INTEGRATED CIRCUIT

Toshiba
東芝 **TECHNICAL DATA**

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP4740P   TMP4720P

SILICON MONOLITHIC

N-CHANNEL SILICON GATE DEPRESSION LOAD

PRELIMINARY

NMOS 4-BIT SINGLE CHIP MICROCOMPUTER (TLCS-47N)   TMP4740P, TMP4720P

GENERAL DESCRIPTION

The TLCS-47 is the high speed and high performance, 4-bit single chip microcomputer series designed for the general purpose use.

The TLCS-47 has variously powerful functions in order to meet with the advanced and complicated applications, which will be made in near future. In addition, software compatible NMOS family (TLCS-47N) and CMOS family (TLCS-47C) are also provided.

The TMP4740P and TMP4720P are the standard chips for the TLCS-47N. These chips are similar to each other, except memory capacity. The TMP4700C is an evaluator chip used for the system development.

Part No.	ROM (Bit)	RAM (Bit)
TMP4740P	4,096 × 8	256 × 4
TMP4720P	2,048 × 8	128 × 4
TMP4700C	Externally provided (4,096 × 8)	256 × 4

FEATURES

- 4-bit single chip microcomputer with built-in ROM, RAM, input/output port, divider, timer/counter, and serial port.

- Instruction execution time : 2 μs (at 4 MHz clock)

- Effective instruction set
  90 instructions, Software compatible in the seires

- Subroutine nesting : Maximum 15 levels

- 6 interrupts (External : 2, Internal : 4)
  Independently latched control and multiple interrupt control

- Input/Output port (35 pins)
Input	1 port	4 pins
Output (corresponding to PLA)	2 ports	8 pins
I/O	4 ports	16 pins
I/O (Note)	2 ports	7 pins

  Note : These I/O ports are also used for the interrupt input, timer/counter input, and serial port; therefore, it is programmably selectable for each application.

- PLA data converting function (Instruction)
  Output of data to output port (8-bit)

- Table look-up and table search function (Instruction)
  Table can be set up in the whole ROM area.

- 12-bit timer/counter (2 channels)
  Event counter, timer, and pulse width measurement mode is programmably selectable.

- Serial port with 4-bit buffer
  Receive/Transfer mode is programmably selectable.
  External/Internal clock and Leading/Trailing edge mode are programmably selectable.

- 18-stage divider (with 4-stage precaler)
  Frequency applied for timer interrupt of divider is programmably selectable.

- High output current (Output ports)
  TYP. 20mA × 8 bits, LED direct drive is available.

- Memory stand-by function : Battery backup is available.

- On chip oscillator

- TTL/CMOS Compatible

- +5V single power supply

- 42-pin DIL plastic package

- N-channel Si gate E/D MOS LSI

PIN CONNECTIONS (TOP VIEW)
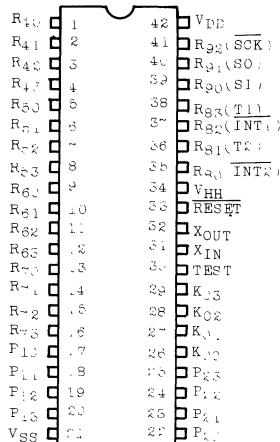
R_{40}	1	42	V_{DD}
R_{41}	2	41	R_{92} ($\overline{SCK}$)
R_{42}	3	40	R_{91} (SO)
R_{43}	4	39	R_{90} (SI)
R_{50}	5	38	R_{83} (T1)
R_{51}	6	37	R_{82} ($\overline{INT1}$)
R_{52}	7	36	R_{81} (T2)
R_{53}	8	35	R_{80} ($\overline{INT2}$)
R_{60}	9	34	V_{HH}
R_{61}	10	33	$\overline{RESET}$
R_{62}	11	32	X_{OUT}
R_{63}	12	31	X_{IN}
R_{70}	13	30	TEST
R_{71}	14	29	K_{03}
R_{72}	15	28	K_{02}
R_{73}	16	27	K_{01}
P_{10}	17	26	K_{00}
P_{11}	18	25	P_{23}
P_{12}	19	24	P_{22}
P_{13}	20	23	P_{21}
V_{SS}	21	22	P_{20}

PIN NAMES AND PIN DESCRIPTION

Pin Name	No.of pins	Input/Output	Function
$K_{03} \sim K_{00}$	4	Input	Input port
$P_{13} \sim P_{10}$	4	Output	Output port (Corresponding to PLA)
$P_{23} \sim P_{20}$	4	Output	" (")
$R_{43} \sim R_{40}$	4	I/O	I/O port
$R_{53} \sim R_{50}$	4	I/O	"
$R_{63} \sim R_{60}$	4	I/O	"
$R_{73} \sim R_{70}$	4	I/O	"
R_{83} (T1)	1	I/O	I/O port or timer/counter input
R_{82} ($\overline{INT1}$)	1	I/O	I/O port or interrupt input
R_{81} (T2)	1	I/O	I/O port or timer/counter input
R_{80} ($\overline{INT2}$)	1	I/O	I/O port or interrupt input
R_{92} ($\overline{SCK}$)	1	I/O	I/O port or shift clock for serial port
R_{91} (SO)	1	I/O	I/O port or serial output
R_{90} (SI)	1	I/O	I/O port or serial input
X_{IN}, X_{OUT}	2	Input, Output	Resonator connection terminals
$\overline{RESET}$	1	Input	Initialize signal input
TEST	1	Input	(Low level is input.)
V_{DD}	1	Power supply	+5V
V_{HH}	1	Power supply	+5V (Memory power supply)
V_{SS}	1	Power supply	0V

BLOCK DIAGRAM
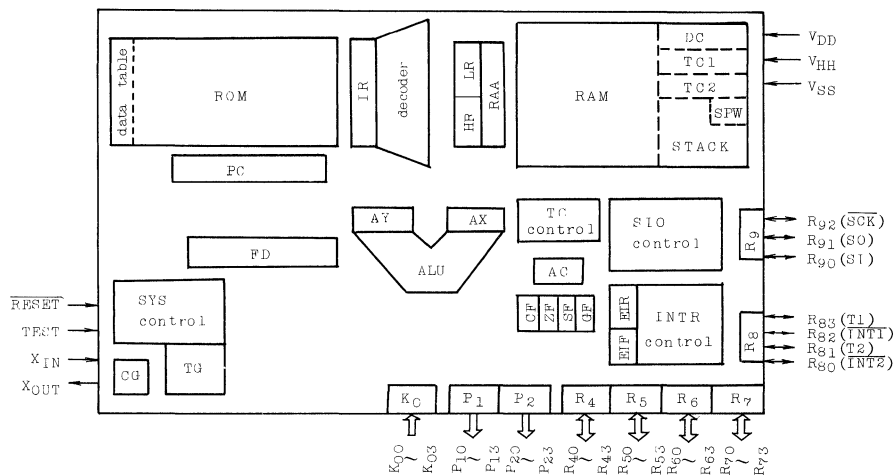
BLOCK NAMES AND DESCRIPTION

Block Name	Function
PC	Program counter (12 bits)
ROM	Program memory (including fixed data)
IR, decoder	Instruction register, Decoder
HR, LR	H register (page assignment of RAM), L register (address assignment in RAM page), (each 4-bit register)
RAA	RAM address buffer register (8 bits)
RAM	Data memory
STACK	Save area of program counter and flags (RAM area)
SPW	Stack pointer word (RAM area)
DC, data table	Data counter (12 bits, RAM area), Data table (ROM area).
AX, AY	Temporary register of ALU input
ALU	Arithmetic and logic unit
AC	Accumulator
FLAG (CF,ZF,SF,GF)	Flags
K, P, R	Ports
INTR Control	Interrupt control (EIF: Enable interrupt master F/F, EIR: Enable interrupt register)
FD	Frequency divider (4-stage prescaler + 18 stages)
TC_1, TC_2	12-bit timer/counter 2 channels (RAM area)
TC control	Timer/counter control
SIO control	Serial port control
SYS control	Generation of various internal control signals
CG, TG	Clock generator, Timing generator

260

FUNCTIONAL DESCRIPTION

1. System Configuration

    1. Program Counter (PC)

    2. Program Memory (ROM)

    3. H Register (HR), L Register (LR), RAM Address Buffer Register (RAA)

    4. Data Memory (RAM)

        (1) Stack (STACK)

        (2) Stack Pointer Word (SPW)

        (3) Data Counter (DC)

    5. ALU, Accumulator (AC)

    6. Flags (FLAC)

    7. Ports (PORT)

    8. Interrupt Control Circuit (INTR)

    9. Frequency Divider (FD)

    10. Timer/Counter (TC$_1$, TC$_2$)

    11. Serial Port (SIO)

Concerning the above component parts, the configuration and functions of hardwares are described :

Hexadecimal notation is used for the description, charts, and tables in order to indicate the address and the like, without assigning identification symbols as far as it does not give rise to confusion.

The following names and symbols are used unconsciously.

   (a) CPU   Control Processing Unit except for the built-in peripheral
                 circuitry, such as interrupt control circuit, timer/counter,
                 and serial port.

   (b) CP   Clock pulse generated in the clock oscillator.
                 It is called the "basic clock" or merely "clock".

   (c) fc   Indicates the frequency of the clock oscillator, namely, the
                 frequency of the basic clock.

   (d) MSB/LSB Indicates Most/Least Significant Bit.

   (e) F/F   Indicates Flip/Flop.

