TMS 5501 I/O CTRLer

LIST OF ILLUSTRATIONS

LIST OF TABLES

TMS 5501 I/O CONROLLER IN TMS 8080A μP SYSTEMS

The Engineering Staff of
Texas Instruments Incorporated
MOS Houston

I. INTRODUCTION

The TMS 5501 multifunction input/output controller is designed for use in TMS 8080A microprocessor systems. The TMS 5501 performs interrupt processing, parallel data input/output, asynchronous serial data input/output, and multiple event timing. The TMS 5501 can significantly reduce the system package count and cost.

This note describes the general operation of the TMS 5501 and the software necessary for its control. Examples of hardware configurations and software routines are included as a guide for the designer.

The reader is assumed to be familiar with the TMS 8080A architecture and instruction set. The "TMS 8080A Microprocessor Data Book" and the "TMS 5501 Multifunction Input/Output Controller Data Book" provide detailed descriptions of timing, signal sequences, and device interactions.

II. TMS 5501 OPERATION

The TMS 5501 is fabricated with the same N-channel silicon-gate process as the TMS 8080A and has compatible timing, signal levels, and power supply requirements. Figure 1 is a block diagram of the TMS 5501.

The I/O section contains an eight-bit input port and an eight-bit latched output port. The inputs and outputs are TTL compatible.

The timer section contains five programmable interval timers which provide time intervals from 64 microseconds to 16.32 milliseconds with a 2-megahertz system clock.

The communication section contains a universal asynchronous receiver and transmitter (UART) for serial data transfers with user peripherals. The TMS 5501 contains a programmable baud rate generator based on the 2-megahertz system clock for the selection of the serial transfer rate.

The TMS 5501 interrupt control processor provides the TMS 8080A with external interval timer, and communication interface interrupts. The interrupt control processor provides programmable interrupt masking, interrupt level prioritizing, and generation of the interrupt service address. The TMS 5501 and the TMS 8080A data transfers are through the TMS 8080A data bus and are controlled by the control, address, and interrupt lines. The TMS 8080A addresses the TMS 5501 control registers to:

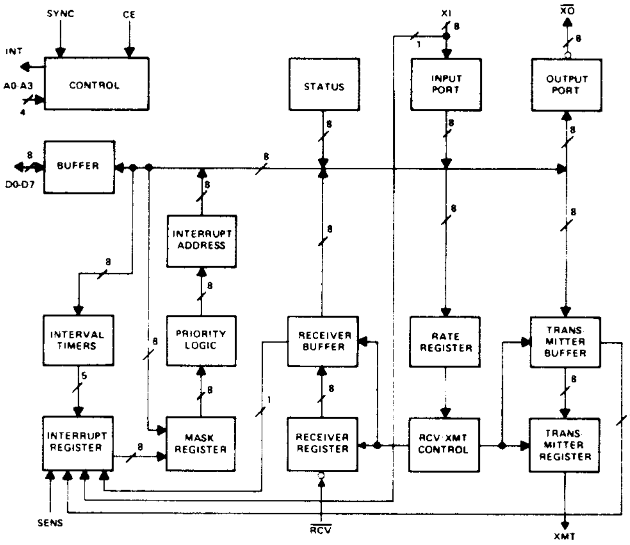


Figure 1. TMS 5501 Block Diagram

- read the receiver buffer
- read the input port
- read the interrupt address
- read the TMS 5501 status
- issue discrete commands
- load the baud rate register
- load the transmitter register
- load the output port
- load the mask register
- load an interval timer.

The commands are generated by executing memory reference instructions (such as MOV) to the addressed register.

A. TMS 5501 INTERFACE SIGNALS

The TMS 5501 has TMS 8080A compatible timing, signal levels, and power supply requirements. The TMS 5501 directly interfaces to the TMS 8080A data bus for status and data transfers as shown in Figure 2.

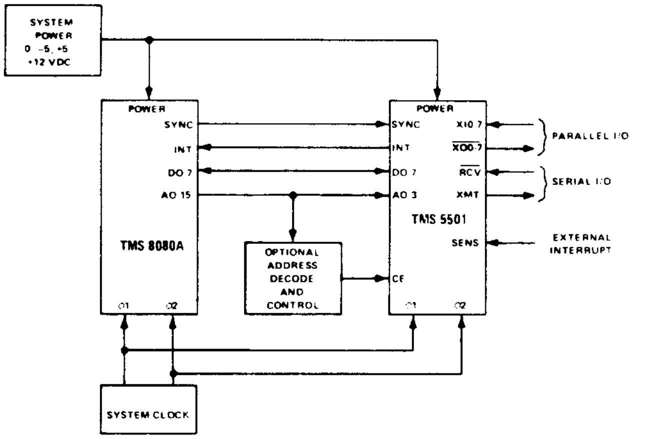


Figure 2. TMS 8080A/TMS 5501 Interface

The TMS 5501 does not use the TMS 8080A DBIN and $\overline{\text{WR}}$ control signals because of pin-out limitations. The TMS 8080A status word is used by the TMS 5501 to control the direction of the data transfers (see SYNC).

The TMS 5501 and the TMS 8080A signal levels are compatible. For systems requiring a small number of memory and I/O circuits, the TMS 8080A can be directly connected to the other circuits such as the TMS 5501. Larger systems may require buffer circuits between the TMS 8080A and the memory and I/O. The additional buffer circuits introduce propagation delays which will affect system timing.

1. Power

The TMS 5501 uses the same power supply voltages as the TMS 8080A (see Table I).

Table I. TMS 5501 Power Supply Requirements

VOLTAGE (V)		TYPICAL SUPPLY CURRENT T_A = 25°C, $t_{c(\phi)}$ = 480 ns
V_{SS}	0 V	Ground
V_{BB}	−5 ± 0.25 V	−0.01 mA
V_{CC}	5 ± 0.25 V	60 mA
V_{DD}	12 ± 0.6 V	40 mA

2. Clocks

The TMS 5501 two-phase clock is identical to the TMS 8080A clock. The clock inputs are not TTL compatible and require high level drivers. Since the clock inputs of the TMS 8080A and the TMS 5501 are driven in parallel, the clock input capacitances for both devices must be summed to determine the AC loading on the clock lines.

The TMS 5501 derives the interval timer clock and the serial communication clock from the input clock (Sections II-C and D).

3. Chip Enable (CE)

When CE is low, the TMS 5501 address decoding is inhibited, which prevents execution of any of the TMS 5501 commands. CE is usually fully or partially decoded from the upper address bits which are not directly input to the TMS 5501. For a minimum system, it may be convenient to connect CE to the most-significant address bit (A15) and to select the system memory only when A15 is low. The TMS 5501 is then selected whenever the upper half of memory is selected. Since the address word is not fully decoded, the TMS 5501 is actually enabled by many unique addresses. If more than one TMS 5501 is used in the system, then the next most-significant bits of the address can be decoded as required. Although this technique limits the size of memory to 32,768 bytes, few applications actually require the full 65,536 byte memory addressing capability of the TMS 8080A.

Since the TMS 5501 is addressed as memory, CE must be low when interrupt acknowledge occurs or improper operation will occur. Since the TMS 8080A acknowledges interrupts during an instruction fetch cycle, this limitation should not affect system operation.

4. SYNC

The SYNC signal is output by the TMS 8080A at the beginning of the machine cycle. While SYNC is high, the TMS 8080A data bus contains the machine status which is used externally to determine the type of machine cycle to occur.

The TMS 5501 monitors the data bus during SYNC for bit D0 (INTA, Interrupt Acknowledge) and bit D1 ($\overline{\text{WO}}$, Read/Write). If D0 is high, then the TMS 8080A is acknowledging an interrupt. If the TMS 5501 has previously received an interrupt acknowledge enable command from the CPU (Section II-E), the RST instruction is transferred by the TMS 5501 to the data bus.

If bit D0 is low and CE is low, then the TMS 5501 will not be affected by the machine cycle. If bit D0 is low and CE is high, then the machine cycle will be either a read or a write data transfer between the TMS 8080A and the TMS 5501.

5. Address Bus (A0 Through A3)

The TMS 5501 operates as a memory device for the TMS 8080A. The TMS 5501 functions are initiated through a four-bit address bus. The TMS 5501, when CE is high, decodes the address bus to determine the command function to be performed.

As shown by Table II, address bits A2 and A3 determine whether the data transfer will be from the TMS 8080A to the TMS 5501 (issue or load) or from the TMS 5501 to the TMS 8080A (read). A2 and A3 are both low for read transfers. An attempt to write data to the TMS 5501 when a read function is addressed will not cause the TMS 5501 to output data. The contents of the TMS 5501 buffers, however, are affected just as if a normal read transfer had occurred. For example, if the TMS 8080A addresses the TMS 5501 read receiver buffer function during a memory write cycle, the TMS 5501 will not output the receiver buffer contents on the data bus, but will clear the receiver buffer loaded flag. An attempt to read data from a write buffer (either A2 or A3 or both are high) will input invalid data both to the TMS 8080A and the addressed TMS 5501 buffer (see Table III).

