

**PRELIMINARY**  
 Notice: This is not a final specification. Some  
 parametric limits are subject to change.  
**(M5M82C54P)**

# MITSUBISHI LSIs

## M5M82C54P,-6

CMOS PROGRAMMABLE INTERVAL TIMER

6249828 MITSUBISHI (MICMPTR/MIPRC)

91D 11984 D

T-51-19

### DESCRIPTION

The M5M82C54P,-6 is a programmable general-purpose timer device developed by using the silicon-gate CMOS process. It offers counter and timer functions in systems using an 8-bit parallel-processing CPU. The use of the M5M82C54P,-6 frees the CPU from the execution of looped programs, count-operation programs and other simple processing involving many repetitive operations, thus contributing to improved system throughputs. The M5M82C54P,-6 works on a single power supply, and both its input and output can be connected to a TTL circuit.

Parameter	M5M82C54P	M5M82C54P-6
Clock high pulse width	60ns	55ns
Clock low pulse width	60ns	110ns
Clock cycle time	125ns	165ns
Propagation time from read to output	120ns	170ns

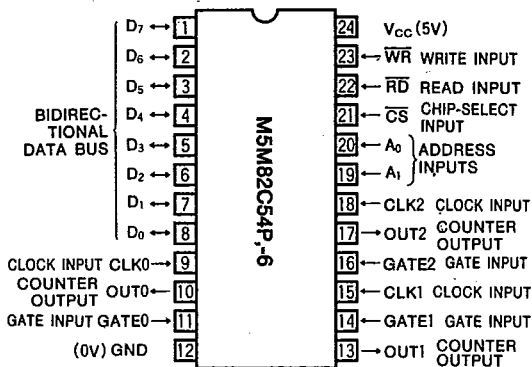
### FEATURES

- Single 5V supply voltage
- TTL compatible
- Pin connection compatible with M5L8253P-5
- Clock period : M5M82C54P-6 ..... DC~6MHz  
 M5M82C54P ..... DC~8MHz
- 3 independent built-in 16-bit down counters
- 6 counter modes freely assignable for each counter
- Binary or decimal counts
- Read-back command for monitoring the count and status

### APPLICATION

Delayed-time setting, pulse counting and rate generation in microcomputers.

### PIN CONFIGURATION (TOP VIEW)



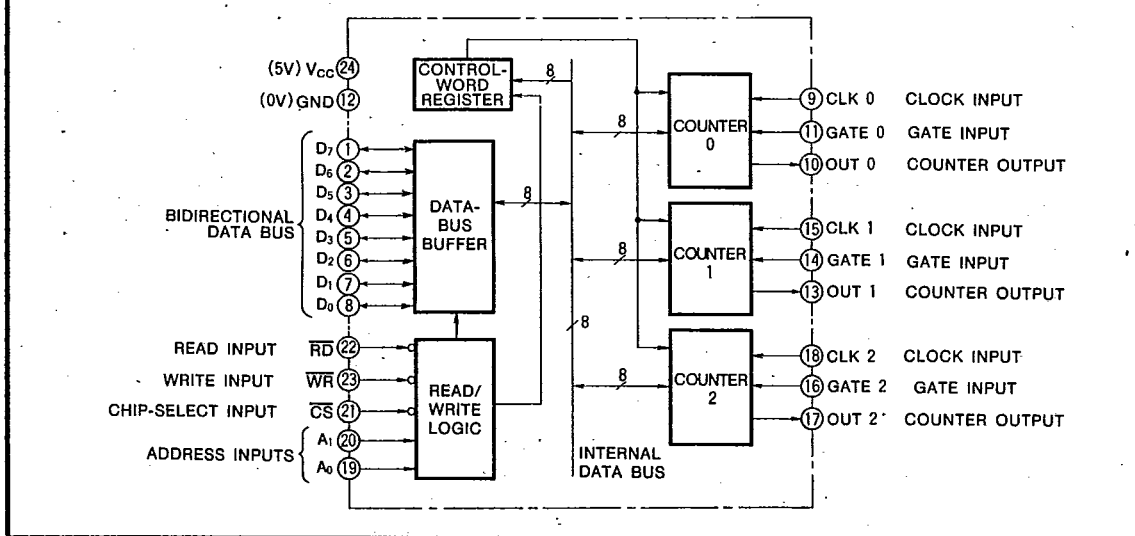
Outline 24P4

### FUNCTION

Three independent 16-bit counters allow free programming based on mode-control instructions from the CPU. When roughly classified, there are 6 modes (0~5). Mode 0 is mainly used as an interruption timer and event counter, mode 1 as a digital one-shot, modes 2 and 3 as a rate generator, mode 4 for a software triggered strobe, and mode 5 for a hardware triggered strobe.

The count can be monitored and set at any time. Besides the count, the status of the counter can be monitored by Read-back command. The counter operates with either the binary or BCD system.