1.1 Program Counter (PC)

It is a 12-bit binary counter, and the contents of the program counter indicate the address of program memory in which the next instruction to be executed is stored.

The program counter generally gains increment at every instruction fetch by the number of bytes assigned to the instruction. However, when executing the branch and subroutine instructions or receiving the interrupt, the values specified by these instructions and operation are set.
Value "0" is specified by initializing the program counter.

The page structure of program memory is made with 64 words per page. The TMP4740P has 64 pages and the TMP4720P 32 pages.

At the execution of (BSS a) instruction, the value assigned by the instruction is set in the lower 6 bits of the program counter when the branch condition is met. That is, the (BSS a) instruction is used as a branch or jump instruction within a page. If the (BSS a) instruction is stored in the last address of the page, the value in the higher 6 bits of the program counter indicates that the branch or jump instruction to the next page is executed.

At the execution of (CALL a) instruction, the value specified by the instruction is set in the program counter after the previous contents of the program counter has been saved in the stack. Since 11 bits are of the address bit length which can be assigned by the instruction, the call address of subroutine should be in the range of addresses 000 - 7FF.

MSB   LSB

PC$_H$				PC$_M$				PC$_L$			
PC$_{11}$	PC$_{10}$	PC$_9$	PC$_8$	PC$_7$	PC$_6$	PC$_5$	PC$_4$	PC$_3$	PC$_2$	PC$_1$	PC$_0$

Page assignment   Address assignment in page

(a) Configuration of Program Counter

(b) Configuration of ROM

(c) Special example of branch caused by (BSS a) instrustion.

Fig.1.1.1   Program Counter and Program Memory (ROM)

263

1.2 Program Memory (ROM)

Processing programs and fixed data are stored in the program memory. The next instruction to be executed is read out from the address indicated by the contents of the program counter.

The fixed data stored in the program memory can be read by using the ROM data referring instruction or the PLA referring instruction. The ROM data referring instruction reads out the higher or lower 4-bit data of the fixed data stored in the address decided by the data counter [(LDH A, @DC+) and (LDL A, @DC) instruction respectively], and stores the data in the accululator. The PLA referring instruction (OUTB @HL) reads out the fixed data (8-bit) stored in the address decided by the contents of the data memory indicated by the contents of H and L registers as well as contents of the carry flag, and outputs the data to output ports (P2 · P1).

Addresses are individually assignged to the program memory and data memory, so that the fixed data in the ROM area cannot be directly read out by the address of the data memory.

Specific Addresses of Program Memory

The following addresses of the program memory are used for specific purposes. When not used for these purposes, the specific addresses can be used to store the processing programs and fixed data.

Specific Address	Specific Purposes
000 (001)	Start address by initialization
002 (003)	INT1 Interrupt vector address
004 (005)	ISIO Interrupt vector address
006 (007)	IOVF1 Interrupt vector address
008 (009)	IOVF2 Interrupt vector address
00A (00B)	ITMR Interrupt vector address
00C (00D)	INT2 Interrupt vector address
$8n + 6$ (n = 1 ∿15) 086 (Note)	Call address by instruction (CALLS a)
FE0 ∂ FFF	PLA data conversion table

Note :   086 (hexadecimal) = 134 (decimal)

Table 1.2.1   Specific Address of Program Memory

Toshiba
東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

ROM CAPACITY

The TMP4740P and TMP4720P contain a program memory with 4,096 x 8-bit
(addresses 000 - FFF) capacity and 2,048 x 8-bit (addresses 000 - 7FF) capacity,
respectively. But the TMP4720P contains a program counter with 12-bit length.
Therefore, when one of addresses 800 - FFF is accessed in a program, the ROM
data corresponding to addresses 000 - 7FF read out. It is because there is
no physical ROM in addresses 800-FFF, but the MSB in the program counter is
not decoded. For example, when the data located in address FF3 is output to
a port by the PLA referring instruction on a program, the data located in
address 7F3 is read out. In the TMP4720P, the PLA data conversion table
(addresses FE0 - FFF) is, therefore, located in addresses 7E0 - 7FF.

"0" [(NOP) instruction] is read out for the ROM data within the range
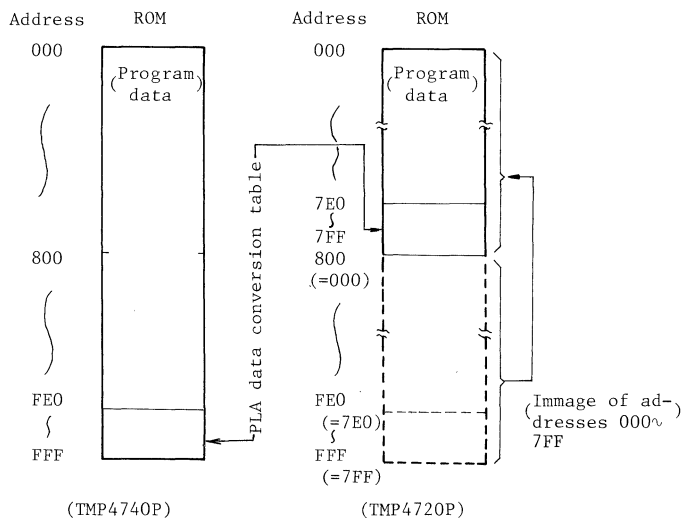of the built-in ROM capacity, if it is not specified by the user.

Fig. 1.2.1 ROM Capacity and Address

266

1.3 H Register (HR), L Register (LR), and RAM Address Buffer Register (RAA)

The H and L registers are 4-bit registers used as the data memory address pointers or general purpose registers.

The page structure of the data memory is based on 16 words per page. Pages are specified by H register, and addresses in page are done by L register, respectively. TMP4740P has 16 pages and TMP4720P 8 pages.

The L register is also used to specify the bits corresponding to pins $R_{73} \sim R_{40}$ of the I/O port when instructions (SET @L), (CLR @L), and (TEST @L), are executed.

The RAM address buffer register is a temporary register used to specify the address in the data memory, and serves as an input of the RAM address decoder. Normally, the data specified by the contents of the H and L registers or immediate data of an instruction is fed into the RAM address buffer register.
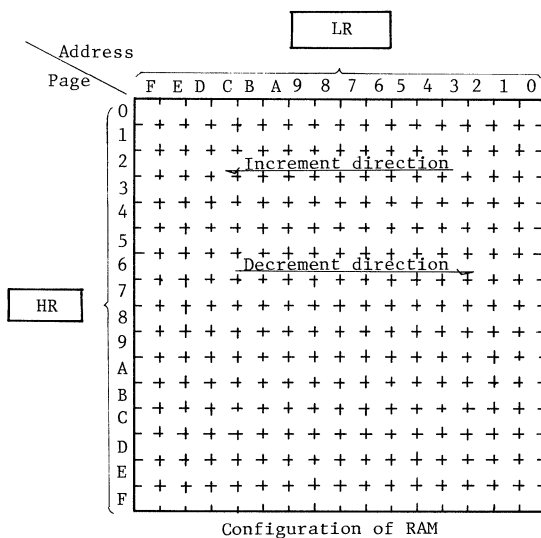
MSB   LSB

HR				LR			
HR_3	HR_2	HR_1	HR_0	LR_3	LR_2	LR_1	LR_0

Page specification   Address specification in page

Address

Page

LR

F E D C B A 9 8 7 6 5 4 3 2 1 0

```
0
1
2        Increment direction
3
4
5
6        Decrement direction
7
8
9
A
B
C
D
E
F
```

HR

Configuration of RAM
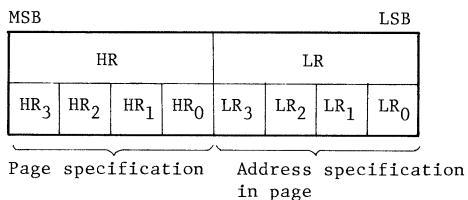
Fig. 1.3.1 H Register, L Register and Data Memory (RAM)

268

INTEGRATED CIRCUIT

TECHNICAL DATA

Toshiba
東芝

TMP4740P
TMP4720P

PRELIMINARY

1.4 Data Memory (RAM)

The processing data of user are stored in the data memory.
The data is read out or written in according to the address indicated
by the contents of the RAM address buffer register.

| Specific addresses of data memory |

The data memory is also used for the following specific purposes.
When it is not used for the respective purposes, the RAM of the
corresponding address can be used to store the user processing data.

(1) Stack (STACK)
(2) Stack pointer word (SPW)
(3) Data counter (DC)
(4) Timer/Counter (TC1, TC2)

(1) Stack (STACK)

The stack, which is contained in the data memory (one level of
the stack consists of 4-word RAM), is area to save the contents of
the program counter (return address) and flag prior to jumping to the
processing program at time of subroutine call or interrupt acceptance.
To return from the processing program, (RET) instruction is used to
restore the contents saved in the stack to the program counter, and
(RETI) instruction is used to restore the contents saved in the stack
to the program counter and flags.

The location of the stack to save/restore the contents is deter-
mined by the stack pointer word, which is automatically decremented
after the saving operation, and incremented prior to the restoring
operation.
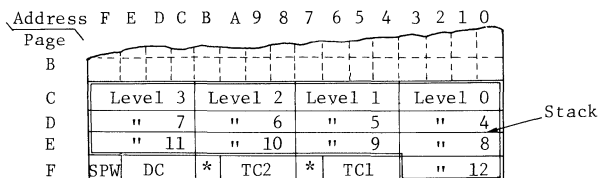
269

(2) Stack Pointer Word (SPW)

The address FF in the data memory is called a stack pointer word and decides the stack pointer. The stack is contained in the RAM, and accessed by the stack pointer.