Table II. Command Address Decoding

OFFSET					WHEN CHIP ENABLE IS HIGH	
(HEX)	A3	A2	A1	A0	COMMAND	FUNCTION
ϕ	L	L	L	L	Read receiver buffer	RBn → Dn
1	L	L	L	H	Read external inputs	XIn → Dn
2	L	L	H	L	Read interrupt address	RST → Dn
3	L	L	H	H	Read TMS 5501 status	(Status) → Dn
4	L	H	L	L	Issue discrete commands	
5	L	H	L	H	Load rate register	
6	L	H	H	L	Load transmitter buffer	Dn → TBn
7	L	H	H	H	Load output port	Dn → $\overline{\text{XO}}$n
8	H	L	L	L	Load mask register	Dn → MRn
9	H	L	L	H	Load interval timer 1	Dn → Timer 1
A	H	L	H	L	Load interval timer 2	Dn → Timer 2
B	H	L	H	H	Load interval timer 3	Dn → Timer 3
C	H	H	L	L	Load interval timer 4	Dn → Timer 4
D	H	H	L	H	Load interval timer 5	Dn → Timer 5
E	H	H	H	L	No function	
F	H	H	H	H	No function	

RBn ≡ Receiver buffer bit n
Dn ≡ Data bus I/O terminal n
XIn ≡ External input terminal n
RST ≡ 11 (IA$_2$) (IA$_1$) (IA$_0$) 1 1 1
TBn ≡ Transmit buffer bit n
$\overline{\text{XO}}$n ≡ Output register bit n
MRn ≡ Mask register bit n

Table III. Read/Write Function Decoding

CE	$\overline{\text{WO}}$*	A3	A2	FUNCTION
L	X	X	X	None
H	H	L	L	Read
H	H	H	X	Invalid Read from Write Buffer
H	H	X	H	
H	L	H	X	Write
H	L	X	H	Write
H	L	L	L	Invalid Write to Read Buffer

X – Don't Care
*Status Bit 1 (D1)

6. External Interrupt (SENS)

The SENS input interfaces an external interrupt to the TMS 5501. A low-to-high transition at the SENS input sets a bit in the TMS 5501 interrupt register which, if enabled, generates an interrupt to the TMS 8080A.

7. Interrupt (INT)

The INT output is high when the TMS 5501 has sensed an enabled interrupt from an interval timer, the UART, or the SENS input. INT can be connected to the TMS 8080A INT input to interrupt the CPU.

8. Receiver Serial Data ($\overline{\text{RCV}}$)

Serial data is input to the TMS 5501 through $\overline{\text{RCV}}$. Between asynchronous input characters, $\overline{\text{RCV}}$ should be held high. A transition from high to low activates the TMS 5501 receiver circuitry which converts the serial input to parallel format, loads the input character into the receiver buffer, and sets the appropriate bits in the status buffer.

9. Transmitter Serial Data (XMT)

Serial data is output by the TMS 5501 through XMT. When the TMS 5501 is not outputting data, XMT is high. A high-to-low transition on XMT following an idle period indicates that the TMS 5501 is outputting a serial character.

10. External Inputs (XI0 Through XI7)

The eight external inputs are gated by the TMS 5501 to the TMS 8080A data bus when the read external inputs function is addressed. The inputs are not complemented; a low (high) external input is gated to the CPU as a low (high) data bit. The TMS 5501 can be programmed to use XI7 as an additional external interrupt. When XI7 is used as an interrupt, interval time number 5 is not used. External pull-up resistors are necessary for many TTL outputs to reach the minimum high-level input voltage of 3.3 V for the TMS 5501.

11. External Outputs ($\overline{\text{XO0}}$ Through $\overline{\text{XO7}}$)

The eight external outputs are driven by the output register. Each output bit is the complement of the bit loaded into the TMS 5501 output register by the TMS 8080A. If the output register is loaded with a low (high) bit, then the corresponding output bit is high (low). The external outputs change only when the load output register function is addressed. The outputs are TTL compatible and are capable of driving a standard TTL load.

B. INTERRUPT PROCESSING OPERATION

The TMS 5501 generates and stores a total of eight interrupts. Four of the interrupts originate with four interval timers, two originate with the serial transmitter and receiver, one originates from the external input (SENS), and the final interrupt originates either from the fifth interval timer or a second external interrupt (XI7).

The interrupts from the interval timers are generated at the end of the programmed time interval (Section II-C). The interrupt from the serial transmitter is generated when the TMS 5501 has finished sending a character and is ready to accept a new character. The interrupt from the serial receiver is generated when a character has been input (Section II-D). The external interrupts are generated by a low-to-high transition on the external inputs (SENS, XI7). The TMS 5501 is programmed to select either the fifth timer interrupt or the second external interrupt (Section II-E).

1. Interrupt Register

The TMS 5501 interrupt register stores each interrupt until the interrupt has been serviced. A discrete reset command from the TMS 8080A resets all interrupts except the serial transmitter interrupt which is set (Section II-E). The interrupt register is eight bits long with one bit for each interrupt. A stored interrupt is a high bit.

2. Mask Register

The TMS 5501 interrupt mask register inhibits undesired interrupts. The mask register contains a separate bit for each of the eight interrupts and is loaded by the TMS 8080A (Section II-E). The contents of the interrupt register and the mask register are ANDed and then input to the TMS 5501 priority gating logic. Thus, a low bit in the mask register will inhibit the corresponding interrupt, while a high bit will enable the interrupt.

An interrupt is stored in the interrupt register independently of the state of the corresponding bit in the mask register. If the interrupt is generated while the mask bit is low, the interrupt is stored but not enabled. If the interrupt is still stored (no reset command) when the mask bit is set high, then the newly enabled interrupt is input to the priority gating logic.

3. Priority Gating

The TMS 5501 prioritizes the enabled interrupts in the following order:

1st	–	Interval Timer 1
2nd	–	Interval Timer 2
3rd	–	External Sensor
4th	–	Interval Timer 3
5th	–	Receiver Buffer Loaded
6th	–	Transmitter Buffer Emptied
7th	–	Interval Timer 4
8th	–	Interval Timer 5 or External Input (XI 7)

The TMS 5501 interrupt address is then generated for the highest priority enabled interrupt (Section II-E). Interrupt masking and priority gating are shown in Figure 3.

4. Interrupt Servicing

The TMS 5501 provides two methods of servicing interrupts: interrupt-driven or polled. In the interrupt-driven system, the TMS 5501 INT output is connected directly to the TMS 8080A INT input. In the polled system, the TMS 8080A must read the TMS 5501 status to determine the status of the TMS 5501 interrupt line (Section III-A). A multiple TMS 5501 system could employ both of these methods (Section III-B).

In the interrupt-driven system, the TMS 5501 INT output goes high when an enabled interrupt is generated. The INT output stays high until all enabled interrupts have been serviced. If the TMS 8080A is enabled to accept the interrupt, it sets the INTA (interrupt acknowledge) status bit (D0) high during SYNC of the next instruction fetch machine cycle. If the TMS 5501 has previously been enabled to respond to INTA by a discrete command function (Section II-E), it will jam the RST instruction onto the data bus. The RST byte is generated by the TMS 5501 priority logic. The highest priority enabled high interrupt bit in the interrupt register is then reset. If the TMS 5501 is not enabled to respond to INTA, it does not output the RST instruction, and the interrupt register is not affected. RST instructions are listed in Table IV.

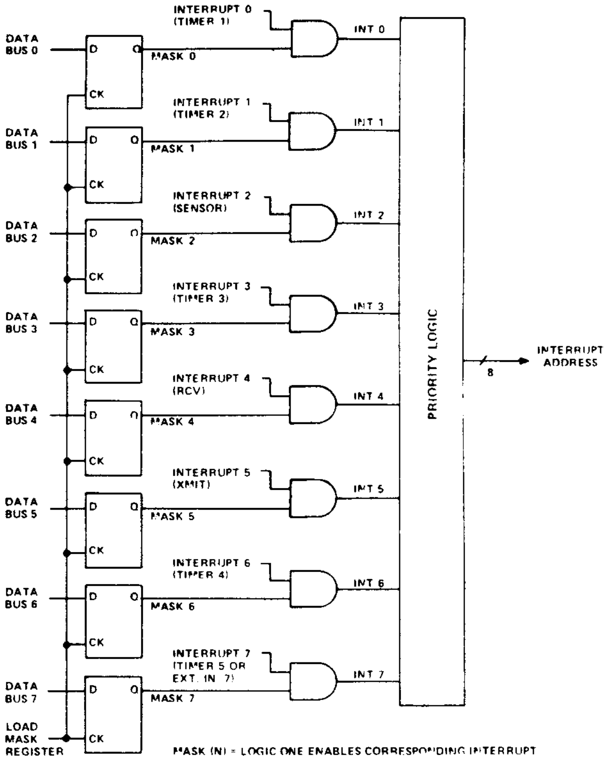


Figure 3. I/O Interrupt Masking and Priority

The polled interrupt system is described in Section III-A.

