from Timer 1 and ORB7 will have no effect on the pin.

In the one-shot mode, writing into the high order latch 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, 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 phase two following the write operation. When the counter reaches zero, the 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 on page 13 of this specification.

Timing for the MCS6522 interval timer one-shot modes is shown in Figure 5.



Notes:
1. R/W = L, $\overline{CS2}$ = L, CS1 = H, RS3 = L, RS2 = H, RS1 = L, RS0 = H.

Figure 5. Interval Timer "One-shot" Mode Timing Sequence.

Timer 1 Free-Running 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 (ACR6 = 1), 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 below. 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 MCS6500 family devices 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. Timing for the free-running mode is shown in Figure 6.

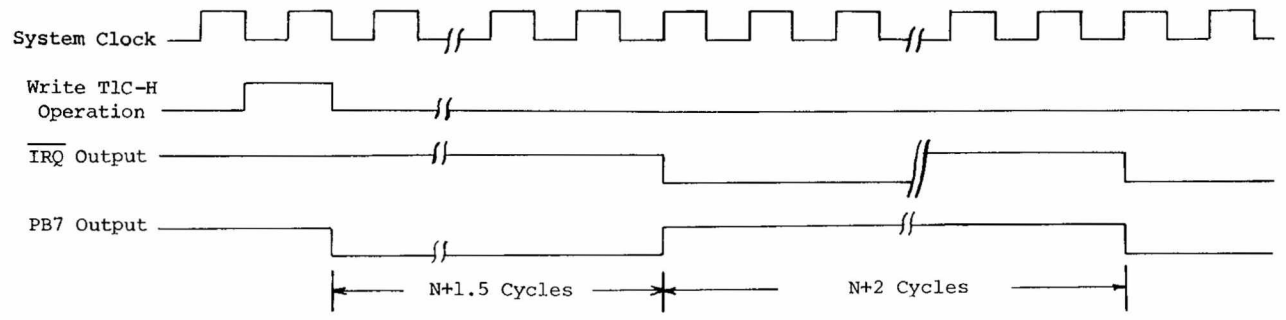


Figure 6. Timer 1 "Free-Running" Mode

F. TIMER 2

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 is provided in the Auxiliary Control Register to select between these two modes. This timer is comprised of a "write-only" low-order latch (T2L-L), a "read-only" low-order counter and a read/write high-order counter. The counter registers act as a 16-bit counter which decrements at Φ2 rate.

Timer 2 addressing can be summarized as follows:

RS3	RS2	RS1	RSØ	R/W = 0	R/W = 1
H	L	L	L	Write T2L-L	Read T2C-L Clear Interrupt flag
H	L	L	H	Write T2C-H Transfer T2L-L to T2C-L Clear Interrupt flag	Read T2C-H

Timer 2 Interval Timer 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 will be disabled after initial time-out so that it will not be set by the counter continuing to decrement 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 5.

Timer 2 Pulse Counting Mode

In the pulse counting mode, T2 serves primarily to count a predetermined number of negative-going pulses on PB6. This is accomplished by first loading a number into T2. 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 will be set when T2 reaches zero. At this time the counter will continue to decrement with each pulse on PB6. However, it is necessary to rewrite T2C-H to allow the interrupt flag to set on subquent down-counting operations. Timing for this mode is shown in Figure 7. The pulse must be low on the leading edge Ø2

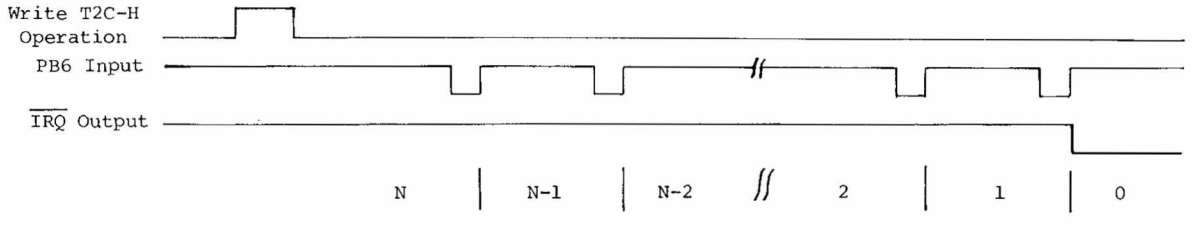


Figure 7. Timer 2 Pulse Counting Mode

G. SHIFT REGISTER

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 shifting in external devices.

The control bits which allow control of the various shift register operating modes are located in the Auxiliary Control Register. These bits can be set and cleared by the system processor to select one of the operating modes discussed in the following paragraphs.

Shift Register Input Modes

Bit 4 of the Auxiliary Control Register selects the input or output modes. There are three input modes and four output modes, differing primarily in the source of the pulses which control the shifting operation. With ACR4 = 0 the input modes are selected by ACR3 and ACR2 as follows:

ACR4	ACR3	ACR2	Mode
0	0	0	Shift Register Disabled
0	0	1	Shift in under control of Timer 2
0	1	0	Shift in at System Clock Rate.
0	1	1	Shift in under control of external input pulses

Mode 000 - Shift Register Disabled

The 000 mode is used to disable the Shift Register. In this mode the microprocessor can write or read the SR, but the shifting operation is disabled and operation of CB1 and CB2 is controlled by the appropriate bits in the Peripheral Control Register (PCR). In this mode the SR Interrupt Flag is disabled (held to a logic 0).

Mode 001 - Shift in under Control of Timer 2

In this mode the shifting rate is controlled by the low order 8 bits of T2. 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.

The shifting operation is triggered by writing or reading the shift register. 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 trailing edge of each clock pulse. As shown in Figure 8, the input data should change before the leading edge of the clock pulse. This data is loaded into the shift register during the system clock cycle following the trailing edge of the clock pulse. After 8 clock pulses, the shift register interrupt flag will be set and $\overline{IRQ}$ will go low.

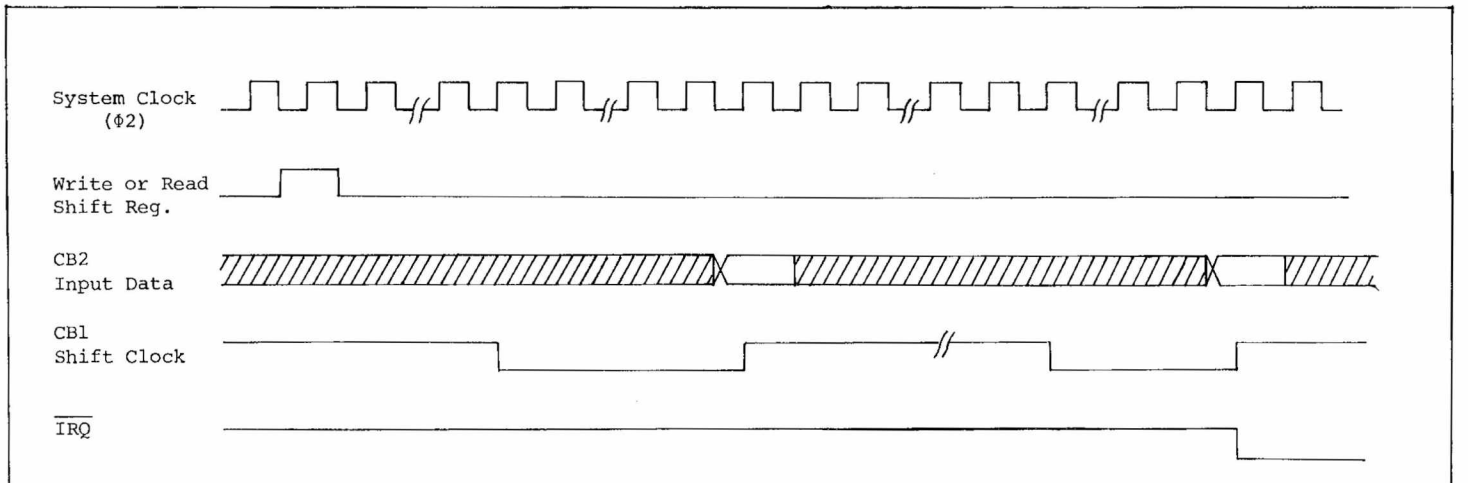


Figure 8. Shifting in Under Control of T2

SHIFT REGISTER (Cont.)