The stack pointer is decided with the format shown in Fig. 1.4.1, but this address indicates the lower RAM address in each level of the stack.

Values "E" - "0" can be assigned for the stack pointer word, so that the maximum of 15 nesting levels are available for the stack. However, when the timer/counter mentioned following is used, the level containing the RAM address corresponding to the timer/counter cannot be used for the stack (value "F" is not assigned to the stack pointer word, because the stack contains the RAM address corresponding to the stack pointer word). The stack pointer word is automatically updated by the subroutine call or interrupt acceptance; however, it cannot exceed the allowable size of the stack for the system configuration.

Since the stack pointer word is never initialized in terms of hardware, it is necessary to set it to the highest possible level of the stack in the user's initialization prpgram. For instance, it is set to "C" level when the two channels of timer/counter are used.

Note: The "level" indicates the depth of the nesting in the stack as well as the location of the next available stack. That is, it represents the contents of the stack pointer word.
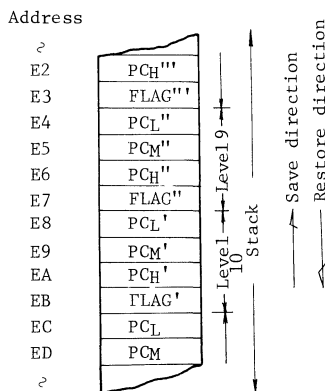
Address F E D C B A 9 8 7 6 5 4 3 2 1 0
Page

B								
C	Level 3		Level 2		Level 1		Level 0	
D	" 7		" 6		" 5		" 4	Stack
E	" 11		" 10		" 9		" 8	
F	SPW DC	*	TC2	*	TC1		" 12	

* : Can be used to store the user processing data

(a)　Specific purposive map of RAM

$\begin{pmatrix}\text{Stack pointer word}\\ \text{RAM address : FF}\end{pmatrix}$

MSB   LSB
3   2   1   0

SPW$_3$	SPW$_2$	SPW$_1$	SPW$_0$

MSB   LSB
7   6   5   4   3   2   1   0

(Stack pointer)

1	1	SPW$_3$	SPW$_2$	SPW$_1$	SPW$_0$	0	0
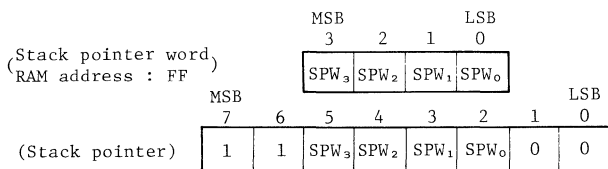
(b)　Stack pointer and stack pointer word

Address

E2	PC$_H$'''
E3	FLAG'''
E4	PC$_L$''
E5	PC$_M$''
E6	PC$_H$''
E7	FLAG''
E8	PC$_L$'
E9	PC$_M$'
EA	PC$_H$'
EB	FLAG'
EC	PC$_L$
ED	PC$_M$

Level 9
Level 10
Stack

Save direction
Restore direction

(c)　Structure of stack

Fig. 1.4.1　Specific Address and Stack of Data Memory

271

(3) Data Counter (DC)

Data counter is a 12-bit binary counter used to specify the address when the data table in the ROM area is referred (ROM data referring instruction).

The RAM address with 4-bit unit is allocated to the data counter, so that the initial value setting and the content reading of the data counter can be executed by the RAM manipulative instructions.

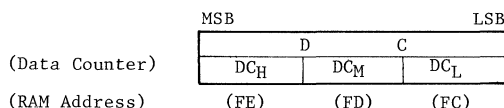|	MSB		LSB
		D	C
(Data Counter)	DC_H	DC_M	DC_L
(RAM Address)	(FE)	(FD)	(FC)

Fig. 1.4.2 Data Counter and RAM Address

(4) Timer/Counter (TC1, TC2)

The two channels of 12-bit timer/counter are built-in, and the RAM address with 4-bit unit is allocated to the timer/counter, so that the initial value setting and the content reading of the timer/counter can be executed by the RAM manipulative instructions.

When the timer/counter 1 is not used, the stack lower from level 13 can be used. When both of the timer/counter 1 and 2 are not used, the stack lower from level 14 can be used.

	MSB		LSB
		TC1	
(Timer/Counter 1)	$TC1_H$	$TC1_M$	$TC1_L$
(RAM Address)	(F6)	(F5)	(F4)

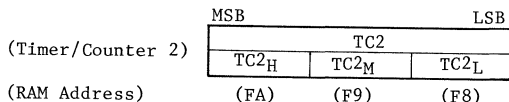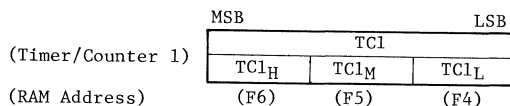|	MSB		LSB
		TC2	
(Timer/Counter 2)	$TC2_H$	$TC2_M$	$TC2_L$
(RAM Address)	(FA)	(F9)	(F8)

Fig. 1.4.3 Timer/Counter and RAM Address

(5) Page 0 in Data Memory

Page 0 in the data memory (addresses 00 - 0F) is effectively used as a flag or pointer in a user's program.

272

RAM Capacity

Data memory contained in TMP4740P has a 256 x 4-bit (addresses 00 - FF) capacity, and that contained in TMP4720P has a 128 x 4-bit (addresses 00 - 7F) capacity.

Since the TMP4720P also has the RAM address buffer register of 8-bit length, there is no physical RAM in addresses 80 - FF in the TMP4720P. However, the RAM equivalent to addresses 00 - 7F are referred when addresses 80 - FF are accessed in a program, because the MSB of RAM address buffer register is not decoded. That is, the specific RAM address is distributed to CO - FF in a program, but the RAM equivalent to addresses 40 - 7F are assigned in the TMP4720P.
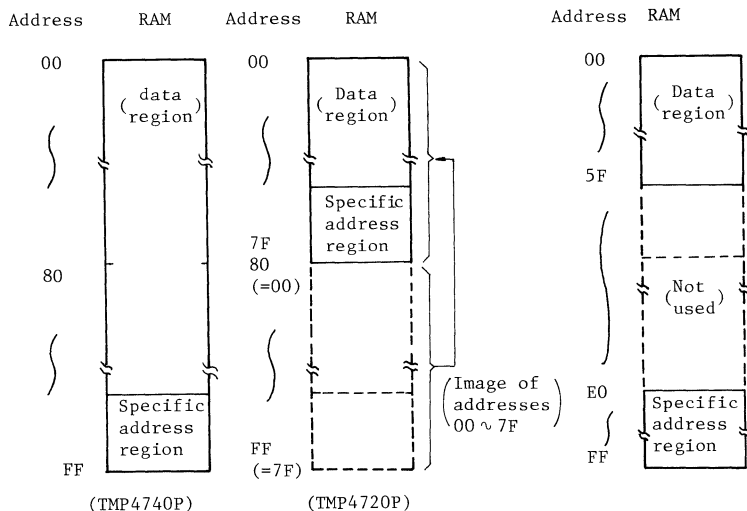
(a) RAM Capacity and Address   (b) RAM Map example of TMP4720P
$\left(\begin{array}{l}TC_1, TC_2 \text{ and stack}\\ 5 \text{ level are used.}\end{array}\right)$

Fig. 1.4.4 RAM Capacity and Address

273

1.5 ALU, Accumulator (AC)

The ALU is a circuit used for various arithmetic and logical
operation for 4-bit binary data. It performs the operation designated
by the instruction, and outputs the 4-bit result, carry (C), and
zero detection signal (Z).

The accumulator is a 4-bit register to use a source operand
for the arithmetic operation, and in which the result is stored.

```
        ┌────┐    ┌────┐
        │ AY │    │ AX │     C_in
        └────┘    └────┘
          ╲╲        ╲╲┐
           ╲    ALU    ╲
            ╲_____╱
        Z ──┐╲╲
        C ──┘ ╲╱   (AX and AY are temporary registers.)
        Output of result

              MSB        LSB
             ┌─3──2──1──0─┐
(Accumulator)│  │  │  │   │
             └──┴──┴──┴───┘
```
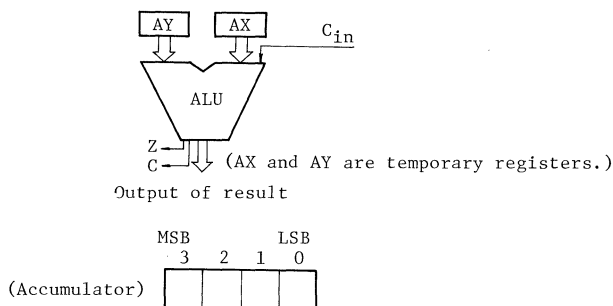
Fig. 1.5.1 ALU, Accumulator

Detection of operating condition

Output C from the ALU indicates the carry output from the most
significant position in the addition operation.
However, the subtraction is executed with the addition of the 2's com-
plement, so that output C in the subraction operation indicates the
"non-borrow" from the most significant position (i.e., in case of non-
borrow, C = "1"). Accordingly, borrow (B) can be represented with "$\overline{C}$".
Output Z indicates the zero detection signal to which "1" is applied
when all of the 4-bit data transferred to accumulator or output of the ALU
are cleared to zero.