Table IV. RST Instructions

DATA BUS BIT 0	1	2	3	4	5	6	7	INTERRUPT CAUSED BY
H	H	H	L	L	L	H	H	Interval Timer 1
H	H	H	H	L	L	H	H	Interval Timer 2
H	H	H	L	H	L	H	H	External Sensor
H	H	H	H	H	L	H	H	Interval Timer 3
H	H	H	L	L	H	H	H	Receiver Buffer
H	H	H	H	L	H	H	H	Transmitter Buffer
H	H	H	L	H	H	H	H	Interval Timer 4
H	H	H	H	H	H	H	H	Interval Timer 5 or X17

C. INTERVAL TIMER OPERATION

The TMS 5501 has five interval timers, each of which can be addressed by the TMS 8080A. The timers are eight-bits long and can be programmed for intervals from 64 microseconds to 16.32 milliseconds.

The TMS 8080A loads an interval counter by addressing one of the five load interval timer command functions and outputting an eight-bit count value (0 to 255). The counter begins the count interval immediately after it is loaded. At the end of the interval, the corresponding bit in the interrupt register is set, and the counter is inhibited until a new count is loaded.

The count clock for the interval timers is derived by dividing the TMS 5501 input clock by 128. For the nominal 2 MHz input clock, the clock resolution is 64 microseconds. The total count interval is then the product of the loaded count and the count resolution. Thus, the maximum count interval is 255 X 64 microseconds = 16.32 milliseconds.

Since the count clock and the loading of the interval count are asynchronous, the accuracy of the timed interval is plus 0, minus 64 microseconds.

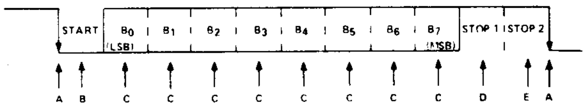
Loading of a zero count generates an immediate interrupt and inhibits the counter. Loading a new count while the timer is still counting a previous interval nullifies the old count and initiates a new interval.

D. SERIAL COMMUNICATION OPERATION

The TMS 5501 contains a UART for communicating with user peripherals. The TMS 5501 generates a common receive/transmit clock from the 2-megahertz system clock. The UART provides serial-to-parallel/parallel-to-serial data format conversion, buffers transmit and receive data, and generates interrupts to synchronize the CPU with the serial data communication link.

1. Serial Data Receiver

The function of the serial data receiver is to synchronize with the incoming serial data at the selected baud rate and to provide bit-to-bit timing for input to the 8-bit character buffer. The serial data receiver has a resettable counter and decoding logic for determining half-bit and full-bit times as shown in Figure 4. When a high-to-low transition is detected on the serial input data line and the counter is in the search (start bit) mode, the counter is incremented until the middle of the start bit is reached. If the serial input data line should return to a high during this interval, the counter resets and starts over again. After the start bit is detected, the counter waits full-bit times to determine the center of all succeeding bits.



A - HIGH TO LOW TRANSITION IS DETECTED AND HALF BIT COUNTER IS ENABLED.

B - CENTER OF START BIT FOUND AND FULL BIT COUNTER IS ENABLED.

C - CENTER OF EACH DATA BIT FOUND AND SHIFTED INTO THE SERIAL RECEIVE REGISTER.

D - FOR 10-BIT CODE, CHECK FOR VALID STOP BIT GENERATE INTERRUPT, AND TRANSFER CHARACTER TO RECEIVE BUFFER REGISTER.

D - FOR 11 BIT CODE, CHECK FOR TWO VALID STOP BITS GENERATE INTERRUPT, AND TRANSFER CHARACTER TO RECEIVE BUFFER REGISTER.

		COUNTS AT 125 kHz		
BAUD RATE	BIT PERIOD (ms)	HALF-BIT	FULL-BIT	RESOLUTION
110	9.08	568	1136	0.09%
150	6.67	416	832	0.13%
300	3.33	208	416	0.25%
1200	0.833	52	104	1%
2400	0.416	26	52	2%
4800	0.208	13	26	4%
9600	0.104	7	13	8%

NOTE: BAUD RATES, BIT PERIODS, AND CORRESPONDING RESOLUTIONS ARE VALID FOR A 2 MHz INPUT CLOCK (φ1, φ2).

Figure 4. Receive Character Timing

When the middle of the start bit is detected, the start bit detected flag is set. When the middle of the first data bit is detected, the full bit detected flag is set. Both of these flags are reset when the (last) stop bit is detected. These flags may be read as I/O status and are intended for test purposes.

When a character has been received, it is transferred from the receiver (serial) register to the receive buffer register from which it is gated to the data bus when the read receive buffer function is addressed by the TMS 8080A.

When the receive buffer register is loaded, the receive buffer loaded flag is set, indicating that a new received character is ready, and an interrupt is generated. This flag is available as a status bit and is reset when the read receive buffer function is addressed.

If a new character is received before the previous character is read, the previous character is lost and the overrun flag is set. The overrun flag may be read as I/O status and is reset when the read TMS 5501 status function is addressed.

When the last stop bit of a character is received, a check is made for the validity of the stop bit(s). If one (or both) of the stop bits is incorrect, the framing error flag is set. If the stop bit(s) is (are) correct, the framing error flag is reset. The framing error flag may be read as I/O status.

Figure 5 is a flow chart describing the serial data receiver.

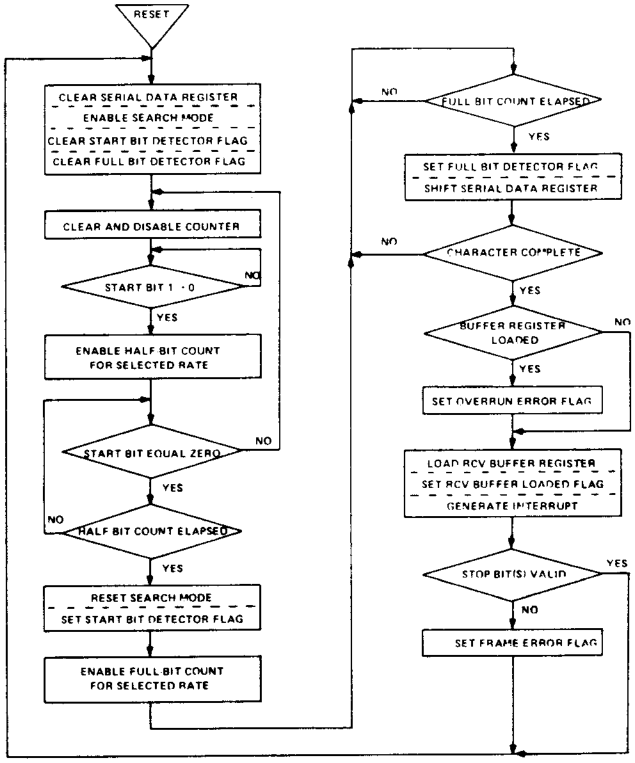


Figure 5. Receiver Flow Chart

2. Serial Data Transmitter

The function of the serial data transmitter is to accept a character from data bus and transmit the character serially at the selected baud rate. Figure 6 shows the data formats for 10-bit and 11-bit codes.

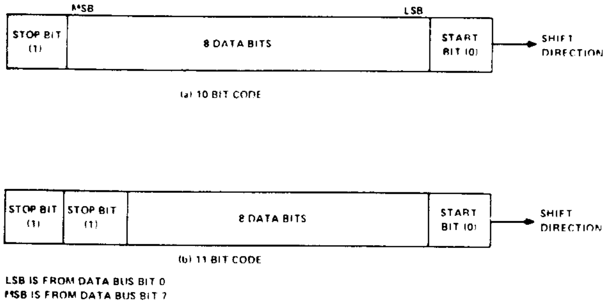


Figure 6. Transmit Data Format

Data transmission is handled automatically once a character from the TMS 8080A has been loaded into the transmit buffer register. If the transmitter is not currently busy, the character in the transmit buffer register is transferred to the serial transmit register and shifted out at the selected baud rate. A start bit is generated before the data character is shifted out, and one or two stop bits (as selected) are generated after the last bit of the data character is shifted out.

Once the transmit buffer register is emptied (contents transferred to the transmit register and transmission is initiated), the buffer register is again ready to be loaded and an interrupt is generated (if enabled). If the buffer register is loaded while a character transmission is in progress, then the new character is simply held in the buffer register until the current transmission is completed.

The transmit buffer empty flag is available as a status bit. A high bit indicates that the buffer is empty (not loaded) and that a new character may be loaded. This flag is reset when the load transmit buffer function is addressed. The flag is set when a character to be transmitted is transferred from the buffer register to the transmit register or when a RESET discrete command is issued by the TMS 8080A.

When a transmission is completed, the transmit data output remains high until the next transmission is initiated.