### BLOCK DIAGRAM



## DESCRIPTION OF FUNCTIONS

### Data-Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the M5M82C54P,-6 to the system-side data bus. Transmission and reception of all the data including control words for mode designation and values written in, and read from, the counters are carried out through this buffer.

### Read/Write Logic

The read/write logic accepts control signals ( $\overline{RD}$ ,  $\overline{WR}$ ) from the system and generates control signals for each counter. It is enabled or disabled by the chip-select signal ( $\overline{CS}$ ); if  $\overline{CS}$  is at the high-level the data-bus buffer enters a floating (high-impedance) state.

### Read Input ( $\overline{RD}$ )

The count of the counter designated by address inputs  $A_0$  and  $A_1$  on the low-level is output to the data bus.

### Write Input ( $\overline{WR}$ )

Data on the data bus is written in the counter or control-word register designated by address inputs  $A_0$  and  $A_1$  on the low-level.

### Address Inputs ( $A_0$ , $A_1$ )

These are used for selecting one of the 3 internal counters and either of the control-word registers.

### Chip-Select Input ( $\overline{CS}$ )

A low-level on this input enables the M5M82C54P,-6. Changes in the level of the  $\overline{CS}$  input have no effect on the operation of the counters.

### Control-Word Register

This register stores information required to give instructions about operational modes and to select binary or BCD counting. It allows reading, using Read back command.

### Counters 0, 1 and 2

These counters are identical in operation and independent of each other. Each is a 16-bit, presettable, down counter, and has clock-input, gate-input and output pins. The counter can operate in either binary or BCD using the falling edge of each clock. The mode of counter operation and the initial value from which to start counting can be designated by software. The count can be read by input instruction at any time, and there is a "read-on-the-fly" function which enables stable reading by latching each instantaneous count to the registers by a special counter-latch instruction.

## CONTROL WORD AND INITIAL-VALUE LOADING

The function of the M5M82C54P,-6 depends on the system software. The operational mode of the counters can be specified by writing control words ( $A_0$ ,  $A_1=1, 1$ ) into the control-word registers.

The programmer must write out to the M5M82C54P,-6 the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

Fig. 1 shows control-word format, which consists of 4 fields. Only the counter selected by the  $D_7$  and  $D_6$  bits of the control word is set for operation. Bits  $D_5$  and  $D_4$  are used for specifying operations to read values in the counter and to initialize. Bits  $D_3 \sim D_1$  are used for mode designation, and  $D_0$  for specifying binary or BCD counting. When  $D_0=0$ , binary counting is employed, and any number from  $0000_{16}$  to  $FFFF_{16}$  can be loaded into the count register. The counter is counted down for each clock. The counting of  $0000_{16}$  causes the transmission of a time-out signal from the count-output pin.

The maximum number of counts is obtained when  $0000_{16}$  is set as the initial value. When  $D_0=1$ , BCD counting is employed, and any number from  $0000_{10}$  to  $9999_{10}$  can be loaded on the counter.

Neither system resetting nor connecting to the power supply sets the control word to any specific value. Thus to bring the counters into operation, the above-mentioned control words for mode designation must be given to each counter, and then 1~2 byte initial counter values must be set. The following is an example of this programming step.

To designate mode 0 for counter 1, with initial value  $8253_{16}$  set by binary count, the following program is used:

```
MVI A, 7016    Control word 7016
OUT n1          n1 is control-word-register address
MVI A, 5316    Low-order 8 bits
OUT n2          n2 is counter 1 address
MVI A, 8216    High-order 8 bits
OUT n2          n2 is counter 1 address
```

Thus, the program generally has the following sequence:

- (1) Control-word output to counter  $i$  ( $i=0, 1, 2$ ).
- (2) Initialization of low-order 8 counter bits
- (3) Initialization of high-order 8 counter bits