Example   (4-bit operation)

(a)   4 + 5 = 9   (C = 0, Z = 0)

(b)   7 + 9 = 0   (C = 1, Z = 1)

(c)   3 - 1 = 2   (B = 0, Z = 0)

(d)   2 - 2 = 0   (B = 0, Z = 1)

(e)   6 - 8 = -2 or E   (B = 1, Z = 0)

Note : B = $\overline{C}$ is indicated.

1.6   Flag (FLAG)

Flag is a 4-bit register used to store the condition of arith-
metic operation, and of which the set/reset conditions are specified
by the instruction. The flag consisting of CF, ZF, SF, and GF is
saved in the stack when the interrupt is accepted. By executing the
(RETI) instruction, it is restored from the stack to the conditions
immediately before the interrupt is accepted.

3	2	1	0
CF	ZF	SF	GF

Fig. 1.6.1   Flag
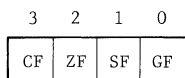
(1)   Carry Flag (CF)

This flag is used to hold the carry in the addition operation
as an input to the ALU by the (ADDC A, @HL) instruction as well as
to hold the non-borrow in the subtraction operation (the carry in the
addition of the 2's complement) as an input to the ALU by the
(SUBRC A, @HL) instruction. The rotate instruction makes the flag
hold the data shifted out of the accumulator.

(2) Zero Flag (ZF)

This flag is stored the zero detection signal (Z) when the instruction designate to change. "1" is set if all 4 bits are cleared to zero by an arithmetic operation or data processing.

(3) Status Flag (SF)

This flag is set or reset according to the condition specified by the instruction. With the exception of particular cases, it is usually presented at every execution of an instruction, and holds the contents of the result during execution of the next instruction. It is normally set to "1", but is reset to "0" for a time under the certain condition (it varies according to the instruction, for examples, when the result is zero, when carry occurs in the addition, or when borrow occurs in the subtraction, the flag is reset).

The status flag is referred to as branch condition in a branch instruction. The memory location is branched when this flag is set to "1"; therefore, normally the branch instruction can be required as "unconditional jump instruction". On the contrary, the instruction becomes a "conditional instruction" if it is executed immediately after loading the instruction to set/reset the status flag according to the condition determined by some previous instruction.

The status flag is initialized to "1" at initialization, and is also set to "1" after the contents have been saved in the stack when the interrupt is accepted. The contents saved in the stack is restored by the (RETI) instruction.

(4) General Flag (GF)

This is a single-bit general purpose flag, being set or reset, and also used in a test by a program. This can be used for any purpose in the user program.

1.7 Port (PORT)

Data transfer to/from the external circuitry, and command/status/data transfer between the built-in periferal circuitry are carried out by the input/output instructions.

(a) Input/Output port : Data transfer to/from external circuitry.

(b) Command/data output : Control of circuitry of built-in peripheral circuitry, and output of data.

(c) Status/data input : Input of status signal(Note) and data from the built-in peripheral circuitry.

Note : Status signal is provided from serial port and is different from the status flag (SF).

To transfer the data or to control the circuitry, each port or register is selected by designating the address (Port address) by input/output operational instructions (13 instructions) in the same way as the memory.
The port address is composed of 5 bits (addresses 0 - 31).
The address to be accessed differs according to a instruction.
By way of caution, the port address space is independent of the program memory address space and the data memory address space.

Every output port contains a latch in order to hold the output data. Since every input port is operated without latching, it is desired to externally hold the data to be input from the external devices till the data is completely read out, or to read the data several times to confirm the contents.

The details to specify the input/output circuit format of ports and initialization of the output latch are 2.6 (2) Input/Output Circuit Format.

Port address	Symbol (Input/ Output)	Port, Register (Input/Output)	Input/Output Instructions						
			IN %P, A / IN %P,@HL	OUT A ,%P / OUT@HL,%P	OUT#K,%P	OUTB @HL	SET%P,b / CLR%P,b	TEST %P,b / TESTP%P,b	SET @L / CLR @L / TEST @L
00	IP00/OP00	K₀ Input port / $\overline{P_1}$ Output port	0					0	
01	IP01/OP01	P₁Output latch/	0	0	0		0	0	
02	IP02/OP02	P₂ " /P₂ "	0	0	0		0	0	
03	IP03/OP03								
04	IP04/OP04	R₄ I/O port	0	0	0		0	0	0
05	IP05/OP05	R₅ "	0	0	0		0	0	0
06	IP06/OP06	R₆ "	0	0	0		0	0	0
07	IP07/OP07	R₇ "	0	0	0		0	0	0
08	IP08/OP08	R₈ "	0	0	0		0	0	
09	IP09/OP09	R₉ "	0	0	0		0	0	
0A	IP0A/OP0A	———							
0B	IP0B/OP0B	———	(*) Serial buffer register (Reception)						
0C	IP0C/OP0C	———	(**) Serial buffer register (Transmission)						
0D	IP0D/OP0D	———							
0E	IP0E/OP0E	Status input/ ———	0					0	
0F	IP0F/OP0F	(*) / (**)	0	0	0				
10	/OP10	/ ———							
11	/OP11	/P₂·P₁ output port (8-bit output)				0			
12	/OP12	/ ———							
13	/OP13	/ ———							
14	/OP14	/ ———							
15	/OP15	/ ———							
16	/OP16	/ ———	(a) Control with timer interrupt of divider						
17	/OP17	/ ———	(b) Timer/Counter 1 control						
18	/OP18	/ ———							
19	/OP19	/ (a)			0				
1A	/OP1A	/ ———	(c) Timer/Counter 2 control						
1B	/OP1B	/ ———	(d) Serial port control						
1C	/OP1C	/ (b)			0				
1D	/OP1D	/ (c)			0				
1E	/OP1E	/ ———							
1F	/OP1F	/ (d)			0				

Note 1: Inputs (IP10 - IP1F) of port addresses 10 - 1F remain undefined.
Note 2: Port addresses with "——" mark are reserved addresses and cannot be used at user's program.
Note 3: OP11 is automatically accessed by (OUTB @HL) instruction, but cannot be done by the instructions other than this one.

Table 1.7.1 Port Address Allocation and Input/Output Instructions

Toshiba
東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

(1) K_0 $(K_{03} \sim K_{00})$ Port

This is a 4-bit port used for input.

Input data $\longleftarrow \vartriangleleft \quad \square$ (Pin)

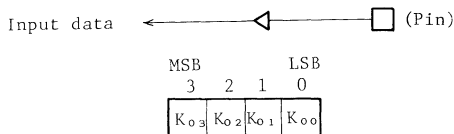| MSB			LSB
3	2	1	0
K_{03}	K_{02}	K_{01}	K_{00}

Fig. 1.7.1 K_0 Port

(2) P_1 $(P_{13} \sim P_{10})$, P_2 $(P_{23} \sim P_{20})$ Port

These ports are 4-bit ports with a latch used for
output. The latch data can be read by the instruction.

These two ports can independently access by specifying port ad-
dresses IPO1/OPO1, and IPO2/OPO2. In addition, they can output 8-bit
data by the (OUTB @HL) instruction.

PLA data conversion

A hardware PLA is not contained in the system; however, the func-
tion equivalent to it can be performed by access to the PLA data con-
version table provided in the RCM by use of the (CUTB @HL) instruction.

The PLA referring instruction (OUTB @HL) : This instruction reads out
the 8-bit data stored in the program memory, whose address is determined
by the contents of the data memory indicated by the contents of the H and
L registers as well as the contents of the carry flag, and outputs the
data to 8-bit ports P2 and P1. At this time OP11 is automatically
selected as the port address.

Toshiba

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

Ports P1 and P2 are capable of reading the latch data by the instruction, so that the data output by the PLA referring instruction can be qualified or modified; that is, the convert pattern can be changed or the numbers of pattern will be increased.
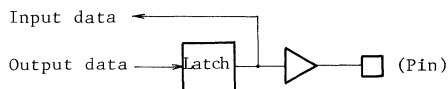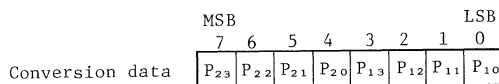
Conversion data

MSB							LSB
7	6	5	4	3	2	1	0
P_{23}	P_{22}	P_{21}	P_{20}	P_{13}	P_{12}	P_{11}	P_{10}

Input data

Output data ──►Latch─── ▷ ─── □ (Pin)

Fig. 1.7.2 P_1 and P_2 Ports

(3) $R_4(R_{43} \sim R_{40})$, $R_5(R_{53} \sim R_{50})$, $R_6(R_{63} \sim R_{60})$, $R_7(R_{73} \sim R_{70})$ Port

Each of these ports is a 4-bit I/O port with a latch. The latch should be set to "1" when the port is used as an input port.

Pins $R_{73} - R_{40}$ can be used for bit scanning for set/reset and test according to the contents of the L register by executing the (SET @L), (CLR @L) and (TEST @L) instructions. Table 1.7.2 shows the pins corresponding to the contents of the L register.