3. UART Accuracy

Distortion in the UART receiver is the difference between the half-bit sample point and the ideal center of the bit. Distortion in the UART transmitter is the difference between the actual and the ideal center of each transmitted bit relative to the ideal leading edge of the start bit. The transmitter distortion, the receiver distortion, and the transmission link distortion can combine under many circumstances to cause the loss of bit synchronization during data reception.

There are four basic causes for the TMS 5501 receiver distortion.

1. The receiver tests for the leading edge of the start bit using a 125-kilohertz sample clock derived from the 2-megahertz system clock asynchronous to the ideal sample period. The leading edge of the start bit can be detected as much as one clock period or 8 microseconds after it actually occurs. Since the receiver begins the half-bit count at this point, as shown in Figure 5, the maximum initial distortion is 8 microseconds.
2. The half-bit time using a standard baud rate may not be an exact multiple of the sample clock. The nearest multiple is used and the difference generates the half-bit distortion.
3. The full-bit time may also not be an exact multiple of the sample clock and results in the full-bit distortion term. This term is cumulative for each of the nonstart bits in the character.
4. If the system clock frequency is not exactly 2 megahertz, the receiver half-bit and full-bit times can be larger or smaller than expected.

The maximum receiver distortion is approximated by summing the distortion terms as shown in Table V. The "maximum distortion accepted" term indicates the amount of distortion that can be generated by the transmission link and transmitter without loss of bit synchronization.

The transmitter distortion is similarly produced. Start-bit generation is asynchronous to the sample clock and the leading edge can be transmitted 8 microseconds later than expected. The bit times are again not exact multiples of the internal sample clock, and the approximations generate bit distortions. Finally, the system clock can also drift from 2 megahertz as before. The transmitter distortion terms are summed as shown in Table VI.

The transmission link distortion is dependent upon the specific application and is normally measured experimentally. The sum of the distortion terms should not exceed 50% of a bit time to avoid false bit sampling.

E. FUNCTION ADDRESSING

The fourteen command functions are used to control the TMS 5501. The TMS 8080A can issue discrete commands to the TMS 5501, select the serial data transfer rate, load and start the interval timers, load the mask register, load the parallel output byte, load the serial output byte, read the parallel input byte, read the serial input byte, read the I/O status, or read the interrupt service address. The serial data transmitter flow chart is shown in Figure 7.

The TMS 8080A executes a memory read or write instruction to a TMS 5501 address to perform each of these functions. The effective address of each of the TMS 5501 command functions is derived by adding the base address of the TMS 5501 to the offset address (value of A0 through A3 in Table II). For example, a possible TMS 5501 base address is 8000 (HEX). Then the effective address of the load output port function is 8000 + 0007 = 8007 (HEX). If A15 alone is used to select the TMS 5501, the base address is actually any multiple of 16 from 8000 to FFF0 (HEX).

Table V. TMS 5501 Receiver Accuracy

Baud Rate	Bit (1/2 Bit) Period (μs)	8 μs Counts Half Bit	 Full Bit	TMS 5501 Bit Timing Count Time (μs) Half Bit Error	 Full Bit Error	Max. TMS 5501 Error (μs) (% Bit)	0.05% Crystal Error (μs) (% Bit)	Max. Total Error (μs) (% Bit)	Maximum Distortion Accepted (%)
110 (2 stop bits)	9091.0 (4545.5)	568	1136	4544 (Δ = −1.5)	9088 (Δ = −3.0)	−39.5 (0.435%)	−47.7 (0.525%)	−87.2 (0.96%)	49.0
150	6666.7 (3333.3)	416	832	3328 (Δ = −5.3)	6656 (Δ = −10.7)	−109.6 (1.64%)	−31.6 (0.475%)	−141.2 (2.12%)	47.8
300	3333.3 (1666.7)	208	416	1664 (Δ = −2.7)	3328 (Δ = −5.3)	−58.4 (1.75%)	−15.8 (0.475%)	−74.2 (2.23%)	47.7
1200	833.3 (416.7)	52	104	416 (Δ = −0.7)	832 (Δ = −1.3)	−20.4 (2.45%)	−3.96 (0.475%)	−24.4 (2.93%)	47.0
2400	416.7 (208.4)	26	52	208 (Δ = −0.4)	416 (Δ = −0.7)	−14.7 (3.53%)	−1.98 (0.475%)	−16.7 (4.02%)	46.0
4800	208.4 (104.2)	13	26	104 (Δ = −0.2)	208 (Δ = −0.4)	−11.8 (5.67%)	−0.988 (0.475%)	−12.8 (6.14%)	43.8
9600	104.2 (52.1)	7	13	56 (Δ = +3.9)	104 (Δ = −0.2)	−5.9 (5.67%)	−0.496 (0.48%)	−6.40 (6.14%)	43.8

Note: TMS 5501 error consists of:

1. Initial 8 μs error (asynchronous clock)
2. Half-bit count error
3. Full-bit count error

For overall error calculation, a 0.05% crystal is utilized.

Table VI. TMS 5501 Transmitter Accuracy

Baud Rate	Bit Period (μs)	8 μs Counts	TMS 5501 Count Time (μs) Error	Max. TMS 5501 Error (μs) TMS 5501 Distortion	0.05% Crystal Error (μs) Clock Distortion	Max. Total Error (μs) Total XMT Distortion
110 (2 stop bits)	9091.0	1136	9088 (Δ = −3.0)	−41 (0.45%)	−50 (0.55%)	−91 (1.0%)
150	6666.7	833	6664 (Δ = −2.7)	−35 (0.53%)	−33.32 (0.5%)	−68.32 (1.03%)
300	3333.3	417	3336 (Δ = +2.7)	+27 (0.81%)	+16.68 (0.5%)	+43.68 (1.31%)
1200	833.3	104	832 (Δ = −1.3)	−21 (2.52%)	−4.16 (0.5%)	−25.16 (2.54%)
2400	416.7	52	416 (Δ = −0.7)	−15 (3.6%)	−2.08 (0.5%)	−17.08 (4.1%)
4800	208.4	26	208 (Δ = −0.4)	−12 (5.76%)	−1.04 (0.5%)	−13.04 (6.26%)
9600	104.2	13	104 (Δ = −0.2)	−10 (9.6%)	−0.52 (0.5%)	−10.52 (10.1%)

Note: TMS 5501 error consists of:

1. Initial 8 μs error (asynchronous clock)
2. Full-bit count error

For overall error calculation, a 0.05% crystal is utilized.

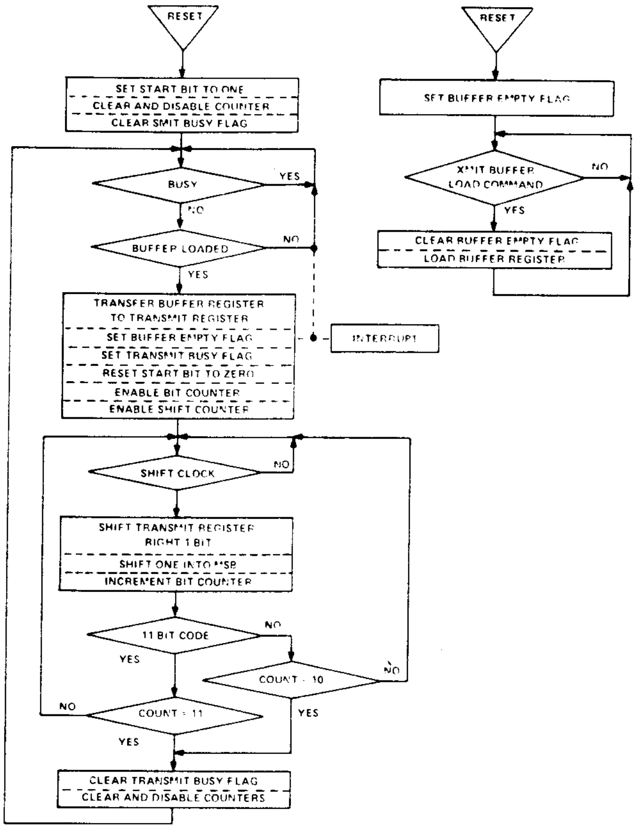


Figure 7. Serial Data Transmitter Flow Chart

The redundancy of addresses is the result of not fully decoding the address. It is convenient, however, to select an arbitrary address such as 8000 (HEX) as the base address.

The TMS 8080A uses memory reference instructions to address and to transfer data to the command functions. For example, the following instruction sequence forces the TMS 5501 output bit $\overline{XO7}$ to be high and bits $\overline{XO0\text{-}6}$ to be low. The routine then reads the input port into the accumulator.

```
    :
MVI     A, 7FH
STA     8007H
LDA     8001H
    :
```

The TMS 5501 output bits are the complements of the bits loaded into the output port. Therefore, when 7F (HEX) is loaded into the output port, 80 (HEX) is actually output by the TMS 5501 on $\overline{XO0\text{-}7}$. The new value in the accumulator is equal to the input data port value since the input port bits are not complemented by the TMS 5501.