Mode 010 - Shift in at System Clock Rate

In this mode the shift rate is a direct function of the system clock frequency. 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 clock pulse. After 8 clock pulses, the shift register interrupt flag will be set, and the output clock pulses on CB1 will stop.

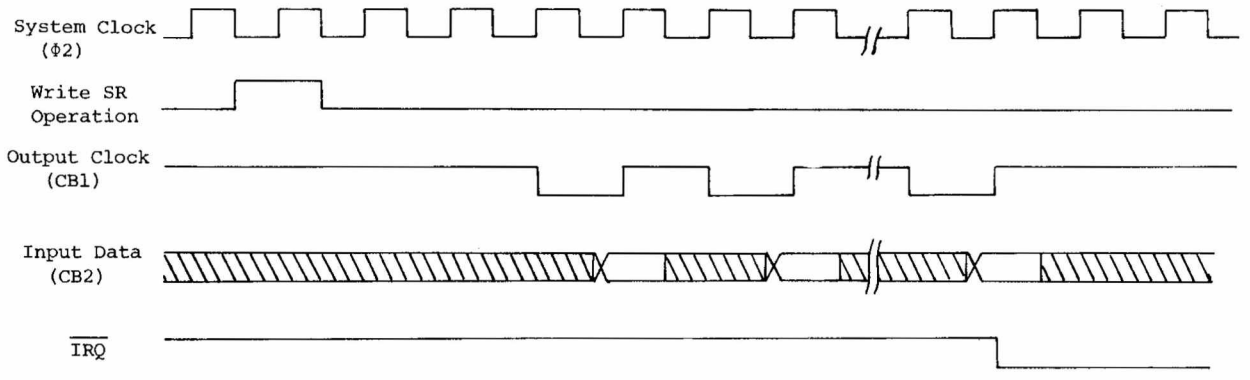
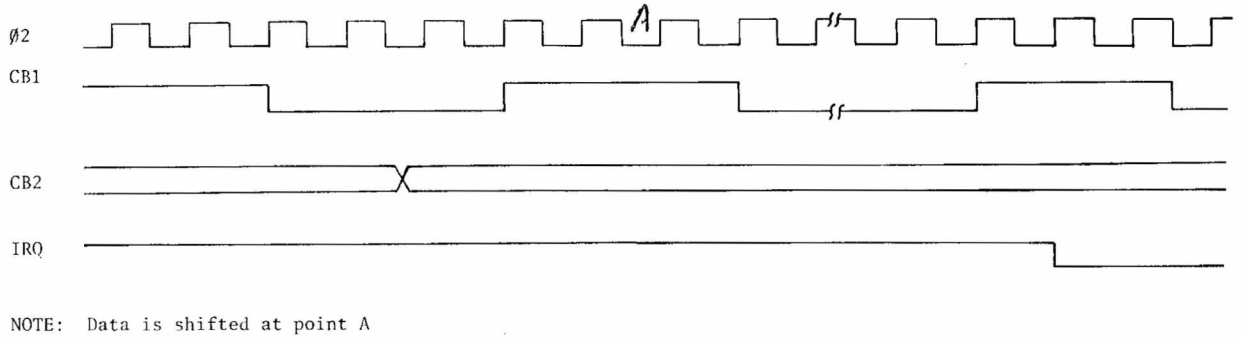


Figure 9. Timing Sequence for Shifting in at System Clock Rate

Mode 011 - Shift in under Control of External Clock

In this mode CB1 becomes an input. 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.

Note that data is shifted during the first system clock cycle following the leading edge of the CB1 shift pulse. For this reason, data must be held stable during the first full cycle following CB1 going high. Timing for this operation is illustrated in Figure 10.



NOTE: Data is shifted at point A

Figure 10. Timing sequence for shifting in under control of external clock

Shift Register Output Modes

The four Shift Register Output Modes are selected by setting the Input/Output Control Bit (ACR4) to a logic 1 and then selecting the specific output mode with ACR3 and ACR2. In each of these modes the Shift Register shifts data out of bit 7 to the CB2 pin. At the same time the contents of bit 7 are shifted back into bit 0. As in the input modes, CB1 is used either as an output to provide shifting pulses out or as an input to allow shifting from an external pulse. The four modes are as follows:

ACR4	ACR3	ACR2	Mode
1	0	0	Shift out - Free-running mode. Shift rate controlled by T2.
1	0	1	Shift out - Shift rate controlled by T2. Shift pulses generated on CB1.
1	1	0	Shift out at system clock rate.
1	1	1	Shift out under control of an external pulse.

Mode 100 Free-Running Output

This mode is very similar to mode 101 in which the shifting rate is set by T2. However, in mode 100 the SR Counter does not stop the shifting operation. 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.

Mode 101 - Shift out under Control of T2

In this mode the shift rate is controlled by T2 (as in the previous mode). 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 goes to a state determined by the CB2 Control bit (PC5) in the Peripheral Control Register.

If the shift register is reloaded before the last time-out, the shifting will continue. This sequence is illustrated in Figure 11.

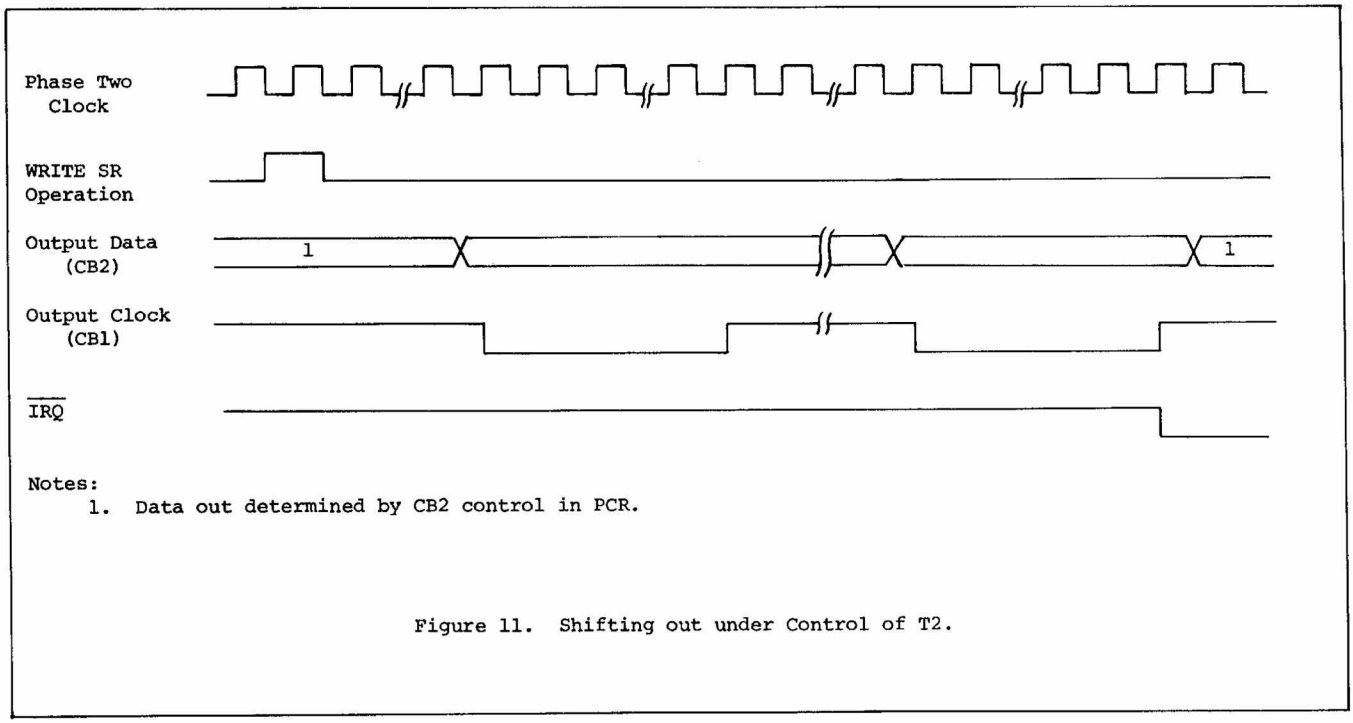


Figure 11. Shifting out under Control of T2.