L register 3 2 1 0	Correspond-ing Pin	L register 3 2 1 0	Correspond-ing Pin
0 0 0 0	R_{40}	1 0 0 0	R_{60}
0 0 0 1	R_{41}	1 0 0 1	R_{61}
0 0 1 0	R_{42}	1 0 1 0	R_{62}
0 0 1 1	R_{43}	1 0 1 1	R_{63}
0 1 0 0	R_{50}	1 1 0 0	R_{70}
0 1 0 1	R_{51}	1 1 0 1	R_{71}
0 1 1 0	R_{52}	1 1 1 0	R_{72}
0 1 1 1	R_{53}	1 1 1 1	R_{73}

Table 1.7.2 Correspondence of Individual Bits of L Register and I/O Port

Input data
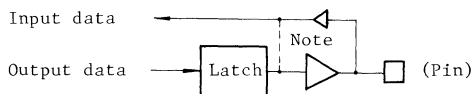
Output data — Latch

Note : For bit set/reset of port,
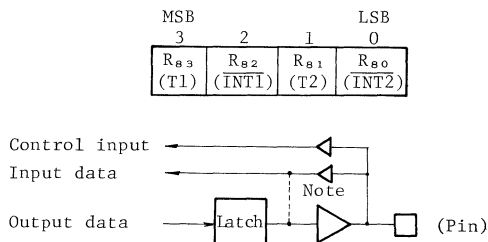latch output serves as input data.

Fig. 1.7.3 $R_4 \sim R_7$ Ports

(4) R_8 ($R_{83} \sim R_{80}$) Port

This is a 4-bit I/O port with a latch. The latch should be set
to "1" when the port is used as an input port.

It is a port common to external interrupt input or external timer/
counter input. When it is driven by the external circuitry, such as
external interrupt input or external timer/counter input, the latch
must be set to "1". When it is used as normal I/O port, some measures,
such as inhibition of the external interrupt input acceptance or disable
of the mode depending on the external input of the timer/counter should
be taken in a program.

(Note)   When pin R_{82} ($\overline{INT1}$) is used as a port, INT1 interrupt request
takes place because the falling edge of the pin input/output
is detected (interrupt enabling master F/F is normally set to
"1"). This causes the CPU to process a dummy interrupt
acceptance [e.g. the (RETI) instruction only is executed].
When pin R_{80} ($\overline{INT2}$) is used, INT2 interrupt request also
takes place in the same manner as the case of pin R_{82}, but
the interrupt request is not accepted by merely resetting
the LSB (EIR_0) of the enable interrupt register to "0" in
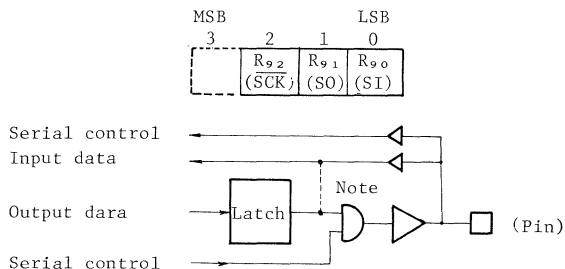advance. Therefore, the above processing is not required.

INTEGRATED CIRCUIT

Toshiba 東芝

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

MSB			LSB
3	2	1	0
R_{83}	R_{82}	R_{81}	R_{80}
(T1)	($\overline{\text{INT1}}$)	(T2)	($\overline{\text{INT2}}$)

Control input

Input data

Note

Output data — Latch — (Pin)

Note: For bit set/reset of port, latch
output serves as input data.

Fig. 1.7.4 R_8 Port

(5) $R_9 (R_{92} \sim R_{90})$ Port

This is a 3-bit I/O port with a latch, and the latch must be
set to "1" when it is used as input port.
The R_9 port is also used as serial port. The latch must be set to
"1" when R_9 port is used as serial port. The port used as normal
I/O port is not entirely influenced by disabling the serial port.
Pin R_{93} is not mounted in the port, but "1" is read by accessing to
pin R_{93} in a program.

MSB		LSB
3	2	1	0
	R_{92}	R_{91}	R_{90}
	($\overline{\text{SCK}}$)	(SO)	(SI)

Serial control

Input data

Note

Output dara — Latch — (Pin)

Serial control

Note: For bit set/reset of port, latch
output serves as input data.

Fig. 1.7.5 R_9 Port

1.8 Interrupt control circuit (INTR)

Interrupt factors are composed of two from the external circuitry, and four from the internal circuitry. By setting the interrupt latch provided for each factor, an interrupt request is generated to the CPU. The interrupt latch is set when the edge of the input signal is detected.

The interrupt request is not always accepted by the CPU if generated. It is not accepted till the priority in the six factors determined according to the hardware and the enabling/disabling control by the program become all affirmative.

In order to control enabling/disabling of interrupt by the program, an F/F (EIF) and a 4-bit register (EIR) are provided. By using these means, preferential acceptance of the interrupt factors by the program, and multiple interrupt control can be realized.

Factor		Priority according to hardware	Interrupt Latch	Enable condition according to program	Vector Address
External interrupt 1 (INT1)		(Higher) 1	$INTL_5$	(Note 1) $EIF = 1$	002
Internal interrupt	Serial Input/Output interrupt (ISIO)	2	$INTL_4$	$EIF \cdot EIR_3 = 1$	004
	Timer counter 1 Overflow interrupt (IOVF1)	3	$INTL_3$	$EIF \cdot EIR_2 = 1$	006
	Timer counter 2 Overflow interrupt (IOVF2)	4	$INTL_2$	(Note 2) $EIF \cdot EIR_1 = 1$	008
	Timer interrupt of divider (ITMR)	5	$INTL_1$	(Note 2) $EIF \cdot EIR_1 = 1$	00A
External interrupt 2 (INT2)		6 (Lower)	$INTL_0$	$EIF \cdot EIR_0 = 1$	00C

Interrupt enabling master F/F   Interrupt enabling register (EIR)

| EIF |

MSB   LSB
3   2   1   0
| EIR_3 | EIR_2 | EIR_1 | EIR_0 |

(Note 1) Since EIR register cannot make disabling of the INT1
          interrupt, this interrupt is always accepted under the
          interrupt enabled condition (EIF = 1). Therefore, this
          should be used for the interrupt requiring the first
          priority such as emregency interrupt.

(Note 2) The given acceptance condition by the program is the same
          in IOVF2 and ITMR; accordingly, the action of these inter-
          rupts to the acceptance/inhibition control is the same.
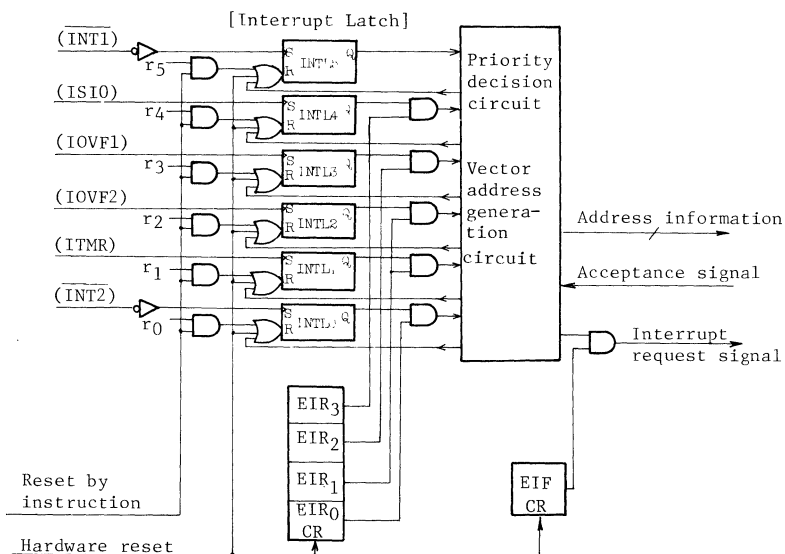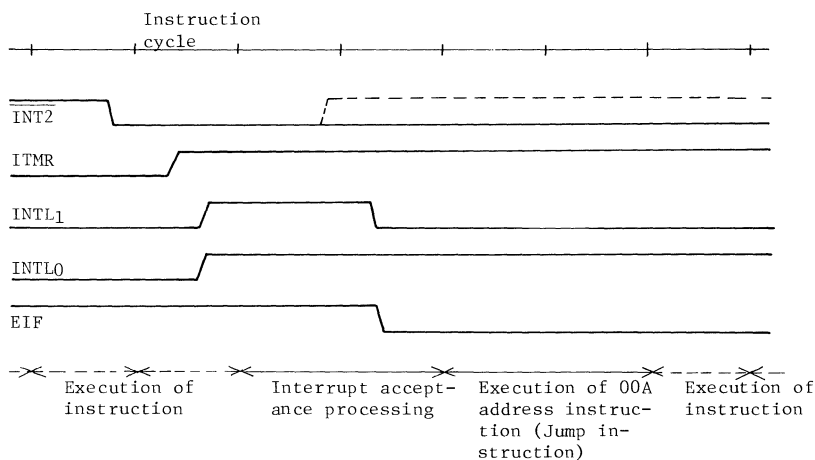
Table 1.8.1 Interrupt Factors

Fig. 1.8.1   Interrupt Control Circuit

Note: On the assumption that $EIR_1 = 1$, without other interrupt requests

Fig. 1.8.2   Interrupt Acceptance Timing Chart (Example)

INTEGRATED CIRCUIT

Toshiba
東芝

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

(1) Interrupt processing

The interrupt request signal to be sent to the CPU is held by the interrupt latch till the request is accpeted or the latch is reset by the initialization operation or instruction.

The processing for the interrupt acceptance is performed within two instruction cycle time after the completion of the execution of instruction (after the completion of the timer/counter processing if it is required).

The following operations are performed by the interrupt service program.

① The contents of the program counter and flag are saved in the stack.

② The vector address is set to the program counter according to the interrupt factor.
(A jump instruction to each interrupt service program is usually stored in the program memory corresponding to the vector address.)