The following descriptions of the TMS 5501 command functions assume that TMS 8080A read instructions are used to address the read functions and write instructions are used to address issue or load functions.

1. Read Receiver Buffer — Offset ϕ

Addressing the read receiver buffer function gates the contents of the serial receive buffer onto the data bus and clears the receiver buffer loaded flag in the status buffer.

2. Read External Inputs — Offset 1

Addressing the read external inputs function gates the true values of the eight external input lines to the data bus. No status bits are affected.

3. Read Interrupt Address — Offset 2

Addressing the read interrupt address function gates to the data bus the encoded address of the highest priority interrupt currently pending. The corresponding bit in the interrupt register is reset following completion of the read operation.

If no interrupt is pending when this function is addressed, a false interrupt address will be read. The false interrupt address is identical to the interrupt 7 encoded address.

Addressing this function produces the same action as does the TMS 8080A interrupt acknowledge except that the function is independent of the interrupt acknowledge enable and the RST op code is placed on the data bus as an operand rather than as an instruction.

4. Read TMS 5501 Status — Offset 3

Addressing the read TMS 5501 status function gates the various status conditions of the TMS 5501 onto the data bus. The status conditions, available as indicated in Figure 8, are described in the following paragraphs.

BIT	7	6	5	4	3	2	1	0
	START BIT DETECT	FULL BIT DETECT	INTRPT PENDING	XMIT BUFFER EMPTY	RCV BUFFER LOADED	SERIAL RCVD	OVERRUN ERROR	FRAME ERROR

Figure 8. Data Bus Assignments for TMS 5501 Status

Bit 0, Framing Error

A high in bit 0 indicates that a framing error was detected on the last character received (either one or both stop bits were in error). The framing error flag is updated at the end of each character. Bit 0 of the TMS 5501 status will remain high until the next valid character is received.

Bit 1, Overrun Error

A high in bit 1 indicates that a new character was loaded into the receiver buffer before a previous character was read out. The overrun error flag is cleared each time the read TMS 5501 I/O status function is addressed or a reset command is issued.

Bit 2, Serial Received Data

Bit 2 monitors the receiver serial data input line. This line is provided as a status input for use in detecting a break and for test purposes. Bit 2 is normally high when no data is being received.

Bit 3, Receiver Buffer Loaded

A high in bit 3 indicates that the receiver buffer is loaded with a new character. The receiver buffer loaded flag remains high until the read receiver buffer function is addressed, at which time the flag is cleared. The reset function also clears this flag.

Bit 4, Transmitter Buffer Empty

A high in bit 4 indicates that the transmitter buffer register is empty and ready to accept a character. Note, however, that the serial transmitter register may be in the process of shifting out a character. The reset function sets the transmitter buffer empty flag high.

Bit 5, Interrupt Pending

A high in bit 5 indicates that one or more of the interrupt conditions has occurred and the corresponding interrupt is enabled. This bit is the status of the interrupt signal INT.

Bit 6, Full Bit Detected

A high in bit 6 indicates that the first data bit of a receive data character has been detected. This bit remains high until the entire character has been received or until a reset is issued and is provided for test purposes.

Bit 7, Start Bit Detected

A high in bit 7 indicates that the start bit of an incoming data character has been detected. This bit remains high until the entire character has been received or until a reset is issued and is provided for test purposes.

5. Issue Discrete Commands Offset 4

Addressing the discrete command function causes the TMS 5501 to interpret the data bus information according to the following descriptions. See Figure 9 for the discrete command format. Bits 1 through 5 are latched until a different discrete command is received.

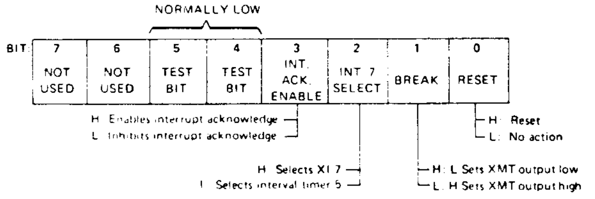


Figure 9. Discrete Command Format

Bit 0, Reset

A high in bit 0 will cause the following:

1. The receiver buffer and register are cleared to the search mode including the receiver buffer loaded flag, the start bit detected flag, the full bit detected flag, and the overrun error flag. The receiver buffer is not cleared and will contain the last character received.
2. The transmitter data output is set high (marking). The transmitter buffer empty flag is set high indicating that the transmitter buffer is ready to accept a character from the TMS 8080A.
3. The interrupt register is cleared except for the bit corresponding to the transmitter buffer interrupt, which is set high.
4. The interval timers are inhibited.

A low in bit 0 causes no action. The reset function has no affect on the output port, the external inputs, interrupt acknowledge enable, the mask register, the rate register, the transmitter register, or the transmitter buffer.

Bit 1, Break

A low in bit 1 causes the transmitter data output to be reset low (spacing).

If bit 0 and bit 1 are both high, the reset function will override.

Bit 2, Interrupt 7 Select

Interrupt 7 may be generated either by a low-to-high transition of external input 7 or by interval timer 5.

A high in bit 2 selects the interrupt 7 source to be the transition of external input 7. A low in bit 2 selects the interrupt 7 source to be interval timer 5.

Bit 3, Interrupt Acknowledge Enable

The TMS 5501 decodes data bus (CPU status) bit 0 at SYNC of each machine cycle to determine if an interrupt acknowledge is being issued.

A high in bit 3 enables the TMS 5501 to accept the interrupt acknowledge decode. A low in bit 3 causes the TMS 5501 to ignore the interrupt acknowledge decode.

Bit 4, Test Bit

When test bit 4 is low, the interval timer clock and serial communication clock operate as described in Sections II-C and D. A high in bit 4 causes both clocks to run eight times faster than normal. Bit 4 should be low for normal operation. When bit 4 is high the TMS 5501 can be used for high-speed-data transfers at rates up to 76.8 kilobaud.

Bit 5, Test Bit

Test bit 5 should be low for proper operation. When test bit 5 is high, the TMS 5501 INT output is not the output of the interrupt priority logic but the buffered output of the logic which derives the interval timer and serial communication clocks from the TMS 5501 input clock. Therefore, when bit 5 is high, the INT output is a clock with a frequency equal to the system clock frequency divided by 128 (test bit 4 low) or by 16 (test bit 4 high). If the TMS 5501 INT is not needed as an interrupt, then a high bit 5 permits the INT output to be used to derive a TTL compatible clock for the system.

6. Load Rate Register Offset 5

Addressing the load rate register function causes the TMS 5501 to load the rate register from the data bus and interpret the data bits (see Figure 10) as follows.

BIT:	7	6	5	4	3	2	1	0
	STOP BIT(s)	9600 baud	4800 baud	2400 baud	1200 baud	300 baud	150 baud	110 baud

H: One stop bit
L: Two stop bits

Figure 10. Data Bus Assignments for Rate Commands

Bits 0 Through 6, Rate Select

The rate select bits (bits 0 through 6) are mutually exclusive, i.e., only one bit may be high. A high in bits 0 through 6 will select the baud rate for both the transmitter and receiver circuitry as defined below and in Figure 10:

Bit 0	110 baud
Bit 1	150 baud
Bit 2	300 baud
Bit 3	1200 baud
Bit 4	2400 baud
Bit 5	4800 baud
Bit 6	9600 baud

If more than one bit is high, the highest rate indicated will result. If bits 0 through 6 are all low, both the receiver and the transmitter circuitry will be inhibited. The nominal baud rates are valid for a 2-megahertz system clock. A slower system clock will cause proportionally slower baud rates.

Bit 7, Stop Bits

Bit 7 determines whether one or two stop bits are to be used by both the transmitter and receiver circuitry. A high in bit 7 selects one stop bit. A low in bit 7 selects two stop bits.

7. Load-Transmitter Buffer — Offset 6

Addressing the load transmitter buffer function transfers the contents of the data bus into the transmit buffer.

8. Load Output Port — Offset 7

Addressing the load output port function transfers the contents of the data bus to the output port. The data is latched and remains on $\overline{\text{XO 0-7}}$ as the complement of the load data until the load output port function is addressed again.

9. Load Mask Register — Offset 8

Addressing the load mask register function transfers the contents of the data bus to the interrupt mask register.

10. Load Timer — Offsets 9-13

Addressing the load timer functions transfers the contents of the data bus to the appropriate timer and initiates the count interval.

III. USING THE TMS 5501

The TMS 5501 is a powerful design tool because of the simplicity of its interface to the TMS 8080A and because of its multifunction operation.

This section contains examples of interrupt processing, serial data communications, and design precautions for the TMS 5501.

A. INITIALIZATION

The TMS 5501 does not have an external reset input and must be initialized under program control. Moreover, the TMS 8080A reset routine and the interrupt service routine for the level 0 interrupt share the same memory location. Because of this ambiguity, the service routine starting at address 0 must determine whether a reset or an interrupt has occurred if the TMS 5501 timer 1 interrupt is used.