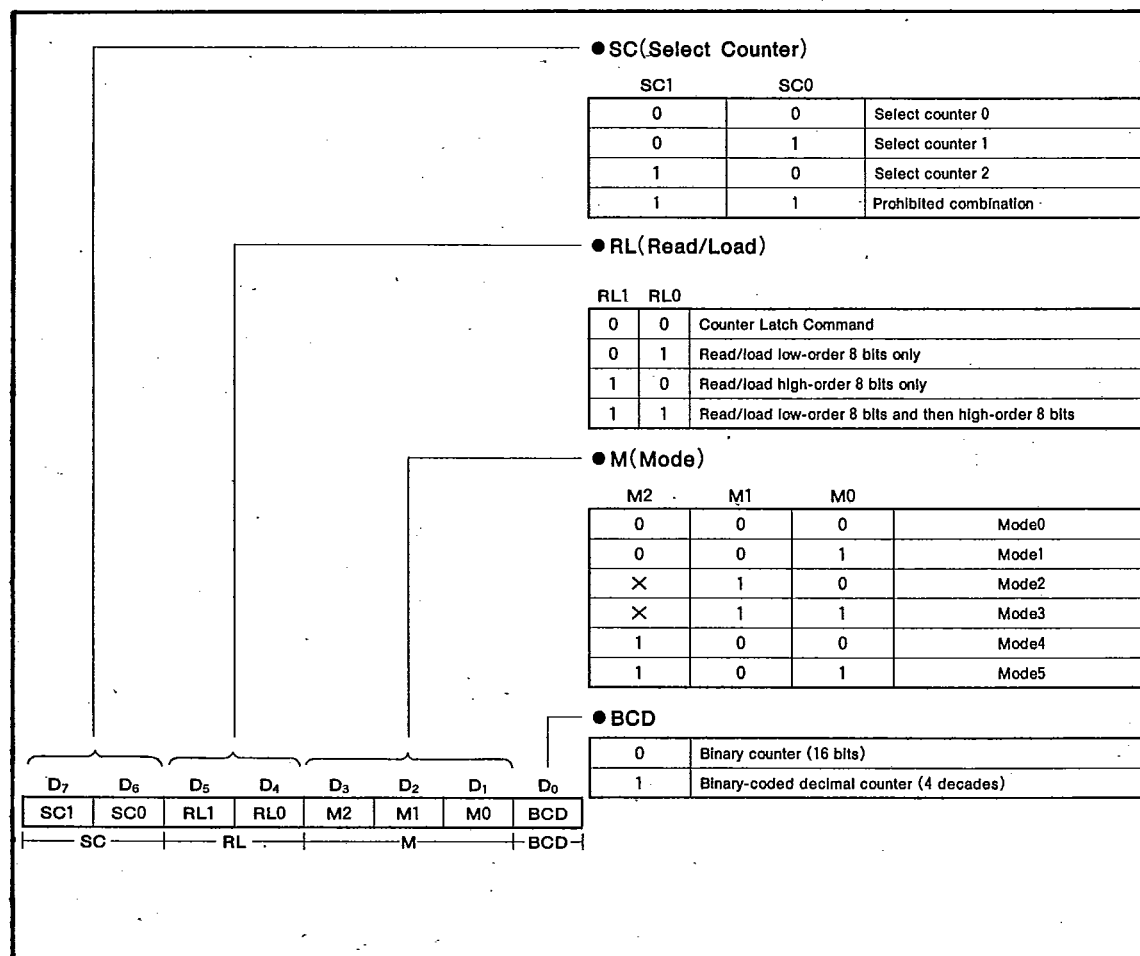
The three counters can be executed in any sequence. It is possible, for instance, to designate the mode of each counter and then load initial values in a different order. Initialization of the counters designated by RL 1 and RL 0 must be executed in the order of the low-order 8 bits and then the high-order 8 bits for the counter in question.

6249828 MITSUBISHI (MICMPTR/MIPRC)

91D 11986 DT-51-19

**CMOS PROGRAMMABLE INTERVAL TIMER**
**Table 1 Basic Functions**

CS	RD	WR	A <sub>1</sub>	A <sub>0</sub>	Function
0	1	0	0	0	Data bus→Counter 0
0	1	0	0	1	Data bus→Counter 1
0	1	0	1	0	Data bus→Counter 2
0	1	0	1	1	Data bus→Control-word register
0	0	1	0	0	Data bus←Counter 0
0	0	1	0	1	Data bus←Counter 1
0	0	1	1	0	Data bus←Counter 2
0	0	1	1	1	3-state
1	X	X	X	X	3-state
0	1	1	X	X	3-state


**Fig. 1 Control-Word Format**

6249828 MITSUBISHI{MICMPTR/MIPRC}

91D 11987

D T-519

**CMOS PROGRAMMABLE INTERVAL TIMER**
**MODE DEFINITION**
**Mode 0 (Interrupt on Terminal Count)**

Mode set and initialization cause the counter output to go low-level (see Fig. 2). When the counter is loaded with an initial value, it will start counting the clock input. When the terminal count is reached, the output will go high and remain high until the selected count register is reloaded with the mode. This mode can be used when the CPU is to be interrupted after a certain period or at the time of counting up.

Fig. 2 shows a setting of 4 as the initial value. If gate input goes low, counting is inhibited for the duration of the low-level period.

Reloading of the initial value during count operation will stop counting by the loading of the first byte and start the new count by the loading of the second byte.

**Mode 1 (Programmable One-Shot)**

The gate input functions as a trigger input. A gate-input rising edge causes the generation of low-level one-shot output with a predetermined clock length starting from the next clock. Fig. 3 shows an initial setting of 4. While the counter output is at the low-level (during one-shot), loading of a new value does not change the one-shot pulse width, which has already been output. The current count can be read at any time without affecting the width of the one-shot pulse being output. This mode permits retriggering.

**Mode 2 (Rate Generator)**

Low-level pulses during one clock operation are generated from the counter output at a rate of one per  $n$  clock inputs (where  $n$  is the value initially set for the counter). When a new value is loaded during the counter operation, it is reflected on the output after the pulses by the current count have been output. In the example shown in Fig. 4,  $n$  is given as 4 at the outset and is then changed to 3.