③ The status flag is set to "1".

④ The interrupt enabling master F/F is reset to "0" to inhibit the subsequent interrupt acceptance for a time.

⑤ The interrupt latch of the accepted interrupt factor is reset to "0".

⑥ The instruction stored in the vector address is executed.

The interrupt service program terminates after the execution of the (RETI) instruction.
The following operations are performed by the (RETI) instruction.

① The contents of the program counter and flag are restored out of the stack.

② The interrupt enabling master F/F is set to "1".

When the multiple interrupt is accepted, the interrupt enabling master F/F should be set by the instruction. At this time, the enabling/disabling for each interrupt factor can be changed by updating the interrupt enabling register by the (XCH A, EIR) instruction.

The program counter and flag are automatically saved/restored in the interrupt processing. However, if saving/restoring of the accumulator and other registers is necessary, it should be designated by a program.

(2) Interrupt control by program

EIF

This is an enabling interrupt master F/F. Interrupt is put in the interrupt acceptance enabling state by setting the EIF to "1". It is reset to "0" immediately after having accepted an interrupt to inhibit the subsequent interrupt acceptance for a time, but is set to "1" again by the (RETI) instruction after the completion of the interrupt service program to return the enable state again. And then the other interrupt can be received.

The EIF can be set/reset in a program by using the (EICLR IL,r) and (DICLR IL, r) instructions. It is reset to "0" at initialization operation.

EIR register

This is a 4-bit register used for selection/control of enabling/disabling of the interrupt acceptance in a program.

Read/write operation is performed by use of the (XCH A, EIR) instruction. It is set to "0" at the initialization operation.

Interrupt latch

The interrupt latches ($INTL_5$ - $INTL_0$) provided for each interrupt factor are set by the rising edge of the input signal if the interrupt is caused by the internal factors, and are set by the falling edge of the input pin if it is caused by the external factors. Then, interrupt request signal is sent to the CPU. The interrupt latch holds the signal till the interrupt request is accepted, and is reset to "0" immediately after the interrupt has been accepted.

Since the interrupt latch can be reset to "0" by the (EICLR IL, r), (DICLR IL, r) and (CLR IL, r) instructions, the interrupt request signal can be initialized by a program. The latch is reset to "0" at the initialization operation.

1.9 Frequency divider (FD)

The divider (FD_1 - FD_{18}) is made up 18-stage binary counter, and its output is used to generate various internal timing.

The basic clock (fc Hz) is divided into sixteen by the timing generator and input to the divider; therefore, the output frequency at the last stage is $fc/2^{22}$ Hz.
It is reset to "0" at the initialization operation.

Timer Interrupt of divider (ITMR)

The divider is capable of sending the interrupt request for a certain frequency. Four different frequencies can be selected for timer interrupt by instructions.

The command register is accessed as port address OP19, and is reset to "0" at time of the initialization.

The timer interrupt of divider is caused from the rising edge of the first output of the divider after the data has been written in the command resister.

Basic clock (CP) (fc Hz) → Prescaler (TG) → (fc/16 Hz) → FD 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 → (fc/2^{22}Hz)
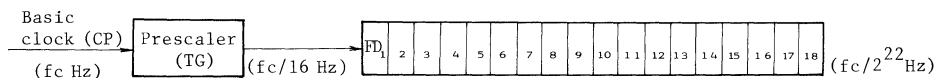
(a) Structure of frequency divider

$\binom{\text{Port address}}{\text{OP19}}$

MSB   LSB
3   2   1   0

(*: don't care)

*	0	*	*	:	Disable
*	1	0	0	:	Interrupt frequency	fc/2^{10}Hz
*	1	0	1	:	"	fc/2^{11}Hz
*	1	1	0	:	"	fc/2^{12}Hz
*	1	1	1	:	"	fc/2^{13}Hz

Interrupt fre-quency (Hz)	For example, fc=4.194304MHz
fc/2^{10}	4,096 Hz
fc/2^{11}	2,048 Hz
fc/2^{12}	1,024 Hz
fc/2^{13}	512 Hz

(b) Command register

Fig. 1.9.1 Frequency Divider

1.10   Timer/Counter (TC_1, TC_2)

Two channels of 12-bit binary counter is contained to count time or event.

Since the RAM address with 4-bit unit is allocated to the timer/counter, the initial value setting and the content reading of the timer/counter can be executed by the RAM manipulated instructions.

	MSB		LSB
(Timer/Counter 1)		TC1	
	TC1H	TC1M	TC1L
(RAM Address)	(F6)	(F5)	(F4)

	MSB		LSB
(Timer/Counter 2)		TC2	
	TC2H	TC2M	TC2L
(RAM Address)	(FA)	(F9)	(F8)
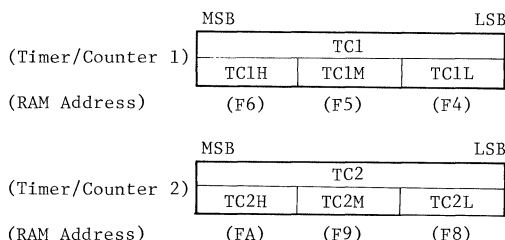
Fig. 1.10.1   Timer/Counter

(1)   Timer/Counter Control

The timer/counter is controlled by the command specifying the operation mode. The command register for the timer/counter 1 and timer/counter 2 is accessed as port addresses OP1C and OP1D, respectively. It is reset to "0" at the initialization operation. The count operation is started from the first rising edge of the count pulse applied by setting the value (mode) to the command register.

When the timer/counter is not used, the RAM addresses corresponding to the timer/counter can be used to store the user processing data by selecting the "disable" state. In the timer mode, the external input pins can be used as I/O ports [R_{83} (T1), R_{81} (T2)].

Toshiba

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

TC1 Command register
(Port Address OP1C)

MSB   LSB
3   2   1   0

TC2 Command register
(Port Address OP1D)

MSB   LSB
3   2   1   0

Selection of Count Mode

00 : Disable state

01 : Event count mode
     (Counts external input)

10 : Timer mode
     (Counts internal pulse.)

11 : Pulse width measurement mode
     (Counts the pulse obtained by
     sampling of external input by
     use of internal pulse.)

Selection of Internal Pulse
(Divider Output) Rate

00 : $fc/2^{10}$ Hz is counted.

01 : $fc/2^{14}$ Hz is counted.

10 : $fc/2^{18}$ Hz is counted.

11 : $fc/2^{22}$ Hz is counted.

(fc: Basic clock frequency)

(a) Command register

Internal Pulse Rate (Hz)	Max. Setting Time (SEC)	For example, fc=4.194304 MHz	
		Internal Pulse Rate (Hz)	Max. Setting Time (SEC)
$fc/2^{10}$	$2^{22}/fc$	4,096	1
$fc/2^{14}$	$2^{26}/fc$	256	16
$fc/2^{18}$	$2^{30}/fc$	16	256
$fc/2^{22}$	$2^{34}/fc$	1	4,096

(b) Selection of timer rate

        Fig. 1.10.2 Control of Timer/Counter

(2) Count Operation

When the rising edge of the count pulse is detected, the count latch is set to send a count request to the CPU.

The count operation of the timer/counter is performed requiring one instruction cycle time after completion of the instruction execution. The execution of the next instruction and the acceptance of the interrupt request are kept waiting during the operation. When the count request is sent from the timer/counter 1 and 2, at the same time, the count request of the timer/counter 1 is preferentially executed.

The maximum frequency applied to the external input pin under the event counter mode is fc/32 Hz if one channel is used. When two channels are used, fc/32 Hz is applied to the timer/counter 1, and fc/40 Hz to the timer/counter 2.

In the timer mode, the maximum frequency is determined by a command.
The maximum frequency applied to the external input pin in the pulse width measurement mode should be the frequency level available for analyzing the count value in the program. Normally, the frequency sufficiently slower than the designated internal pulse rate is applied to the external input pin.

INTEGRATED CIRCUIT

東芝 Toshiba

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

Count pulse

Counter

| n | n + 1 | n + 2 | n + 3 |

(a) Event counter/timer mode

External input pin

Internal pulse

Count pulse

Counter

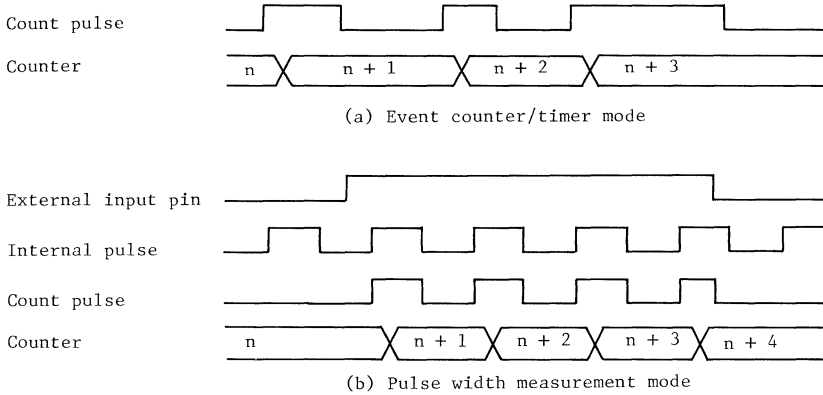| n | n + 1 | n + 2 | n + 3 | n + 4 |

(b) Pulse width measurement mode

Fig. 1.10.3 Mode and Count Value of Timer/Counter

Decrease in execution speed of instruction due to count operation

The CPU carries out the count operation requiring one instruction cycle time for the count request. Therefore, this causes the decrease in the apparent speed of instruction execution. Some examples are shown below :

(a) In the timer mode with count pulse rate of $fc/2^{10}$ Hz :

The count operation is inserted once every 128-instruction cycle time, so that the apparent speed is decreased by $1/127 \fallingdotseq 0.8\%$ instruction execution speed. For example, the apparent speed is 2.016μs to 2μs instruction execution speed.