Since both a reset and a TMS 5501 level 0 interrupt can occur concurrently, it is not sufficient to check for the presence or absence of the interrupt. The service routine should check for the occurrence of the reset. If a reset has occurred, the TMS 5501 (and the interrupt) will normally be reset by the initialization routine. If a reset has not occurred, the interrupt service routine is then entered.

The reset signal can be stored externally in a flip-flop or latch which can be tested and cleared under program control. The reset signal can also be connected to the TMS 5501 SENS or XI7 inputs. The reset then generates an interrupt. If the reset and the timer 1 interrupt occur simultaneously, then the timer 1 interrupt has precedence. The service routine can then selectively enable the TMS 5501 interrupts to determine if a reset has occurred.

The application requirements must dictate which approach is used. If timer 1 is not required, it can be disabled. Otherwise, the designer should weigh the cost of the added storage element against the use of an interrupt input for the reset. The speed and complexity of the service routine must also be considered.

Once the TMS 8080A reset service routine is entered, the TMS 5501 is initialized by addressing the issue discrete commands function with a high reset bit. The TMS 5501 registers should then be loaded with the required initialization values.

The routine in Figure 11 is an example of a TMS 5501 initialization routine. The TMS 5501 has a base address of 8000 (HEX). The output port is cleared, the timer interrupts are enabled, the 300 baud rate is selected, and interrupt acknowledge is enabled.

B. INTERRUPT SERVICE

The TMS 8080A has a special one-byte call for interrupt processing. The restart (RST) instruction stores the program counter on the stack and transfers control to one of eight locations. A three-bit field in the RST opcode specifies which location is used. The eight locations are 0, 8, 10, 18, 20, 28, 30, and 38 (HEX). Each location and the seven succeeding bytes are normally used for the reset or an interrupt service routine. If a service routine requires more than eight bytes, then a branch statement can be used to transfer control to another part of program memory.

The TMS 5501 provides two methods of servicing interrupts: interrupt-driven and polled (Section II-B). In an interrupt-driven system, the TMS 5501 generates the

```
          ORG   0
          JMP   IORST
           .
           .
IORST:    LXI   H, 8004H      ; SET UP M POINTER
          MVI   A, 01H        , ISSUE DISCRETE
          MOV   M, A          ; RESET COMMAND
          INX   H
          MVI   A, 04H        ; SELECT 300 BAUD
          MOV   M, A
          INX   H
          INX   H
          MVI   A, 0FFH       ; CLEAR OUTPUT PORT
          MOV   M, A
          INX
          MVI   A, 0D3H       ; ENABLE TIMER INTERRUPTS
          MOV   M, A
          MVI   A, 08H        ; ENABLE INTERRUPT ACKNOWLEDGE
          STA   8004H
```

Figure 11. Sample TMS 5501 Initialization Routine

interrupt and inserts the RST op code on the data bus when the interrupt is acknowledged by the TMS 8080A. Since the TMS 5501 prioritizes interrupts and provides the RST op code, the TMS 8080A quickly begins the service routine of the highest priority interrupt. The routine in Figure 12 is an example of an interrupt service routine for timer 2. The routine increments a memory location (COUNT) and reloads the counter for a new interval. The TMS 5501 base address is 8000 (HEX). The programmed interval is equal to 179 X 64 microseconds = 11.46 milliseconds.

```
ADDRESS (HEX)

0–7       ; RESET/TIMER 1 SERVICE ROUTINE
8         TIME 2:  PUSH    PSW        ; SAVE ACC + STATUS ON STACK
9                  LDA     COUNT      ; INCREMENT COUNT
12                 INR     A
13                 JMP     REST       ; GO TO REST OF ROUTINE
10–39     ; OTHER SERVICE ROUTINES
 .          .
 .          .
 .          .
          REST:    STA     COUNT
                   MVI     A, 0B3H    ; LOAD COUNTER FOR
                   STA     800AH      ; NEW INTERVAL
                   PULL    PSW        ; RESTORE ACC + STATUS
                   EI                 ; ENABLE INTERRUPTS
                   RET                ; RETURN
```

Figure 12. Example of Interrupt-Driven Routine

Some applications may, however, preclude use of the interrupt-driven method. The TMS 8080A tests or polls the TMS 5501 interrupt request under program control. If an interrupt request is present, the TMS 8080A addresses the read interrupt address function to determine the origin of the interrupt. The TMS 5501 interrupt address is identical to the opcode for the RST instruction. The TMS 8080A must decode the interrupt address under program control.

The routine in Figure 13 is an example of an interrupt polling routine. The TMS 8080A tests for an interrupt and reads the interrupt address if an interrupt is present. Sixty-four program bytes beginning at location 40 (HEX) are reserved for the interrupt service routines with eight bytes for each interrupt. The polling routine and one service routine are shown. The service routine pulses the TMS 5501 output bits when interval timer 3 completes the interval and then loads the timer for a 20 millisecond interval. The TMS 5501 base address is A000 (HEX). Note that the service routines are not stored in the same order as they are prioritized. This is the result of the decoding technique used in the POLL routine.

```
ADDRESS (HEX)

40–47     ; TIMER 1 SERVICE ROUTINE
48–4F     ; SERIAL RECEIVE SERVICE ROUTINE
50–57     ; EXTERNAL SENSOR SERVICE ROUTINE
58–5F     ; TIMER 4 SERVICE ROUTINE
60–67     ; TIMER 2 SERVICE ROUTINE
67–6F     ; SERIAL TRANSMIT SERVICE ROUTINE
  70      TIME 3:  SUB    A        ; ALL ZEROES → ACCUMULATOR
  71               STA    0A006H   ; OUTPUT ONES
  74               CMA             ; ALL ONES → ACCUMULATOR
  75               JMP    REST     ; GO TO REST OF ROUTINE
78–7F     ; TIMER 5 OR XI7 SERVICE ROUTINE

          REST:    STA    0A006H   ; OUTPUT ZEROES
                   MVI    A,313D   ; INTERVAL COUNT → A
                   STA    0A00BH   ; LOAD TIMER 3
                   JMP    POLL     ; RETURN TO START OF POLLING
                                   ; ROUTINE TO TEST FOR
                                   ; NEXT INTERRUPT

          POLL:    LDA    0A003H   ; READ I/O STATUS
                   ANI    20H      ; TEST INTERRUPT STATUS
                   JNC    NOINT    ; BRANCH IF NO INTERRUPT
                   LDA    0A002H   ; READ INTERRUPT ADDRESS
                   ANI    1CH      ; MASK VECTOR FIELD
                   RLC             ; MULTIPLY VECTOR BY TWO
                   ORI    40H      ; ADD OFFSET OF 40 (HEX)
                   MVI    H,0      ; CLEAR H
                   MOV    L,A      ; JUMP ADDRESS → L
                   PCHL            ; JUMP TO SERVICE ROUTINE
          NOINT:   —
```

Figure 13. Example of Polling Routine

C. INTERRUPT EXPANSION

When multiple TMS 5501s are used in the TMS 8080A system, there are more than the eight interrupts normally serviced by the RST instruction. The use of a combination interrupt-driven and polling service routine provides a simple expanded interrupt capability. When two TMS 5501s are used in a system, the INT output of one TMS 5501 is connected to the INT input of the TMS 8080A. The INT output of the other TMS 5501 is connected to the SENS or the XI7 input of the first TMS 5501 as shown in Figure 14.

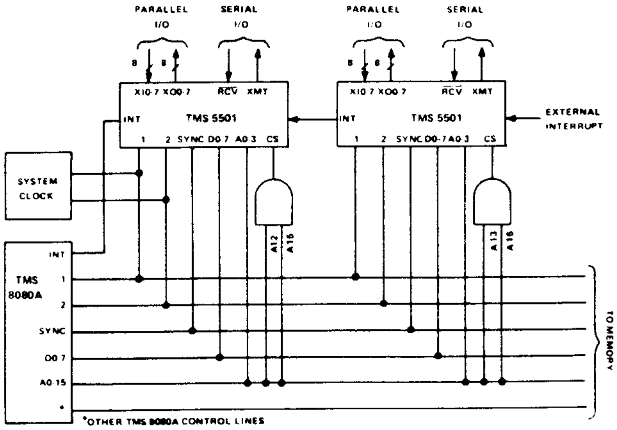


Figure 14. Interrupt Expansion for TMS 5501s

Note that one interrupt input, SENS or XI7, of the first TMS 5501 is used to connect the INT output of the second TMS 5501 and is therefore unavailable for use as an external interrupt input. Consequently, there are fifteen user interrupt sources in a two TMS 5501 system.

The system is interrupt-driven since any enabled interrupt in either TMS 5501 causes an interrupt to the TMS 8080A to be generated. Only the first TMS 5501, however, should be enabled for interrupt acknowledge since only one device should jam the RST instruction on the data bus.

After the TMS 8080A acknowledges the interrupt and executes the RST instruction, one of eight service routines is entered. Seven of the eight service routines should be dedicated to the seven user interrupts of the first TMS 5501. The eighth service routine must poll the second TMS 5501 to determine which of the eight interrupts of the second TMS 5501 generated the interrupt.