In this mode, the gate input provides a reset function. While it is on the low-level, the output is maintained high; the counter restarts from the initial value, triggered by a rising gate-input edge. This gate input, therefore, makes possible external synchronization of the counter by hardware.

After the mode is set, the counter does not start counting until the rate  $n$  is loaded into the count register, with the counter output remaining at the high-level.

**Mode 3 (Square Rate Generator)**

This is similar to Mode 2 except that it outputs a square wave with the half count of the set rate. When the set value  $n$  is odd, the square-wave output will be high for  $(n+1)/2$  clock-input counts and low for  $(n-1)/2$  counts. When a new rate is reloaded into the count register during its operation, it is immediately reflected on the count directly following the output transition (high-to-low or low-to-high) of the current count. Gate-input operations are exactly the same as in Mode 2. Fig. 5 shows an example of Mode 3 operation.

**Mode 4 (Software Triggered Strobe)**

After the mode is set, the output will be high. By loading a number on the counter, however, clock-input counts can be

started and on the terminal count, the output will go low for one input-clock period and then will go high again. Mode 4 differs from Mode 2 in that pulses are not output repeatedly with the same set count. The pulse output is delayed one clock period in Mode 2, as shown in Fig. 6. When a new value is loaded into the count register during its count operation, it is reflected on the next pulse output without affecting the current count. The count will be inhibited while the gate input is low-level.

**Mode 5 (Hardware Triggered Strobe)**

This is a variation of Mode 1. The gate input provides a trigger function, and the count is started by its rising edge. On the terminal count, the counter output goes low for one clock period and then goes high. As in Mode 1, retriggering by the gate input is possible. An example of timing in Mode 5 is shown in Fig. 7.

As mentioned above, the gate input plays different roles according to the mode. The functions are summarized in Table 3.

**Table 2 Gate Operations**

Gate Mode	Low or going low	Rising	High
0	Disables counting		Enables counting
1		(1) Initiates counting (2) Resets output after next clock	
2	(1) Disables counting (2) Sets output high immediately	(1) Reloads counter (2) Initiates counting	Enables counting
3	(1) Disables counting (2) Sets output high immediately	(1) Reloads counter (2) Initiates counting	Enables counting
4	Disables counting		Enables counting
5		Initiates counting	