(b) In the event count mode :

It depends on the count pulse rate applied to the external input pin. In the worst case, when the timer/counter 1 and 2 are operated at the same time with the maximum count pulse rate, the count operation is inserted once every 4-instruction cycle time for the timer/counter 1, and once every 5-instruction cycle time for the timer/counter 2.

The apparent speed of the instruction execution, therefore, de-
creases by $9/11 \doteqdot 82\%$. The apparent speed is $3.64\mu s$ to $2\mu s$ instruc-
tion execution speed.

(3) Interrupt by overflow (IOVF1, IOVF2)

At the time when the overflow occurs, the timer/counter gene-
rates the interrupt request.
That is, the interrupt request is generated when the count value of
FFF is changed to 000. The counting is continued after the interrupt
request signal is generated. Assuming that the CPU provides the
interrupt enabling state, and that the interrupt is accepted as soon
as the overflow interrupt has been generated, the interrupt process-
ing can be performed in the sequence illustrated in Fig. 1.10.4.

Instruction
cycle

Count pulse

Count request

Detection of overflow

Occurrence of IOVF

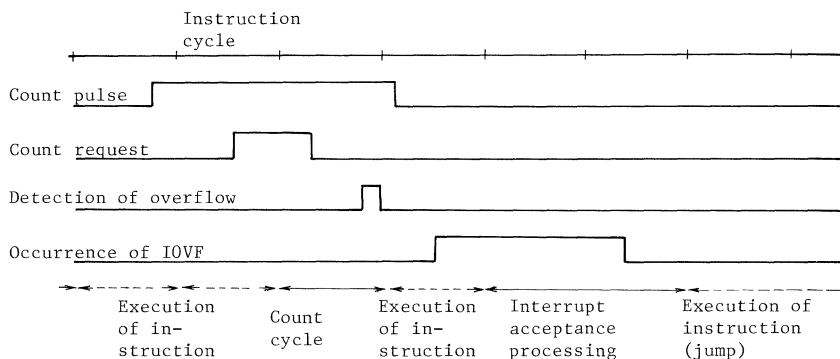| Execution of in- struction | Count cycle | Execution of in- struction | Interrupt acceptance processing | Execution of instruction (jump) |

Fig. 1.10.4   Timing Chart of Timer/Counter in
Interrupt by Overflow

1.11 Serial Port (SIO)

A 4-bit serial port with a buffer is provided to transfer the serial data from/to the external circuitry. It is connected to the external circuitry through three pins [R92 ($\overline{SCK}$), R91 (SO), R90 (SI)]. Since these pins are also used as port R9, the output latch of the R9 port should be set to "1" when the serial port is used. When it is not used, the pins can be used as I/O port R9.

Pin R90 in the transmit mode and pin R91 in the receive mode are also available as I/O port pin.

(1) Circuit configuration

The serial port consists of a 4-bit shift register, a 4-bit buffer register, and its control circuit.
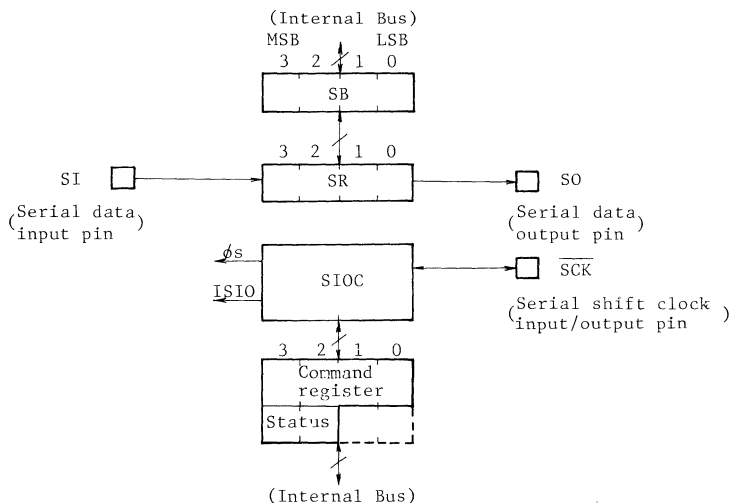
SR : 4-bit shift register   SIOC : Serial port control circuit
SB : 4-bit buffer register ϕs   : Internal shift clock
                              ISIO : Interrupt request

Fig. 1.11.1   Circuit Configuration of Serial Port

(2) Serial port control

The serial port operation is controlled by the command. The command register is accessed with port address OP1F, and reset to "0" at the initialization operation. The operation status can be informed through the status input, which is accessed with port address IPOE.

```
                      MSB          LSB
Serial port com-       3    2    1    0
mand register       ┌─────┬────┬────┬────┐
(Port address       │ESIO │ RM │ LM │ECKM│
 OP1F    )          └─────┴────┴────┴────┘
```

1 : Serial transfer enable or start request.

0 : Serial transfer disable or end request.

1 : Receive mode

0 : Transmit mode

1 : Shift operation at the leading edge of the shift clock

0 : Shift operation at the trailing edge of the shift clock

1 : External input is served as shift clock.

0 : Internal clock is served as shift clock.

```
                      MSB          LSB
Serial port            3    2    1    0
status input        ┌─────┬────┬────┬────┐
(Port address       │SIOF │SEF │ *  │ *  │     (* : don't care)
 IPOE    )          └─────┴────┴────┴────┘
```

1 : Operating condition of serial transfer

0 : Stopping or disabling condition of serial transfer

: Shifting operation can be monitored, and is used at time of transfer end request.
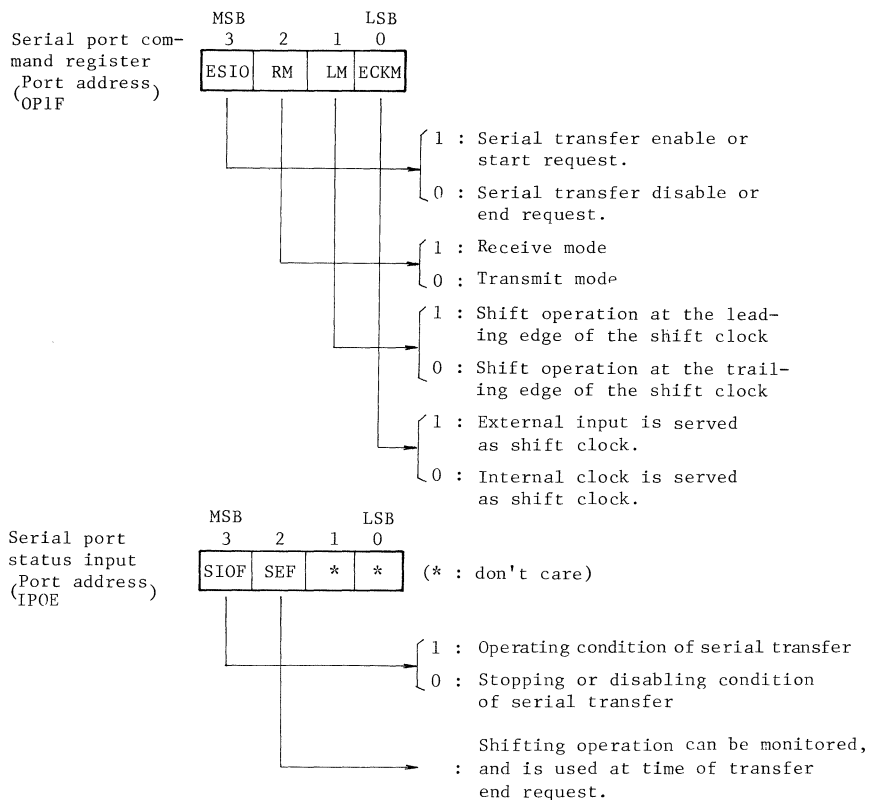
Fig. 1.11.2   Command Register, Status Input

(3) Shift clock (SCK)

The following shift clock modes can be selected by the contents of the command register.

(a) Clock source (External/internal mode)
(b) Shift edge of clock (Leading edge/trailing edge mode)

Internal clock mode

$fc/2^7$ Hz is used for the shift clock (when the basic clock frequency fc is 4.194304 MHz, the shift clock frequency is 32.768 kHz.). At this time, the clock is supplied to the external devices through the $\overline{SCK}$ pin. If the data setting (transmit mode) or the data reading (receive mode) rate by the program cannot follow the clock rate, the shift clock is automatically stopped and the next shift operation is suspended until the data processing is completed ("Wait"operation).

External clock mode

The shift operation is performed by the clock provided from the external circuitry since the $\overline{SCK}$ pin serves as an input.

Leading edge shift mode

Data is transmitted (transmit mode) or received (receive mode) at the leading edge of the $\overline{SCK}$ pin signal.

Trailing edge shift mode

Data is received (receive mode) at the trailing edge of the $\overline{SCK}$ pin signal.

The $\overline{SCK}$ pin must be set to the "high" level when the serial transfer is started. In the internal clock mode, the $\overline{SCK}$ pin is automatically set to the "high" level because it serves as an output.