Mode 110 - Shifting out at System Clock Rate

In this mode the shift register operation is similar to that shown in Figure 11. However, the shifting rate is a function of the system clock on the chip enable pin (Φ2) and is independent of T2. Timer 2 resumes its normal function as an independent interval timer. Figure 12 illustrates the timing sequence for mode 110.

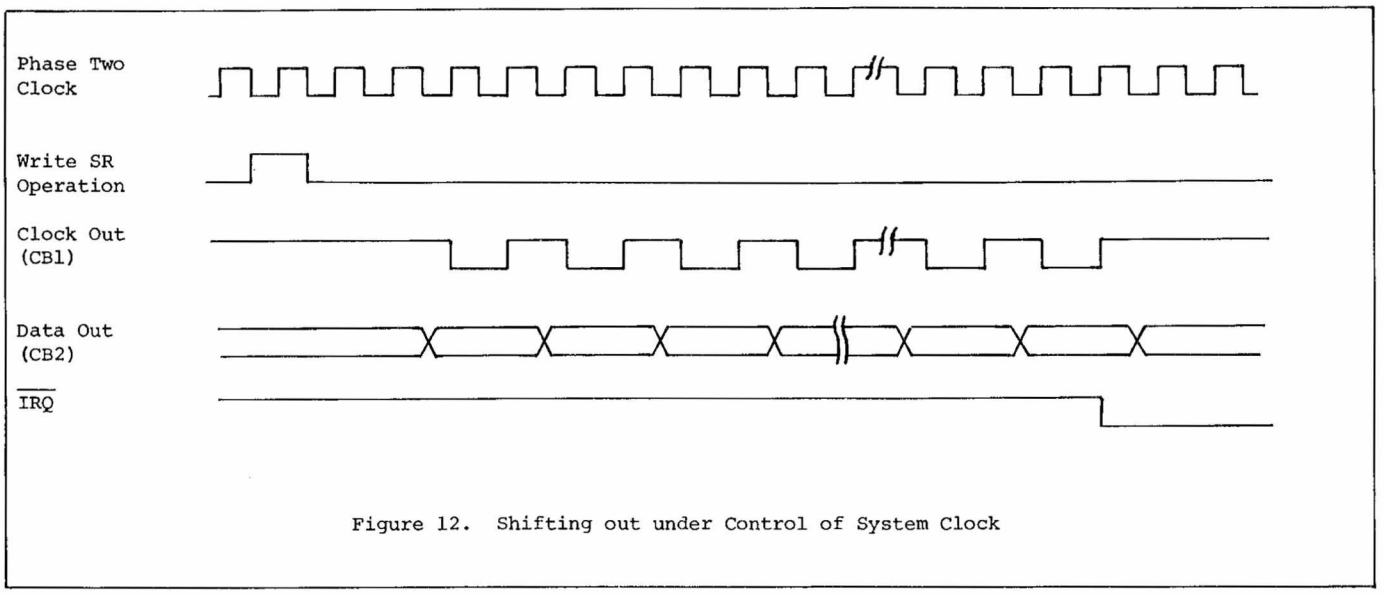


Figure 12. Shifting out under Control of System Clock

Mode 111 - Shift out under Control of an External Pulse

In this mode, shifting is controlled by pulses applied to the CB1 pin by an external device. 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.

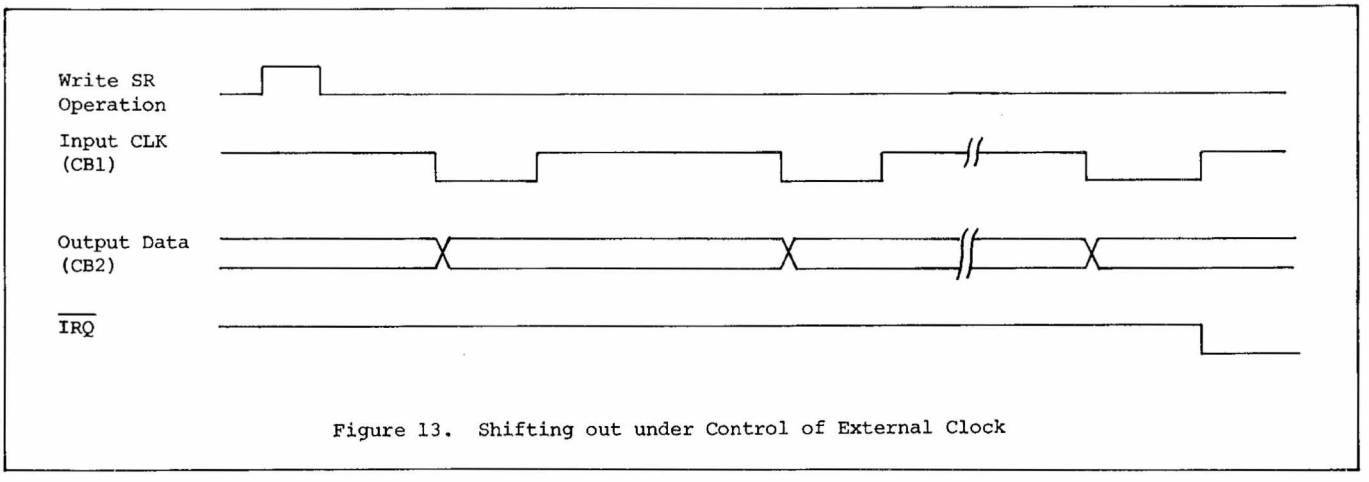


Figure 13. Shifting out under Control of External Clock

H. INTERRUPT CONTROL

Controlling interrupts within the MCS6522 involves three principal operations. These are flagging the interrupts, enabling interrupts and signalling to the processor that an active interrupt exists within the chip. Interrupt flags are set by interrupting conditions which exist within the chip or on inputs to the chip. 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. This is accomplished by reading the flag register into the processor accumulator, shifting this register either right or left and then using conditional branch instructions to detect an active interrupt.

Associated with each interrupt flag is an interrupt enable bit. This bit 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 Output ($\overline{IRQ}$) will go low. $\overline{IRQ}$ is an "open-collector" output which can be "wire or'ed" with other devices in the system to interrupt the processor.

In the MCS6522, all the interrupt flags are contained in one register. 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.

	7	6	5	4	3	2	1	0
Interrupt Flag Register	IRQ	T1	T2	CB1	CB2	SR	CA1	CA2
Interrupt Enable Register	Set/ clear control	T1	T2	CB1	CB2	SR	CA1	CA2

Interrupt Flag Register

The IFR is a read/bit-clear register. 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 = IFR6 X IER6 + IFR5 X IER5 + IFR4 X IER4 + IFR3 X IER3 + IFR2 X IER2 + IFR1 X IER1 + IFR0 X IER0. Note: X = logic AND, + = Logic OR.

Bits six through zero are latches which are set and cleared as follows:

Bit #	Set by	Cleared by
0	Active transition of the signal on the CA2 pin.	Reading or writing the A Port Output Register (ORA) using address 0001.
1	Active transition of the signal on the CA1 pin.	Reading or writing the A Port Output Register (ORA) using address 0001.
2	Completion of eight shifts	Reading or writing the Shift Register.
3	Active transition of the signal on the CB2 pin.	Reading or writing the B Port Output Register.
4	Active transition of the signal on the CB1 pin.	Reading or writing the B Port Output Register.
5	Time-out of Timer 2.	Reading T2 low order counter. Writing T2 high order counter.
6	Time-out of Timer 1.	Reading T1 low order counter. Writing T1 high order latch.