# M5M82C54P,-6

6249828 MITSUBISHI (MICMPTR/MIPRC)

91D 11988 DT-51-19

## CMOS PROGRAMMABLE INTERVAL TIMER

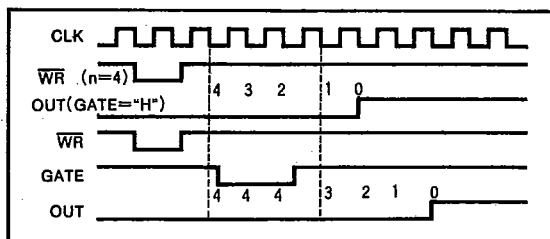


Fig. 2 Mode 0

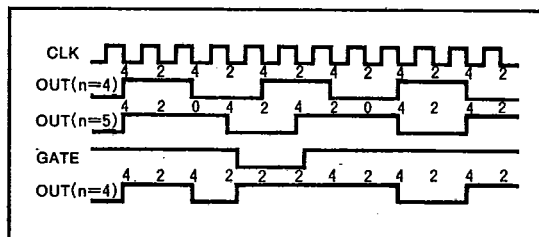


Fig. 5 Mode 3

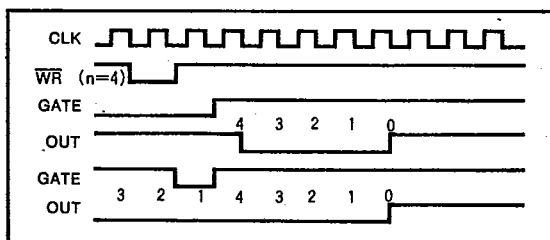


Fig. 3 Mode 1

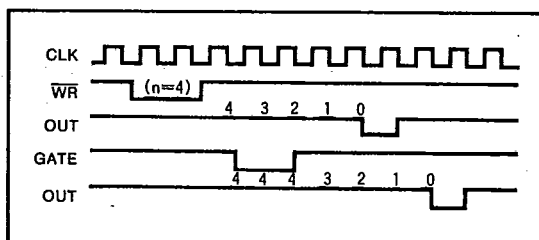


Fig. 6 Mode 4

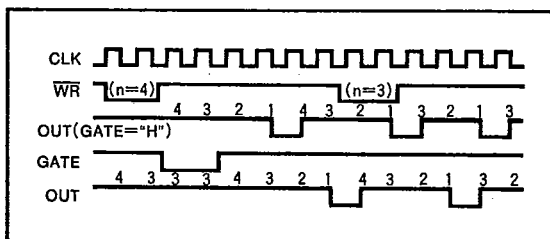


Fig. 4 Mode 2

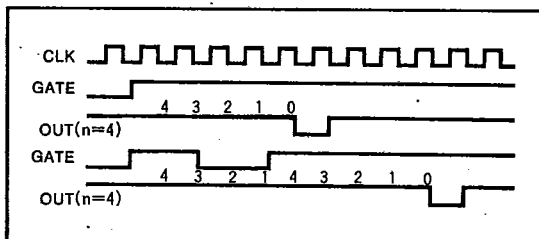


Fig. 7 Mode 5

### COUNTER MONITORING

Sometimes the counter must be monitored by reading its count or using it as an event counter. The M5M82C54P,-6 offers the following two methods for count reading:

#### Read Operation

The count can be read by designating the address of the counter to be monitored and executing a simple I/O read operation. In order to ensure correct reading of the count, it is necessary to cause the clock input to pause by external logic or prevent a change in the count by gate input. An example of a program to read the counter 1 count is shown below. If RL1, RL0=1, 1 has been specified in the control word, the first IN instruction enables the low-order 8 bits to be read and the second IN instruction enables the high-order 8 bits.

```
IN    n2 ... n2 is the counter 1 address
MOV   D, A
IN    n2
MOV   E, A
```

The IN instruction should be executed once or twice by the RL1 and RL0 designations in the control-word register.

#### Read-on-the-Fly Operation

This method makes it possible to read the current count without affecting the count operation at all. A special counter-latch command is first written in the control-word register. This causes latching of all the instantaneous counts to the register, allowing retention of stable counts. An example of a program to execute this operation for counter 2 is given below.

```
MVI   A, 1000XXXX ... D5=D4=0 designates counter latching
OUT    n1 ... n1 is the control-word-register address
IN     n3 ... n3 is the counter 2 address
MOV    D, A
IN     n3
MOV    E, A
```

In this example, the IN instruction is executed twice. Due to the internal logic of the M5M82C54P,-6 it is absolutely essential to complete the entire reading procedure. If two bytes are programmed to be read, then two bytes must be read before any OUT instruction can be executed to the same counter.

### READ BACK COMMAND

M5M82C54P,-6 has a function of reading not only the count but also status (Read Back Command). The read back command enables the next four functions.

- (1) read the current count "on the fly"
- (2) monitor the current state of the OUT pin
- (3) monitor the current state of the counter element (whether the count is loaded into the counter element or not)
- (4) read the control word

Read back operation can be specified by writing read back command into the control word registers ( $A_0, A_1 = 1, 1$ ). Fig. 8 shows the format of read back command.

Bits  $D_7$  and  $D_6$  are used for specifying read back command and fixed 1 ( $D_7 = 1, D_6 = 1$ ). Respectively bits  $D_5$  (count) and  $D_4$  (status) are used for reading the count and the status of the counter selected by the  $D_3 \sim D_1$  bits. Bit  $D_0$  must be fixed 0.

Only the count can be read "on the fly" by setting  $D_5 = 0$  and  $D_4 = 1$  as well as counter latch command above mentioned. If  $D_3 \sim D_1$  are set 1 all, the counts of three counters are simultaneously latched by one read back command. (By counter latch command, it must be latched for each counter.) Next, by read operation, the latched count is read out.

Only the status can be latched by setting  $D_5 = 1$  and  $D_4 = 0$ . By read operation, the status shown in Fig. 9 can be read.

Bit  $D_7$  gives the current state of OUT pin. When  $D_7 = 1$ , OUT = "H", and when  $D_7 = 0$ , OUT = "L". Bit  $D_6$  indicates the current state of counter element. When  $D_6 = 1$ , the initial counter value has not been loaded to counter element. This state is following.

- (1) The control word is written, but the initial counter value is not loaded
- (2) The initial counter value is written to count register, and the CLK inputs are not.

When  $D_6 = 0$ , the initial counter value has already been loaded. It is the state when the CLK falls following the rising edge after the initial value is written. Bits  $D_5 \sim D_0$  show the current state of the control word register.

It is possible to read both the count and the status. By setting  $D_5 = 0$  and  $D_4 = 0$ , the status can be read first, and the count next.

The count and/or the status are unlatched when read, so by the next read operation the current counting value can be read. And they are unlatched too when the control word is set, so the read back command must be set on all such occasions.

If multiple read back commands are written before the read operation, only the first one is valid.

Thus, the read of the status is effective when the state of output and the timing of count reading can be monitored by software.