The priorities of the interrupt are dependent upon whether SENS or XI7 is used for connecting the INT output of the second TMS 5501. Table VII lists the interrupt priorities for both schemes.

Table VII. Interrupt Priorities for Expanded System

	SIGNAL	
PRIORITY	USING SENS	USING XI7
1	TIMER 1 (1)	TIMER 1 (1)
2	TIMER 2 (1)	TIMER 2 (1)
3	TIMER 1 (2)	SENS (1)
4	TIMER 2 (2)	TIMER 3 (1)
5	SENS (2)	SERIAL RECEIVE (1)
6	TIMER 3 (2)	SERIAL TRANSMIT (1)
7	SERIAL RECEIVE (2)	TIMER 4 (1)
8	SERIAL TRANSMIT (2)	TIMER 1 (2)
9	TIMER 4 (2)	TIMER 2 (2)
10	TIMER 5 OR XI7 (2)	SENS (2)
11	TIMER 3 (1)	TIMER 3 (2)
12	SERIAL RECEIVE (1)	SERIAL RECEIVE (2)
13	SERIAL TRANSMIT (1)	SERIAL TRANSMIT (2)
14	TIMER 4 (1)	TIMER 4 (2)
15	TIMER 5 OR XI7 (1)	TIMER 5 OR XI7 (2)

The interrupt service routine is similar to the routine illustrated in Figure 11. The primary difference is that the poll routine replaces the service routine for SENS or XI7 for the first TMS 5501. The interrupt service routines for the first TMS 5501 require the first sixty-four memory bytes plus whatever additional memory is required in another part of program memory. The interrupt service routines for the second TMS 5501 require the second block of sixty-four memory bytes plus whatever additional memory is required. A possible memory map of the service routines is illustarted in Figure 15.

If three TMS 5501s are used, the interrupt system is similarly expanded. The INT outputs of the second and third TMS 5501s can be connected to the SENS and the XI7 inputs of the first TMS 5501. Alternately, the INT output of the third TMS 5501 is connected to the SENS or XI7 input of the second TMS 5501, which is similarly connected to the first TMS 5501. Additional TMS 5501s can be added as required.

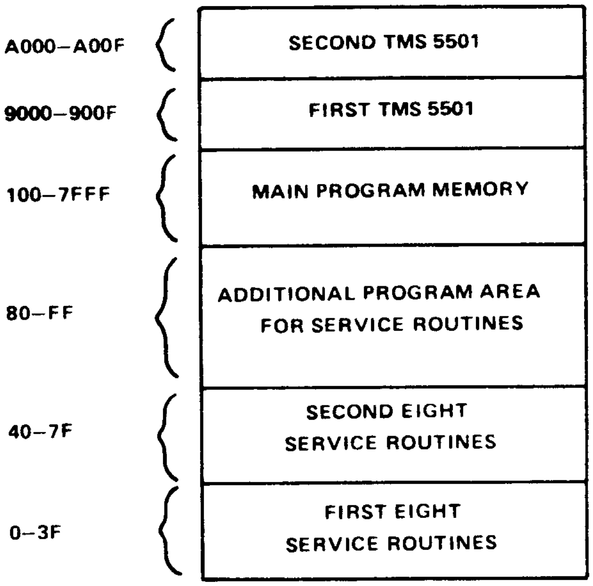

A000–A00F
SECOND TMS 5501
9000–900F
FIRST TMS 5501
100–7FFF
MAIN PROGRAM MEMORY
80–FF
ADDITIONAL PROGRAM AREA FOR SERVICE ROUTINES
40–7F
SECOND EIGHT SERVICE ROUTINES
0–3F
FIRST EIGHT SERVICE ROUTINES

Figure 15. Example Memory Map for Multiple TMS 5501 System

D. SERIAL COMMUNICATIONS

The serial communications capability of the TMS 5501 provides a powerful and economical tool for data communications. The asynchronous serial receiver and transmitter, often referred to as a UART, perform the communications related tasks such as format conversion, timing, and buffering as described in Section II-D.

The TMS 8080A initiates a serial output by addressing the load transmitter buffer function in order to transfer the output character to the UART. To avoid loading the UART faster than it can transmit characters, the TMS 5501 has an I/O status bit called transmitter buffer empty. This status bit is high whenever the UART is ready to accept a new character from the TMS 8080A. If enabled, a low-to-high transition (full-to-empty) of the transmitter buffer empty bit generates the level 5 interrupt. Consequently, the TMS 5501 should be polled for a high transmitter buffer empty bit or the level 5 interrupt should be received before a new character is loaded into the UART.

When a serial character is input to the UART, the receiver buffer loaded status is set. If enabled, the level 4 interrupt is generated. Since the input data stream is asynchronous to the main program, the receiver buffer loaded status bit must be polled or the level 4 interrupt received before the TMS 8080A can read in the character. The UART also has I/O flags which indicate transmission errors and UART status. The I/O service routine can check the frame error and overrun error flags to determine the validity of the input. Some applications, however, may not need to test these flags.

The TMS 5501 UART does not implement modem control signals. These signals are not used in many applications where control of a modem is not required such as communications with a local data terminal. If the modem control signals are required, the TMS 8080A can use the TMS 5501 parallel input and output ports to test and to control the modem under program control.

The UART is designed to send and to receive eight-bit data characters (plus framing). Shorter characters may be transmitted by right justifying the character and left filling with high or one bits. Figure 16 shows how a five-bit character is formatted for transmission. The most significant three bits are high and cause the receiver to interpret them as stop and idle bits.

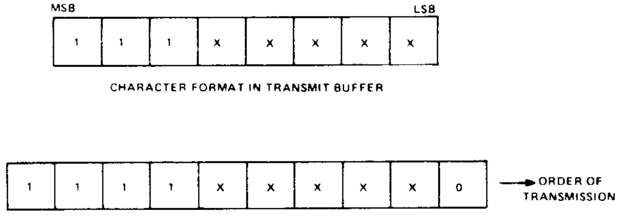

MSB
LSB
1 1 1 x x x x x
CHARACTER FORMAT IN TRANSMIT BUFFER
1 1 1 1 x x x x x 0
ORDER OF TRANSMISSION
CHARACTER FORMAT AS TRANSMITTED (1 STOP BIT)

Figure 16. Transmission Format for Five-Level Code

The receiver, however, cannot interpret shorter codes correctly unless sufficient stop or idle bits are transmitted to space start bits at least 10 bit times apart.

The UART does not perform parity generation or checking. The TMS 8080A, however, has a parity condition flag which simplifies the parity generation and check tasks.

The routine in Figure 17 illustrates the use of the UART. The routine inputs ASCII characters, checks their parity (ODD), and outputs them. A null character is transmitted if the parity is incorrect. The routine demonstrates the use of the interrupt (receive buffer loaded) and polling (transmit buffer empty) techniques. The transmit interrupt is disabled which allows the receive service routine to use a 16-byte block.

```
; RECEIVE SERVICE ROUTINE
IORCV: PUSH   PSW     ; SAVE STATUS AND ACC.
       LDA    8000H   ; INPUT CHAR → ACC
       CMP    A       ; TEST PARITY
       JPO    SKIP    ; JUMP IF GOOD PARITY
       MVI    A,80H   ; ELSE REPLACE WITH NULL
SKIP:  PUSH   PSW     ; SAVE CHARACTER ON STACK
       JMP    REST    ; JUMP TO REST OF ROUTINE
 • • •
REST:  LDA    8003H   ; INPUT I/O STATUS
       ANI    10H     ; TEST TRANSMIT STATUS
       JZ     REST    ; LOOP UNTIL READY
       POP    PSW     ; RESTORE CHAR → ACC
       STA    8006H   ; TRANSMIT CHARACTER
       POP    PSW     ; RESTORE STATUS AND ACC
       EI             ; ENABLE INTERRUPTS
       RET            ; RETURN
```

Figure 17. Example UART Control Routine

The transmitter and receiver outputs are TTL compatible and normally require buffering to interface with a transmission line. The Texas Instruments *Linear and Interface Circuits Applications* book contains a description of common line transmission interface circuits.

E. DESIGN CONSIDERATIONS

1. Clock Frequency

The TMS 5501 interval count frequency and serial communication clock frequency are derived from the system clock inputs ($\phi 1$, $\phi 2$). The 64-microsecond interval count time and the standard baud rates are dependent upon using a two-megahertz system clock. Operation at any other frequency will result in a different interval count time and non-standard baud rates.

2. WAIT States

The TMS 8080A READY input is used to synchronize the CPU with slow memories such as ROMs. When not-ready is indicated during a memory cycle, the TMS 8080A enters a wait state and suspends operation until the memory system indicates ready. The TMS 5501 does not have a ready input or output and cannot detect that the TMS 8080A is in a wait state. Therefore, the TMS 5501 will not operate correctly if wait states occur while it is being addressed. Therefore, READY must be true whenever the TMS 5501 is selected. This requirement can be met by disabling a wait request whenever the TMS 5501 is addressed. Figure 18 illustrates the use of slow memories with the TMS 5501.