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 the next section.

Interrupt Enable Register (IER)

For each interrupt flag in IFR, there is a corresponding bit in the Interrupt Enable Register. The system processor can set or clear selected bits in this register to facilitate controlling individual interrupts without affecting others. This is accomplished by writing to address 1110 (IER address). 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 Interrupt Enable Register. For each zero in bits 6 through 0, the corresponding bit is unaffected.

Setting selected bits in the Interrupt Enable Register is accomplished by writing to the same address 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 interrupts during system operation.

In addition to setting and clearing IER bits, the processor can read the contents of this register by placing the proper address on the register select and chip select inputs with the R/W line high. Bit 7 will be read as a logic 0.

I. FUNCTION CONTROL

Control of the various functions and operating modes within the MCS6522 is accomplished primarily through two registers, the Peripheral Control Register (PCR), and the Auxiliary Control Register (ACR). The PCR is used primarily to select the operating mode for the four peripheral control pins. The Auxiliary Control Register selects the operating mode for the interval timers (T1, T2), and the serial port (SR).

Peripheral Control Register

The Peripheral Control Register is organized as follows:

Bit #	7	6	5	4	3	2	1	0
Function-	CB2 Control			CB1 Control	CA2 Control			CA1 Control

Each of these functions is discussed in detail below.

1. CA1 Control

Bit 0 of the Peripheral Control Register selects the active transition of the input signal applied to the CA1 interrupt input pin. If this bit is a logic 0, the CA1 interrupt flag will be set by a negative transition (high to low) of the signal on the CA1 pin. If PCR0 is a logic 1, the CA1 interrupt flag will be set by a positive transition (low to high) of this signal.

2. CA2 Control

The CA2 pin can be programmed to act as an interrupt input or as a peripheral control output. As an input, CA2 operates in two modes, differing primarily in the methods available for resetting the interrupt flag. Each of these two input modes can operate with either a positive or a negative active transition as described above for CA1.

In the output mode, the CA2 pin combines the operations performed on the CA2 and CB2 pins of the MCS6520. This added flexibility allows processor to perform a normal "write" handshaking in a system which uses CB1 and CB2 for the serial operations described above. The CA2 operating modes are selected as follows:

PCR3	PCR2	PCR1	Mode
0	0	0	Input mode--Set CA2 interrupt flag (IFR0) on a negative transition of the input signal. Clear IFR0 on a read or write of the Peripheral A Output Register.
0	0	1	Independent interrupt input mode--Set IFR0 on a negative transition of the CA2 input signal. Reading or writing ORA does not clear the CA2 interrupt flag.
0	1	0	Input mode--Set CA2 interrupt flag on a positive transition of the CA2 input signal. Clear IFR0 with a read or write of the Peripheral A Output Register.
0	1	1	Independent interrupt input mode--Set IFR0 on a positive transition of the CA2 input signal. Reading or writing ORA does not clear the CA2 interrupt flag.
1	0	0	Handshake output mode--Set CA2 output low on a read or write of the Peripheral A Output Register. Reset CA2 high with an active transition on CA1.
1	0	1	Pulse Output mode--CA2 goes low for one cycle following a read or write of the Peripheral A Output Register.
1	1	0	Manual output mode--The CA2 output is held low in this mode.
1	1	1	Manual output mode--The CA2 output is held high in this mode.

In the independent input mode, writing or reading the ORA register has no effect on the CA2 interrupt flag. This flag must be cleared by writing a logic 1 into the appropriate IFR bit. This mode allows the processor to handle interrupts which are independent of any operations taking place on the peripheral I/O ports.

The handshake and pulse output modes have been described previously. Note that the timing of the output signal varies slightly depending on whether the operation is initiated by a read or a write.

3. CB1 Control

Control of the active transition of the CB1 input signal operates in exactly the same manner as that described above for CA1. If PCR4 is a logic 0 the CB1 interrupt flag (IFR4) will be set by a negative transition of the CB1 input signal and cleared by a read or write of the ORB register. If PCR4 is a logic 1, IFR4 will be set by a positive transition of CB1.

If the Shift Register function has been enabled, CB1 will act as an input or output for the shift register clock signals. In this mode the CB1 interrupt flag will still respond to the selected transition of the signal on the CB1 pin.

4. CB2 Control

With the serial port disabled, operation of the CB2 pin is a function of the three high order bits of the PCR. The CB2 modes are very similar to those described previously for CA2. These modes are selected as follows:

PCR7	PCR6	PCR5	Mode
0	0	0	Interrupt input mode--Set CB2 interrupt flag (IFR3) on a negative transition of the CB2 input signal. Clear IFR3 on a read or write of the Peripheral B Output Register.
0	0	1	Independent interrupt input mode--Set IFR3 on a negative transition of the CB2 input signal. Reading or writing ORB does not clear the interrupt flag.
0	1	0	Input mode--Set CB2 interrupt flag on a positive transition of the CB2 input signal. Clear the CB2 interrupt flag on a read or write of ORB.
0	1	1	Independent input mode--Set IFR3 on a positive transition of the CB2 input signal. Reading or writing ORB does not clear the CB2 interrupt flag.
1	0	0	Handshake output mode--Set CB2 low on a write ORB operation. Reset CB2 high with an active transition of the CB1 input signal.
1	0	1	Pulse output mode--Set CB2 low for one cycle following a write ORB operation.
1	1	0	Manual output mode--The CB2 output is held low in this mode.
1	1	1	Manual output mode--The CB2 output is held high in this mode.

Auxiliary Control Register

Many of the functions in the Auxiliary Control Register have been discussed previously. However, a summary of this register is presented here as a convenient reference for the MCS6522 user. The Auxiliary Control Register is organized as follows:

Bit #	7	6	5	4	3	2	1	0
Function	T1 Control		T2 Control	Shift Register Control			PB Latch Enable	PA Latch Enable

1. PA Latch Enable

The MCS6522 provides input latching on both the PA and PB ports. In this mode, the data present on the peripheral A input pins will be latched within the chip when the CA1 interrupt flag is set. Reading the PA port will result in these latches being transferred into the processor. As long as the CA1 interrupt flag is set, the data on the peripheral pins can change without affecting the data in the latches. This input latching can be used with any of the CA2 input or output modes.

It is important to note that on the PA port, the processor always reads the data on the peripheral pins (as reflected in the latches). For output pins, the processor still reads the latches. This may or may not reflect the data currently in the ORA. Proper system operation requires careful planning on the part of the system designer if input latching is combined with output pins on the peripheral ports.

Input latching is enabled by setting bit 0 in the Auxiliary Control Register to a logic 1. As long as this bit is a 0, the latches will directly reflect the data on the pins.

2. PB Latch Enable

Input latching on the PB port is controlled in the same manner as that described for the PA port. However, with the peripheral B port the input latch will store either the voltage on the pin or the contents of the Output Register (ORB) depending on whether the pin is programmed to act as an input or an output. As with the PA port, the processor always reads the input latches.

3. Shift Register Control

The Shift Register operating mode is selected as follows:

ACR4	ACR3	ACR2	Mode
0	0	0	Shift Register Disabled.
0	0	1	Shift in under control of Timer 2.
0	1	0	Shift in under control of system clock.
0	1	1	Shift in under control of external clock pulses.
1	0	0	Free-running output at rate determined by Timer 2.
1	0	1	Shift out under control of Timer 2.
1	1	0	Shift out under control of the system clock.
1	1	1	Shift out under control of external clock pulses.

4. T2 Control

Timer 2 operates in two modes. If ACR5 = 0, T2 acts as an interval timer in the one-shot mode. If ACR5 = 1, Timer 2 acts to count a predetermined number of pulses on pin PB6.