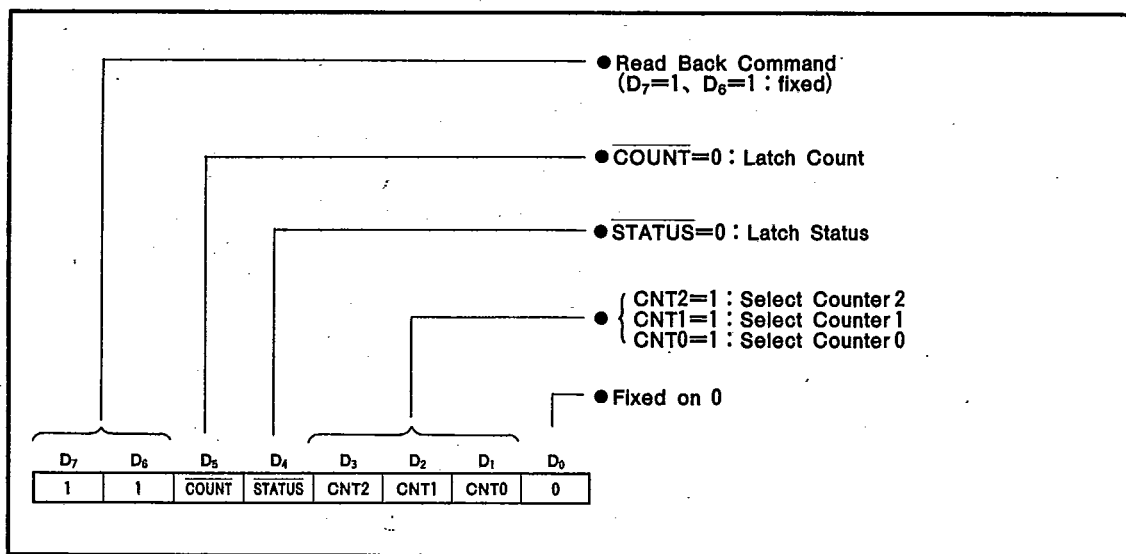


Fig. 8 Read Back Command Format

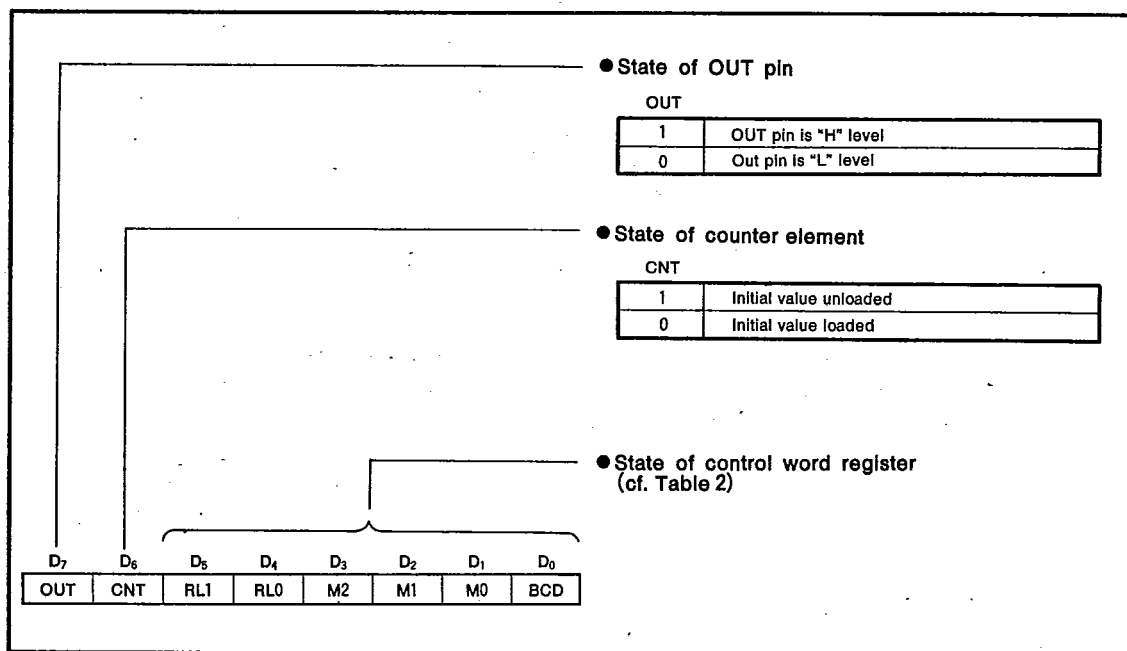


Fig. 9 Status Byte

6249828 MITSUBISHI{MICMPTR/MIPRC}

91D 11991

D T-51-19

**CMOS PROGRAMMABLE INTERVAL TIMER**
**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Conditions	Ratings	Unit
$V_{CC}$	Power supply voltage	With respect to GND	-0.3~7	V
$V_I$	Input voltage		-0.3~ $V_{CC}+0.3$	V
$V_O$	Output voltage		-0.3~ $V_{CC}+0.3$	V
$T_{opr}$	Operating free-air temperature range		-20~75	°C
$T_{stg}$	Storage temperature range		-65~150	°C