3. Halt

A halt instruction should not be placed in a memory location immediately contiguous to a TMS 5501 base address. The execution of the halt in such a location would inadvertently select the TMS 5501 read receiver buffer function and clear the receiver buffer loaded flag. The designer should be particularly careful to avoid this situation in systems which do not fully decode the memory address.

4. I/O Addressing

During I/O operations, the TMS 8080A I/O address appears in duplicate on A7 through A0 and A15 through A8. Since the TMS 5501 does not check the status word during SYNC for the INP and OUT status indicators, it is possible to select the TMS 5501 during an I/O operation. The restriction of the I/O addressing space or the external gating of the TMS 5501 chip select by the I/O status indicators will eliminate this problem. The designer can use this characteristic to use I/O instructions to access the TMS 5501. For example, use address bit 15 to select the TMS 5501 as described in Section II-A. The TMS 5501 can then be accessed by the OUT and IN instructions. Any I/O address greater than 7F (HEX) selects the TMS 5501

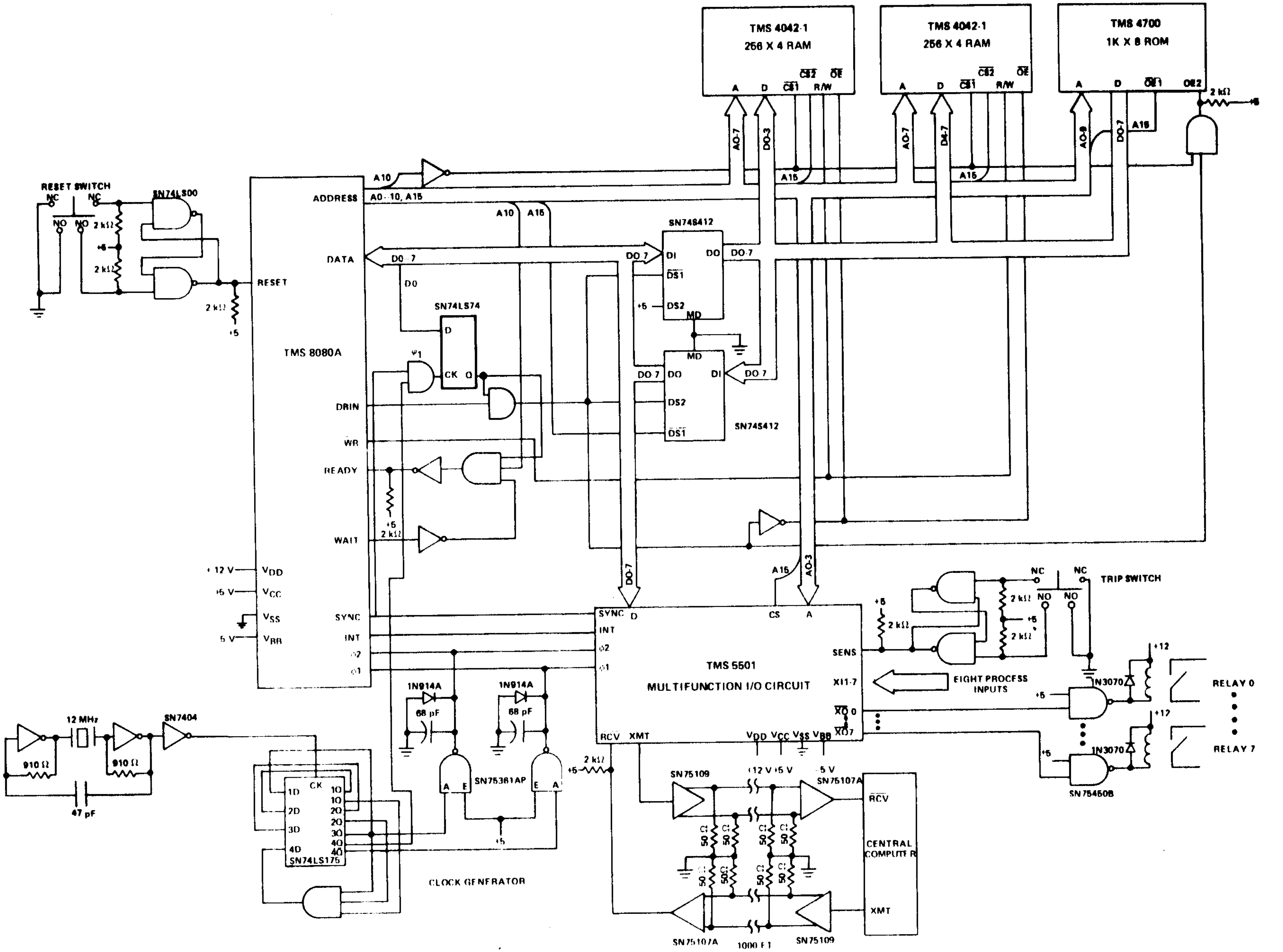


Figure 18. Sequential Controller Using TMS 5501 With Slow Memory

since the I/O address is duplicated on the address bus. The least-significant four bits still select the TMS 5501 function. Table VIII lists a few of the memory and I/O instructions which perform the same function in such a system.

Table VIII. TMS 5501 I/O Addressing

MEMORY		I/O		FUNCTION
LDA	8000H	IN	80H	RECEIVE BUFFER → ACCUMULATOR
LDA	8003H	IN	83H	I/O STATUS → ACCUMULATOR
MOV	A,M	IN	81H	EXT INPUTS → ACCUMULATOR
	(HL) 8001H			
STAX	D	OUT	88H	ACCUMULATOR → MASK REGISTER
	(DE) 8008H			
STA	800AH	OUT	8AH	ACCUMULATOR → TIMER 2

IV. TMS 5501 APPLICATIONS

The TMS 5501 is a multifunction I/O controller suitable for any TMS 8080A-based system requiring data I/O, serial communications, interrupt control, and interval timing. It is ideal for applications such as intelligent terminals, data entry equipment, data collection, machinery control, and process control.

The system shown in Figure 18 is an example of a TMS 8080A-based system using the I/O functions of the TMS 5501. The sample system is an intelligent sequential controller with remote data entry capability. The controller can maintain up to eight relay closures and can sense up to eight discrete process inputs. The serial asynchronous communication link is used for data transfers between the control computer and the remote controller.

The central computer transmits control data, such as contact closure interval times, to the sequential controller over the serial communication link. The TMS 8080A then executes the ROM-based control program and sequences the output closures in response to transmitted intervals and process inputs. For example, the controller can close a relay for either one second or until an input becomes true, whichever occurs first. The controller releases all relay closures whenever the trip switch is closed. At periodic intervals, the controller transmits the output and input status to the central computer, which is connected to the sequential controller by a 1000-foot long double twisted pair cable.

The TMS 5501 provides the multiple I/O functions needed by the controller. The TMS 5501 buffers the process inputs and outputs, performs the serial data communication task, times event intervals for accurate control, and senses the trip signal. The interrupt control logic of the TMS 5501 generates a prioritized interrupt to the CPU at the end of an interval, when new data has been received, when the transmitter is ready, or when the controller is to be tripped. The CPU is thus free to execute the main control program or diagnostics concurrently with event timing and serial data transfers.

Note that the serial transmission baud rate is generated by the TMS 5501 and requires no external timing elements. Test bit 4 can be set high to allow high speed data transfer at up to 76.8 kilobaud. The interval timers, however, are also then driven by a higher speed clock. Long events can be accurately timed by dividing the long delay into several smaller delays and sequentially timing the lesser intervals.

The system memory consists of a TMS 4700 1024 X 8 ROM and two TMS 4042-1 256 X 4 RAMs. The 650-nanosecond access TMS 4042-1 RAM has been selected to demonstrate the use of wait states for slow memory in TMS 5501 systems. Recall that memory cycles which reference the TMS 5501 should not contain wait states. The memory READY input is false only when the RAM is addressed (A10 high) and for only one wait state since an active WAIT output disables the wait request. The TMS 4042-2 RAM can be used instead to eliminate the need for wait states.

The SN75450B dual peripheral drivers interface the TMS 5501 external outputs to the relays. The process inputs are assumed to be TTL compatible inputs. Input filtering, if desired, can be performed either by a RC network or by the CPU under program control. A SN75107A receiver and a SN75109 transmitter are used to buffer the TMS 5501 serial input and output signals. The differential line driver and receiver provide reliable data transmission over twisted pair cables in excess of 1000 feet with excellent noise rejection. Other line transmission circuits are available for applications requiring RS-232C or current loop interfaces.

The TMS 5501 multifunction input/output controller is a powerful and versatile design tool for a TMS 8080A-based system. It performs parallel I/O buffering, serial data transfers, multiple event timing, and interrupt processing in a single integrated circuit. It provides improved system performance and flexibility, reduced component count, and reduced system cost.