5. T1 Control

Timer 1 operates in the one-shot or free-running mode with the PB7 output control enabled or disabled. These modes are selected as follows:

ACR7	ACR6	Mode
0	0	One-shot mode- Output to PB7 disabled.
0	1	Free-running mode- Output to PB7 disabled.
1	0	One-shot mode- Output to PB7 enabled.
1	1	Free-running mode. Output to PB7 enabled.

APPLICATION OF THE MCS6522

The MCS6522 represents a significant advance in general-purpose microprocessor I/O. Unfortunately, its many powerful features, coupled with a set of very flexible operating modes, cause this device to appear to be very complex at first glance. However, a detailed analysis will show that the VIA is organized to allow convenient control of these powerful features. This section seeks to assist the system designer in his understanding of the MCS6522 by illustrating how the device can be used in microprocessor-based systems.

A. Control of MCS6522 Interrupts

Organization of the MCS6522 interrupt flags into a single register greatly facilitates the servicing of interrupts from this device. Since there is only one IRQ output for the seven possible sources of interrupt within the chip, the processor must examine these flags to determine the cause of an interrupt. This is best accomplished by first transferring the contents of the flag register into the accumulator. At this time it may be necessary to mask off those flags which have been disabled in the Interrupt Enable Register. This is particularly important for the edge detecting inputs where the flags may be set whether or not the interrupting function has been enabled. Masking off those flags can be accomplished by performing an AND operation between the IER and the accumulator or by performing an "AND IMMEDIATE." The second byte of this AND # instruction should specify those flags which correspond to interrupt functions which are to be serviced.

If the N flag is set after these operations, an active interrupt exists within the chips. This interrupt can be detected with a series of shift and branch instructions.

Clearing interrupt flags is accomplished very conveniently by writing a logic 1 directly into the appropriate bit of the Interrupt Flag Register. This can be combined with an interrupt enable or disable operation as follows:

```
LDA #@10010000 ; initialize accumulator
STA IFR        ; clear interrupt flag
STA IER        ; set interrupt enable flag
```

or:

```
LDA #@00001000 ; initialize accumulator
STA IFR        ; clear interrupt flag
STA IER        ; disable interrupt
```

Another very useful technique for clearing interrupt flags is to simply transfer the contents of the flag register back into this register as follows:

```
LDA IFR  ;  transfer IFR to accumulator
STA IFR  ;  clear flags corresponding to active interrupts
```

After completion of this operation the accumulator will still contain the interrupt flag information. Most important, writing into the flag register clears only those flags which are already set. This eliminates the possibility of inadvertently clearing a flag while it is being set.

B. Use of Timer 1

Timer 1 represents one of the most powerful features of the MCS6522. The ability to generate very evenly spaced interrupts and the ability to control the voltage on PB7 makes this timer particularly valuable in various timing, data detection and waveform generation applications.

Time-of-Day Clock Applications

An important feature of many systems is the time-of-day clock. In microprocessor-based systems the time of day is usually maintained in memory and is updated in an interrupt service routine. A regular processor interrupt will then assure that this time of day will always be available when it is needed in the main program.

Generating very regular interrupts using previously available timers presented difficulties because of the need to re-load the timer for each interrupt. Unfortunately, the time between the interrupts will fluctuate due to variations in the interrupt response time. This problem is eliminated in the Timer 1 "free-running" mode. The accuracy of these "free-running" interrupts is only a function of the system clock and is not affected by interrupt response time.

Asynchronous Data Detection

The extraction of clock and data information from serial asynchronous ASCII signals or from any single channel data recording device relies on the ability to establish accurate strobes. As discussed previously, the period of these strobes can be seriously affected by the interrupt response time using conventional timers. However, T1 again allows generation of very accurate interrupts. The processor responds to these interrupts by strobing the input data. The ability to reload the T1 latches without affecting the count-down in progress is very useful in this application. This allows the strobe time to be doubled or halved during data detection. This sequence of operations is as follows:

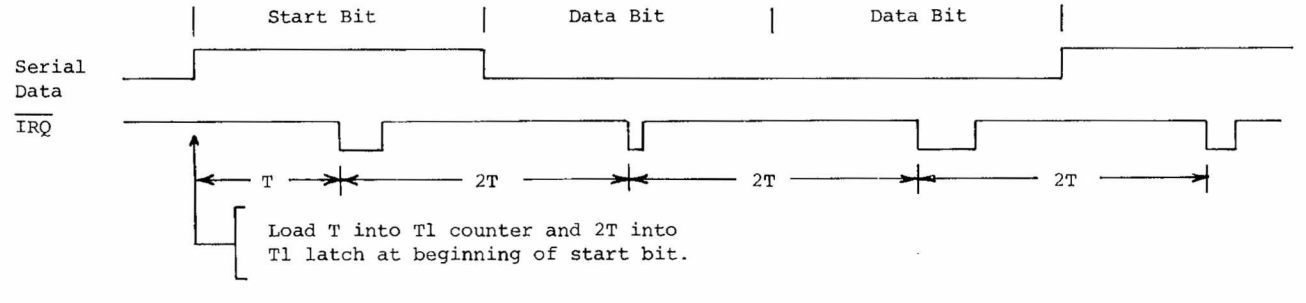


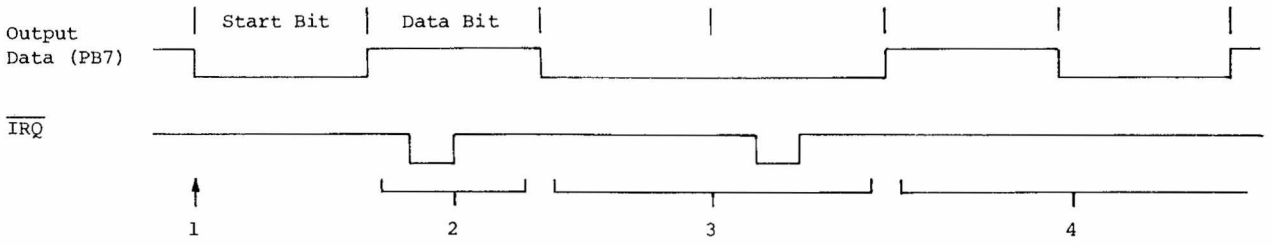
Figure 14. Detecting Asynchronous Data Using Timer 1

Waveform Generation with Timer 1

In addition to generating processor interrupts, Timer 1 can be used to control the output voltage on peripheral pin PB7 (output mode). In this mode a single negative pulse can be generated on PB7 (one-shot mode) or, in the free-running mode, a continuous waveform can be generated. In this latter mode the voltage on PB7 will be inverted each time T1 times out.

A single solenoid can be triggered very conveniently in the one-shot mode if the PB7 signal is used to control the solenoid directly. With this configuration the solenoid can be triggered by simply writing to T1C-H.

Generating very complex waveforms can be a simple problem if T1 is used to control PB7 in the free-running mode. During any count-down process the latches can be loaded to determine the length of the next count-down period. Figure 15 shows this timing sequence for generating ASCII serial data.

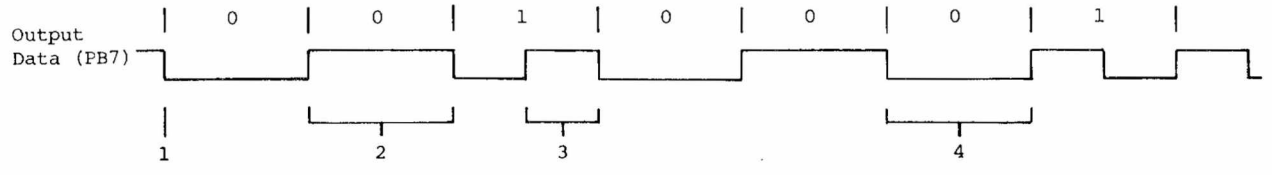


1. Load T into T1 counter and latch. Load T into T2 to trigger T1 latch reload.
2. Load 2T into T1 latch during this bit time. Load 2T into T2.
3. Load T into T1 latch anytime during this period. Load NT into T2. N = number of 1's or 0's which follow.
4. A series of 1's and 0's will be generated until the T1 latch is again changed. Note that the use of T2 to control reloading the T1 latch eliminates the need to interrupt on each transition.