**RECOMMENDED OPERATING CONDITIONS** ( $T_a = -20 \sim 75^\circ\text{C}$ , unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
$V_{CC}$	Power supply voltage	4.5	5	5.5	V
$V_{SS}$	Supply voltage (GND)		0		V

**ELECTRICAL CHARACTERISTICS** ( $T_a = -20 \sim 75^\circ\text{C}$ ,  $V_{CC} = 5V \pm 10\%$ , unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min	Typ	Max	
$V_{IH}$	High-level input voltage		2.0		$V_{CC}+0.3$	V
$V_{IL}$	Low-level input voltage		-0.3		0.8	V
$V_{OH}$	High-level output voltage	$V_{SS}=0V, I_{OH}=-400\mu A$	2.4			V
		$V_{SS}=0V, I_{OH}=-20\mu A$	4.4			
$V_{OL}$	Low-level output voltage	$V_{SS}=0V, I_{OL}=2.0mA$			0.45	V
$I_{IH}$	High-level input current	$V_{SS}=0V, V_I=5.50V$			$\pm 10$	$\mu A$
$I_{IL}$	Low-level input current	$V_{SS}=0V, V_I=0V$			$\pm 10$	$\mu A$
$I_{OZ}$	Off-state output current	$V_{SS}=0V, V_I=0 \sim V_{CC}$			$\pm 10$	$\mu A$
$I_{CC}$	Power supply current	M5M82C54P $V_{SS}=0V, f=8MHz$			10	mA
		M5M82C54P-6 $V_{SS}=0V, f=6MHz$				
$I_{CC}$	Power supply current during STAND BY	$V_{SS}=0V$ , other inputs are $V_{SS}$ or $V_{CC}$			10	$\mu A$
$C_I$	Input capacitance	$V_{IL}=V_{SS}, f=1MHz, 25mVrms, T_a=25^\circ C$			10	pF
$C_{I/O}$	Input/output capacitance	$V_{I/O}=V_{SS}, f=1MHz, 25mVrms, T_a=25^\circ C$			20	pF

6249828 MITSUBISHI (MICMPTR/MIPRC)

91D 11992 DT-51-19

**CMOS PROGRAMMABLE INTERVAL TIMER**
**TIMING REQUIREMENTS** ( $T_a = -20 \sim 75^\circ\text{C}$ ,  $V_{CC} = 5V \pm 10\%$ ,  $V_{SS} = 0V$ , unless otherwise noted)

**Read cycle**

Symbol	Parameter	Alternative symbol	Test condition	Limits			Unit
				Min	Typ	Max	
$t_{w(R)}$	Read pulse width	$t_{RR}$	$C_L = 150\text{pF}$	150			ns
$t_{SU(S-R)}$	CS setup time before read	$t_{SR}$		0			ns
$t_{SU(A-R)}$	Address setup time before read	$t_{AR}$		45			ns
$t_{H(R-A)}$	Address hold time after read	$t_{RA}$		0			ns
$t_{rec(R)}$	Read recovery time	$t_{RV}$		200			ns

**Write cycle**

Symbol	Parameter	Alternative symbol	Test condition	Limits			Unit
				Min	Typ	Max	
$t_{w(W)}$	Write pulse width	$t_{WW}$	$C_L = 150\text{pF}$	150			ns
$t_{SU(S-W)}$	CS setup time before write	$t_{SW}$		0			ns
$t_{SU(A-W)}$	Address setup time before write	$t_{AW}$		0			ns
$t_{H(W-A)}$	Address hold time after write	$t_{WA}$		0			ns
$t_{SU(D-W)}$	Data setup time before write	$t_{DW}$		120 [100]	(Note 1)		ns
$t_{H(W-D)}$	Data hold time after write	$t_{WD}$		0			ns
$t_{rec(W)}$	Write recovery time	$t_{RV}$		200			ns

Note 1 : M5M82C54P/P-6 is also invested with the extended specification showed in the bracket.

**Clock and gate timing**

Symbol	Parameter	Alternative  symbol	Test condition	Limits				Unit
				M5M82C54P		M5M82C54P-6		
				Min	Max	Min	Max	
$t_{W(\#H)}$	Clock high pulse width	$t_{PWH}$	$C_L=150pF$	60		55		ns
$t_{W(\#L)}$	Clock low pulse width	$t_{PWL}$		60		110		ns
$t_{C(\#)}$	Clock cycle time	$t_{CLK}$		125		165		ns
$t_r(\#)$	Clock rise time	$t_R$			100		100	ns
$t_f(\#)$	Clock fall time	$t_F$			100		100	ns
$t_{W(GH)}$	Gate high pulse width	$t_{GW}$		50		50		ns
$t_{W(GL)}$	Gate low pulse width	$t_{GL}$		50		50		ns
$t_{SU(G-\#)}$	Gate setup time before clock	$t_{GS}$		50		50		ns
$t_h(\#-G)$	Gate hold time after clock	$t_{GH}$		50		50		ns