Figure 15. ASCII Serial Data Generation Using T1

An application where this mode of operation is also very powerful is in the generation of bi-phase encoded data for tape or disk storage. This encoding technique and the sequence of operations which would take place are illustrated in Figure 16.

These applications represent only a tiny portion of the potential T1 applications. Some other possibilities are pulse width modulation waveforms, sound generation for video games, A/D techniques requiring very accurate pulse widths, and waveform synthesis in electronic games.



1. Load T1 counter and latch.
2. Shift T1 latch one bit to the right during this period.
3. Shift T1 latch left during this period.
4. Shift T1 latch right during this period.

Note that T1 must be accessed only when the output data changes. A string of 1's or 0's can be generated without processor intervention.

Figure 16. Generating Bi-phase Encoded Data

Using the MCS6522 Shift Register

The Shift Register in the MCS6522 is designed primarily as a synchronous serial communications port for distributed systems. These systems can be either single-processor with distributed peripheral controllers or distributed processor systems. The most important characteristic of the Shift Register in these applications is its ability to transfer information at relatively slow data rates to allow the use of R-C noise suppression techniques. This transfer can be accomplished while the processor is servicing other aspects of the system. An example of a simple 2-processor distributed system is shown in Figure 17. Use of the MCS6522 Shift Register allows effective communication between the two systems without the use of relatively complex asynchronous communications techniques.

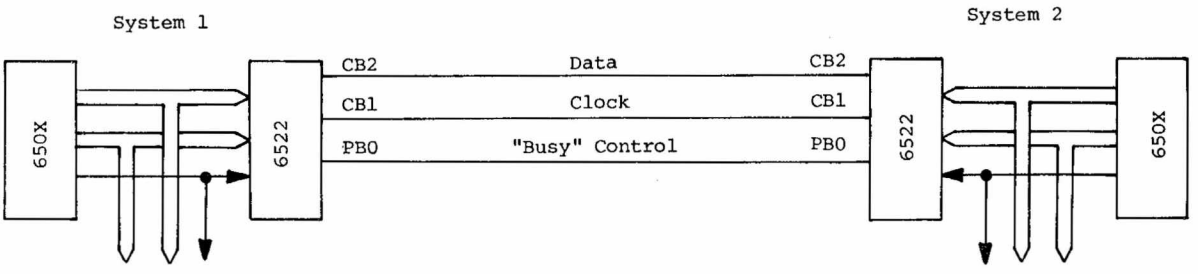


Figure 17. Using Shift Register for Inter-System Communication

In a system with distributed peripherals, the Shift Register can be used to transfer data to the peripheral interface devices. This is illustrated in Figure 18 for a system with a number of distributed status displays. These displays are serviced by stand-alone controllers which actuate the lamps in the status displays through simple drivers. The data and clock lines are wired in parallel to each unit. In addition, a single MCS6522 peripheral port output allows selection of the display to be loaded. These select lines can be eliminated if all displays are to contain the same information. With the system shown, the status display can be updated at any time by simply selecting the desired display and then writing to the Shift Register.

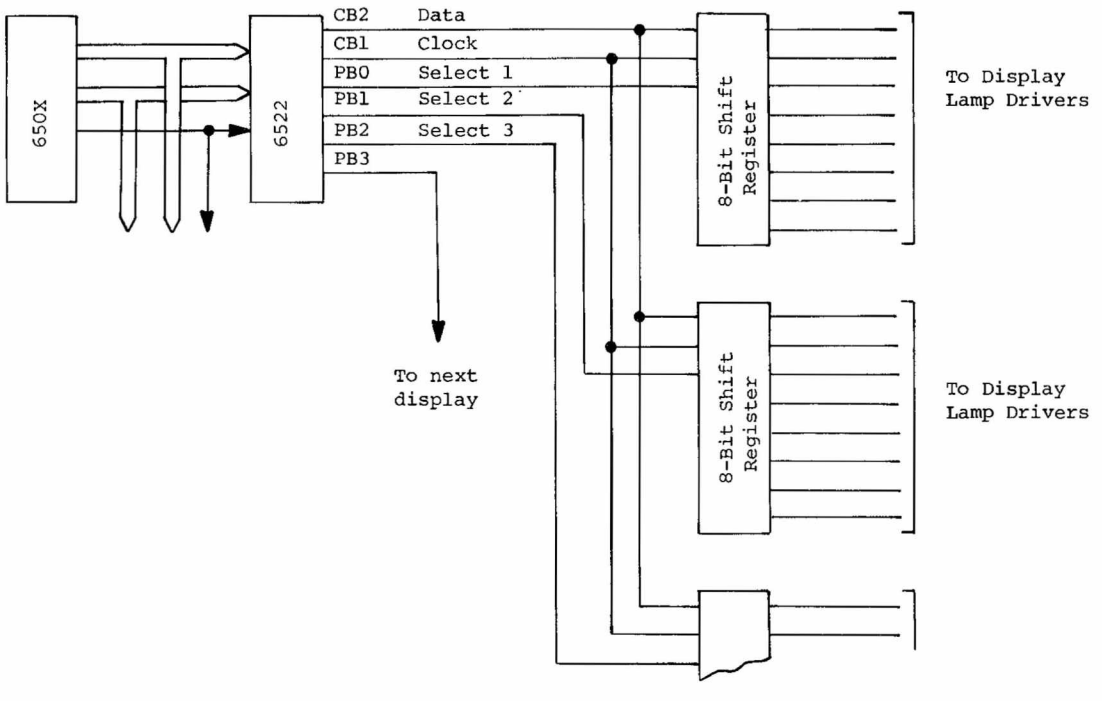


Figure 18. Using the Shift Register for Servicing Remote Status Displays

Remote input devices can be serviced in much the same manner by shifting data into the Shift Register under control of a peripheral port output as shown in Figure 18. Each set of input switches can be polled by first selecting the set to be polled and then triggering the shifting operation with a Shift Register read operation. A shift register interrupt can be used to cause the processor to read the resulting input information after shifting is complete.

The techniques described above can be utilized to expand I/O capability in a microprocessor based system. In a system with many status lamps or many input switches, simple TTL shift registers will provide the necessary I/O in a very cost effective manner. This is illustrated in Figure 19.

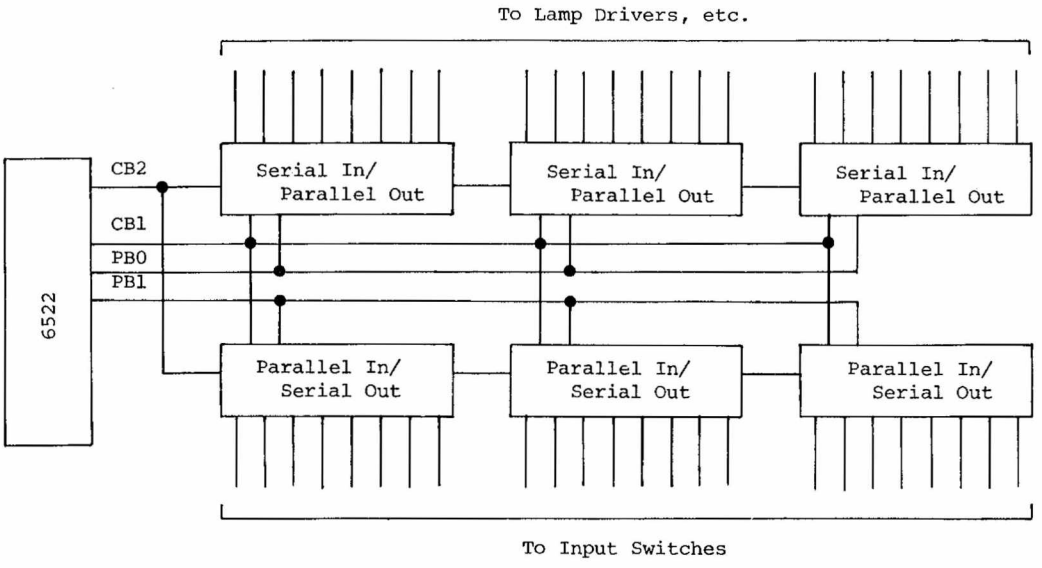
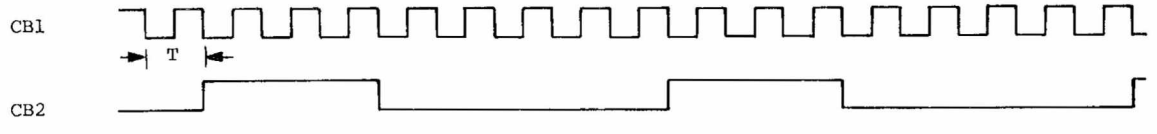


Figure 19. Expanding System I/O Using Shift Register

Clock Generation Using the Shift Register

In all output modes the data shifted out of bit 7 will also be shifted into bit 0. For this reason the Shift Register need not be re-loaded if the same data is to be shifted out each time. A Shift Register read operation can be used to trigger the shifting operation.

This capability is very useful for generating peripheral clocks in the continuous output mode. This mode allows an 8-bit pattern to be shifted out continuously. This is illustrated in Figure 20. Note that in this mode the shifting operation is controlled by Timer 2. A single bit time can therefore be up to 256 clock cycles in length.



Notes:

1. Shift Register loaded with $1110\ 0000_2$ initially.
2. T determined by Timer 2.

Figure 20. Clock Generation Using SR Free-Running Mode

MAXIMUM RATINGS

	Symbol	Value	Unit
Supply Voltage	Vcc	-0.3 to +7.0	Vdc
Input Voltage	Vin	-0.3 to +7.0	Vdc
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages. However, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages.

STATIC D. C. CHARACTERISTICS (Vcc = 5.0 V ± 5%, Vss = 0, T_A = 0 to +70 °C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input high voltage (normal operation)	V_{IH}	+2.4	-	Vcc	Vdc
Input low voltage (normal operation)	V_{IL}	-0.3	-	+0.4	Vdc
Input leakage current- V_{in} = 0 to 5 Vdc $\overline{R/W}$, $\overline{RES}$, RS0, RS1, RS2, RS3, CS1, $\overline{CS2}$, CA1, $\Phi 2$	I_{IN}	-	±1.0	±2.5	µAdc
Off-state input current- V_{in} = .4 to 2.4 V Vcc = Max, D0 to D7	I_{TSI}	-	±2.0	±10	µAdc
Input high current- V_{IH} = 2.4 V PA0 - PA7, CA2, PB0 - PB7, CB1, CB2	I_{IH}	-100	-250	-	µAdc
Input low current- V_{IL} = 0.4 Vdc PA0 - PA7, CA2, PB0 - PB7, CB1, CB2	I_{IL}	-	-1.0	-1.6	mAdc
Output high voltage Vcc = min, I_{load} = -100 µAdc PA0 - PA7, CA2, PB0 - PB7, CB1, CB2	V_{CH}	2.4	-	-	Vdc
Output low voltage Vcc = min, I_{load} = 1.6 mAdc	V_{OL}	-	-	+0.4	Vdc
Output high current (sourcing) V_{OH} = 2.4 V V_{OH} = 1.5 V,PB0 - PB7, CB1, CB2	I_{OH}	 -100 -3.0	 -1000 -5.0	 - -	 µAdc mAdc
Output low current (sinking) V_{OL} = 0.4 Vdc.	I_{OL}	1.6	-	-	mAdc
Output leakage current (off state) $\overline{IRQ}$	I_{off}	-	1.0	10	µAdc
Input Capacitance- T_A = 25 °C, f = 1 Mhz R/W, $\overline{RES}$, RE0, RS1, RS2, RS3, CS1, CS2 D0 - D7, PA0 - PA7, CA2, PB0 - PB7, CB1, CB2 $\Phi 2$ input	C_{in}	 - - -	 - - -	 7.0 10 20	 pF pF pF
Output capacitance- T_A = 25 °C, f = 1 Mhz	C_{out}	-	-	10	pF
Power dissipation	P_d	-		1000	MW