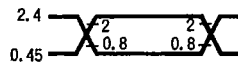
6249828 MITSUBISHI (MICMPTR/MIPRC) 91D 11993 DT-51-19  
**CMOS PROGRAMMABLE INTERVAL TIMER**

**SWITCHING CHARACTERISTICS** ( $T_a = -20 \sim 75^\circ\text{C}$ ,  $V_{CC} = 5V \pm 10\%$ ,  $V_{SS} = 0V$ , unless otherwise noted) (Note 2)

Symbol	Parameter	Alternative symbol	Test condition	Limits				Unit
				M5M82C54P		M5M82C54P-6		
				Min	Max	Min	Max	
t <sub>PZV(A-Q)</sub>	Propagation time from address to output	t <sub>AD</sub>	C <sub>L</sub> =150pF		220		220	ns
t <sub>PZV(R-Q)</sub>	Propagation time from read to output	t <sub>RD</sub>			120		170	ns
t <sub>PVZ(R-Q)</sub>	Propagation time from read to output floating (Note 3)	t <sub>DF</sub>		5	90	5	90	ns
t <sub>PXV(G-Q)</sub>	Propagation time from gate to output	t <sub>ODG</sub>			120		120	ns
t <sub>PXV( # -Q)</sub>	Propagation time from clock to output	t <sub>OD</sub>			150		150	ns

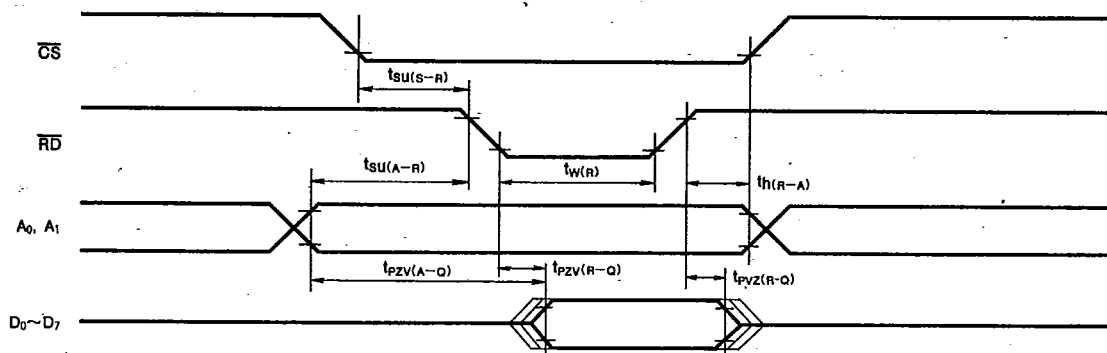
Note 2 : A.C Testing waveform  
 Input pulse level 0.45~2.4V  
 Input pulse rise time 10ns  
 Input pulse fall time 10ns  
 Reference level input  $V_{IH} = 2V$ ,  $V_{IL} = 0.8V$   
 Output  $V_{OH} = 2V$ ,  $V_{OL} = 0.8V$

3 : Test condition is not applied

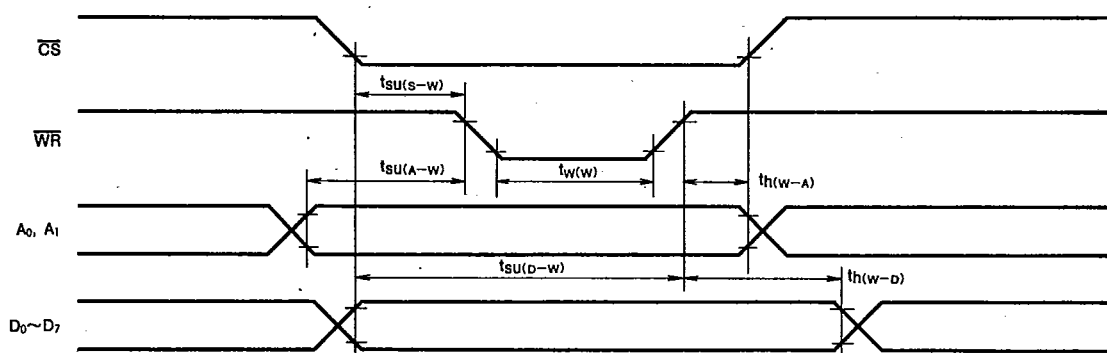


**TIMING DIAGRAMS** (Reference voltage: High=2.0V, Low=0.8V)

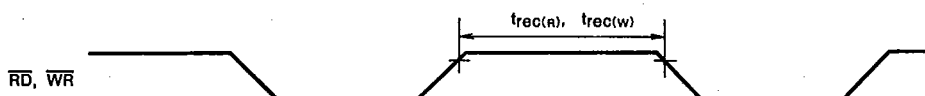
**Read Cycle**



**Write Cycle**



**(Recovery Time)**



**Clock and Gate Cycle**