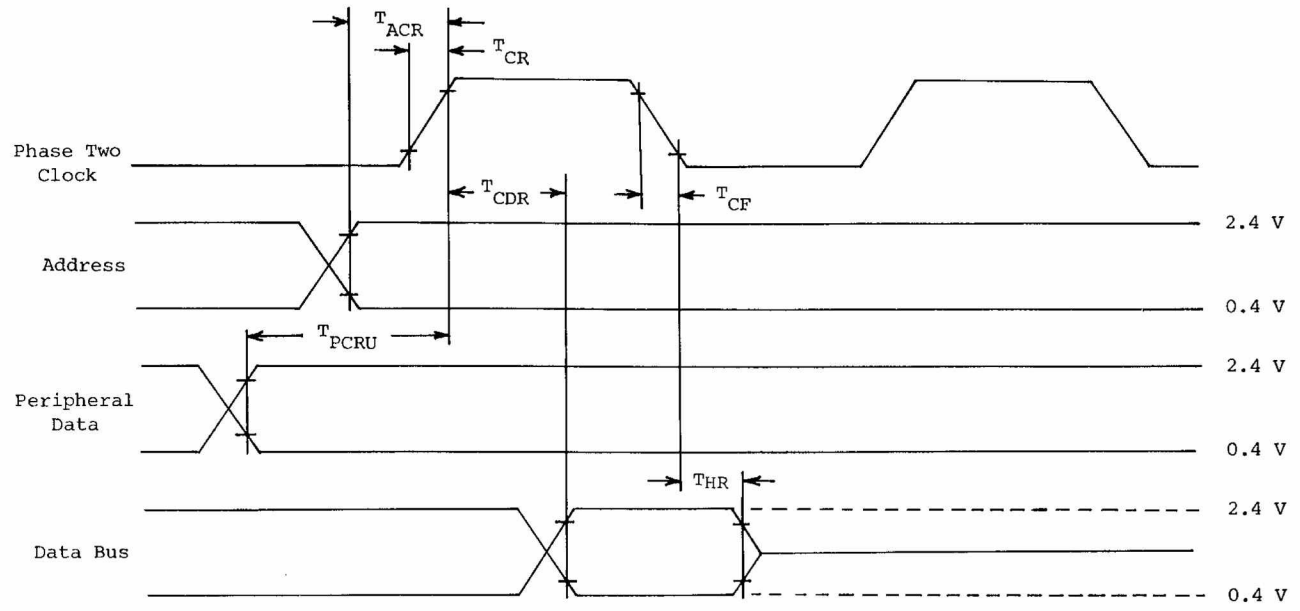


Figure 21. Read Timing Characteristics

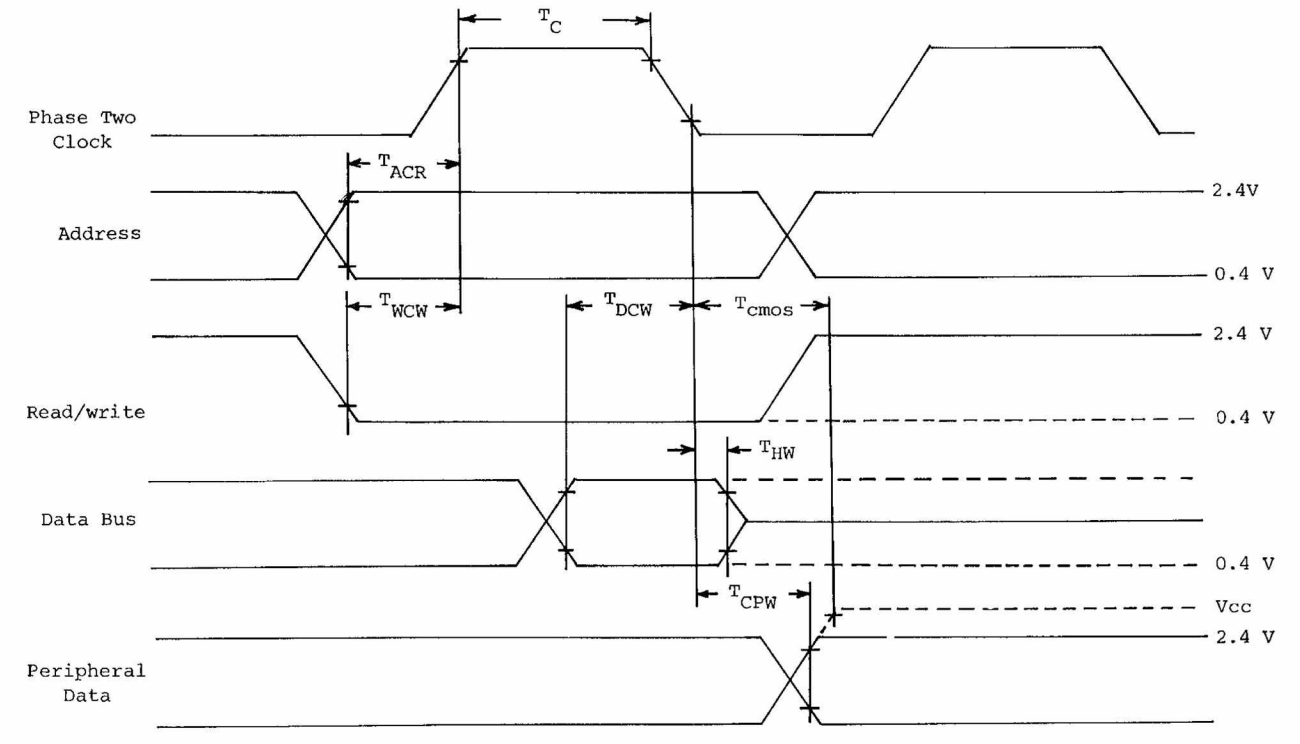


Figure 22. Write Timing Characteristics

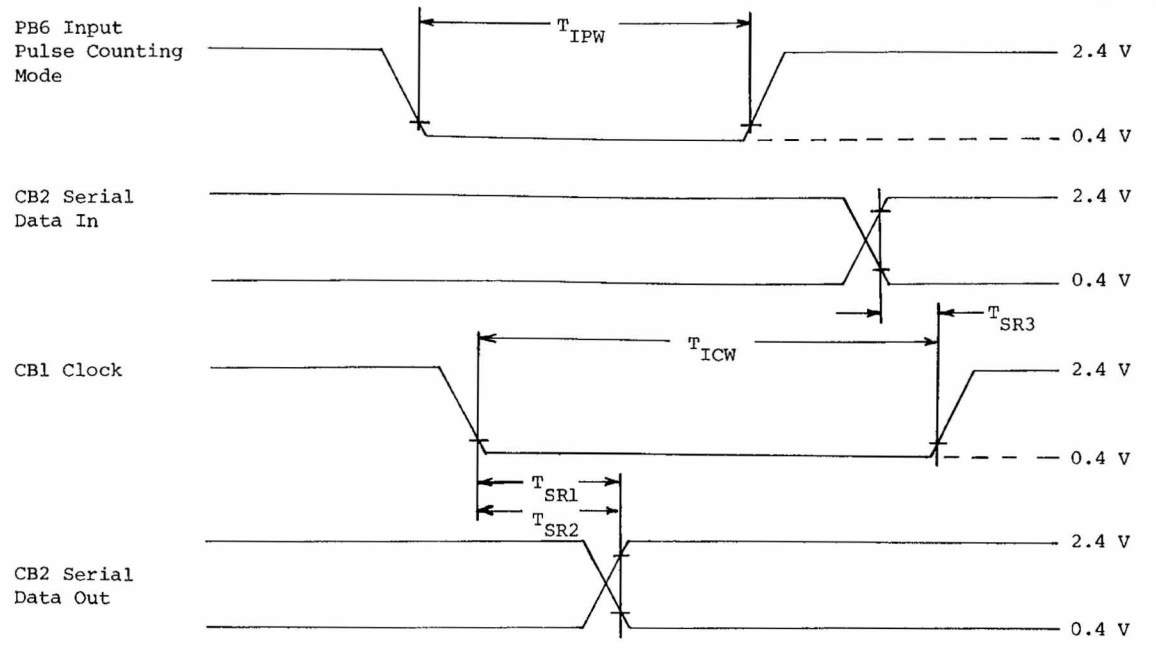


Figure 23. I/O Timing Characteristics

A. C. CHARACTERISTICS

Read Timing Characteristics (Figure 22, loading 130 pF and one TTL load)

Characteristic	Symbol	Min	Typ	Max	Unit
Delay time, address valid to clock positive transition	T_{ACR}	180	-	-	nS
Delay time, clock positive transition to data valid on bus	T_{CDR}	-	-	395	nS
Peripheral data setup time	T_{PCR}	300	-	-	nS
Data bus hold time	T_{HR}	10	-	-	nS
Rise and fall time for clock input	T_{RC} T_{RF}	-	-	25	nS

Write Timing Characteristics (Figure 22)

Characteristic	Symbol	Min	Typ	Max	Unit
Enable pulse width	T_C	0.47	-	25	µS
Delay time, address valid to clock positive transition	T_{ACW}	180	-	-	nS
Delay time, data valid to clock negative transition	T_{DCW}	300	-	-	nS
Delay time, read/write negative transition to clock positive transition	T_{WCW}	180	-	-	nS
Data bus hold time	T_{HW}	10	-	-	nS
Delay time, Enable negative transition to peripheral data valid	T_{CPW}	-	-	1.0	µS
Delay time, clock negative transition to peripheral data valid CMOS (Vcc - 30%)	T_{CMOS}	-	-	2.0	µS

Peripheral Interface Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Rise and fall time for CA1, CB1 CA2 and CB2 input signals.	T_{RF}	-	-	1.0	µS
Delay time, clock negative transition to CA2 negative transition (read handshake or pulse mode).	T_{CA2}	-	-	1.0	µS
Delay time, clock negative transition to CA2 positive transition (pulse mode).	T_{RS1}	-	-	1.0	µS
Delay time, CA1 active transition to CA2 positive transition (handshake mode).	T_{RS2}	-	-	2.0	µS
Delay time, clock positive transition to CA2 or CB2 negative transition (write handshake).	T_{WHS}	-	-	1.0	µS
Delay time, peripheral data valid to CB2 negative transition.	T_{DC}	0	-	1.5	µS
Delay time, clock positive transition to CA2 or CB2 positive transition (pulse mode).	T_{RS3}	-	-	1.0	µS
Delay time, CB1 active transition to CA2 or CB2 positive transition (handshake mode).	T_{RS4}	-	-	2.0	µS
Delay time, peripheral data valid to CA1 or CB1 active transition (input latching).	T_{IL}	300	-	-	nS
Delay time, CB1 negative transition to CB2 data valid (internal SR clock, shift out).	T_{SR1}	-	-	300	nS
Delay time, negative transition of CB1 input clock to CB2 data valid (external clock, shift out).	T_{SR2}	-	-	300	nS
Delay time, CB2 data valid to positive transition of CB1 clock (shift in, internal or external clock)	T_{SR3}	-	-	300	nS
Pulse Width - PB6 Input Pulse	T_{IPW}	2	-	-	µS
Pulse Width - CB1 Input Clock	T_{ICW}	2	-	-	µS
Pulse Spacing - PB6 Input Pulse	I_{IPS}	2	-	-	µS
Pulse Spacing - CB1 Input Pulse	I_{ICS}	2	-	-	µS