(4) Operation mode

Selection of the following three transfer modes is available by changing the combination of the RM bit and LM bit of the command register.

RM (Bit 2)	LM (Bit 1)	ECKM (Bit 0)	Operation Mode
0	0	1/0	Can not be used
0	1	1/0	Transmit mode (Note) (External/Internal clock)
1	0	1/0	Receive(Trailing edge shift) mode (External/Internal clock)
1	1	1/0	Receive(Leading edge shift) mode (External/Internal clock)

(Note) Leading edge shift operation is performed.

Table 1.11.1 Operation Mode of Serial Port

In the transmit mode, the 4-bit data written to the buffer register from the CPU is shifted out by the shift register, and is output in the SO pin from the data of the LSB in sequence. The buffer register is accessed as the port address OPOF.

In the receive mode, the data to be input to the SI pin is shifted toward the LSB by the shift register in sequence, and is set in the buffer register after the 4-bit data has been received.

The CPU reads the contents of the buffer register, which is accessed as the port address IPOF.

Transmit mode

After this mode is set in the command register, the first
transmit data (4-bit) is written in the buffer register (the data
cannot be written in the buffer register, if the transmit mode is not
set). Then the data can be transmitted by setting the ESIO (MSB
of command register) to "1". The content of the buffer register
is transferred to the shift register by the first shift clock, and
the data in the LSB (D_0) is output to the SO pin. The buffer regis-
ter then becomes empty, so that the interrupt (ISIO) requesting the
next data takes place (buffer empty). After that, the remaining data
(D_1 - D_3) is automatically shifted out by the shift register by one
data at a shift clock. The control by use of a program is not
necessary in this operation.

Data is written in the buffer register by outputting the next
transmit data (4-bit) to the port address OPOF in the interrupt
service program, and at the same time the interrupt request is
reset to "0".
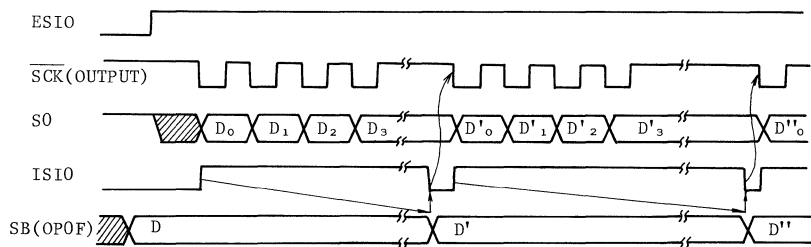
Internal clock operation

In case of $fc/2^7$ Hz internal clock operation, if the next data
is not set in the buffer register (OPOF has not been accessed by the
program) though the 4-bit data has been entirely shifted out, the
shift clock automatically stops, and the wait operation is taking
place until the data is set.

The maximum transmission rate is 31250 bit/sec. at the 4 MHz
basic clock.

External clock operation

Since the shift operation synchronizes entirely with the clock
provided from the external circuitry, the data should have been
written in the buffer register before the next 4-bit data is shifted
out. Therefore, the transfer rate is determined by the maximum
time lag from the receipt of interrupt request (ISIO) to the writing
of data in the buffer register by the interrupt service program.

ESIO

$\overline{SCK}$(OUTPUT)

SO

ISIO

SB(OPOF)

(a) Internal clock operation (with wait operation)

ESIO

$\overline{SCK}$ (INPUT)

SO

ISIO

SB(OPOF)
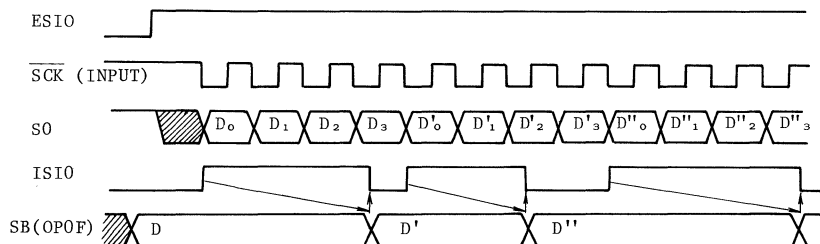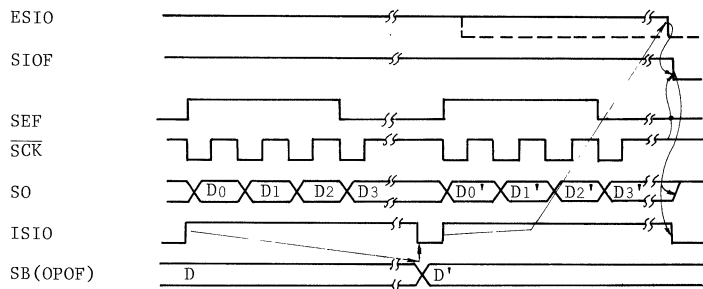
(b) External clock operation

Fig. 1.11.3   Transmit Mode

300

Completion of transmission

When the buffer register becomes empty, the interrupt occurs
to request the next data. In case where the transmission is desired
to be completed after the data is entirely transferred, the transmit
operation can be stopped upon completion of transferring the current
data shifted out, by resetting the ESIO to "0" without outputting
the data.   Whether or not the transfer operation is completed can
be sensed in a program by the SIOF (MSB of the status input).

In the external clock operation, the ESIO must be reset to "0"
before the next data is shifted out as in the data updating opera-
tion (however, the data is not updated when the operation is com-
pleted).   When the wait operation have been already performed in
the internal clock operation, the data transfer is terminated
immediately after ESIO = 0.

One word transfer can be terminated by ESIO = 0 in the inter-
rupt service program on receipt of the interrupt caused by the buffer
empty.

ESIO

SIOF

SEF

$\overline{SCK}$

SO $\quad D_0 \quad D_1 \quad D_2 \quad D_3 \qquad D_0' \quad D_1' \quad D_2' \quad D_3'$

ISIO

SB(OPOF) $\quad$ D $\qquad$ D'

(a) Internal clock operation (with wait operation)

ESIO

SIOF

SEF

$\overline{SCK}$

SO $\quad D_0 \quad D_1 \quad D_2 \quad D_3 \quad D_0' \quad D_1' \quad D_2' \quad D_3'$

ISIO

SB(OPOF) $\quad$ D $\qquad$ D'

(b) External clock operation

ESIO

SIOF

SEF

$\overline{SCK}$

SO $\quad D_0 \quad D_1 \quad D_2 \quad D_3$

ISIO

SB(OPOF) $\quad$ D
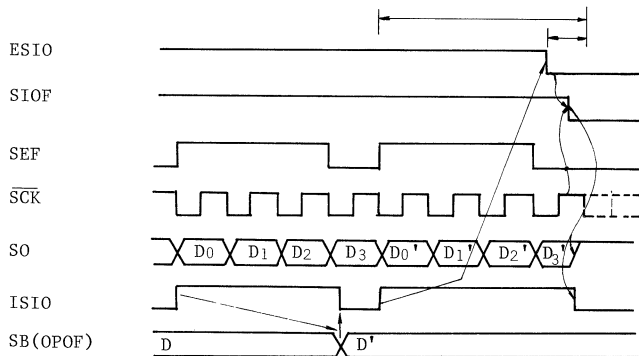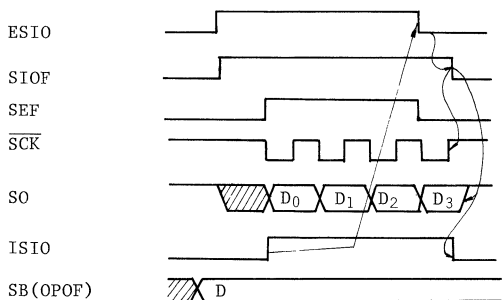
(c) Completion at one-word transfer

Fig. 1.11.4 Completion of Transmission

Toshiba
東芝

Receive (trailing edge shift) mode

Data can be received by setting the receive mode in the command register
as well as by setting the ESIO (MSB of command register) to "1".
When the four data are received from the SI pin, the 4-bit data is
transferred from the shift register to the buffer register. At the same time,
interrupt (ISIO) takes place to request the data reading (buffer full).
Since the shift register has been transferring the data to the buffer reg-
ister, the shift operation is continued without waiting for the data being
read.

When the data received from the port address IPOF is read in the inter-
rupt service program, the interrupt request is reset. And then the next 4-bit
data is transferred from the shift register to the buffer register if the
buffer register has been full.

Internal clock operation

During the operation of the internal clock of $fc/2^7$Hz, if the next 4-
bit data is not read out of the buffer register (the IPOF has not been ac-
cessed) in the program though the 4-bit data has been entirely input, the
shift clock automatically stops, and the wait operation is taking place
until the data is read out.

The maximum receiving rate is 31250 bit/sec at the 4 MHz basic clock.

External clock operation

Since the shift operation synchronizes entirely with the clock provided
from the external circuitry, the current data should have been read by the
instruction before the next 4-bit data is transferred to the buffer register.
The transfer rate is, therefore, determined by the maximum time lag from the
receipt of interrupt request (ISIO) to the read of the data in the buffer
register by the interrupt service program.

303

INTEGRATED CIRCUIT

Toshiba 東芝

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

ESIO

SCK (OUTPUT)

SI

Shift (Sampling)

ISIO

SB

Read (IPOF)

(a)   Internal clock operation (with wait operation)

ESIO

SCK (INPUT)

SI

Shift (Sampling)

ISIO

SB

Read (IPOF)

(b)   External clock operation
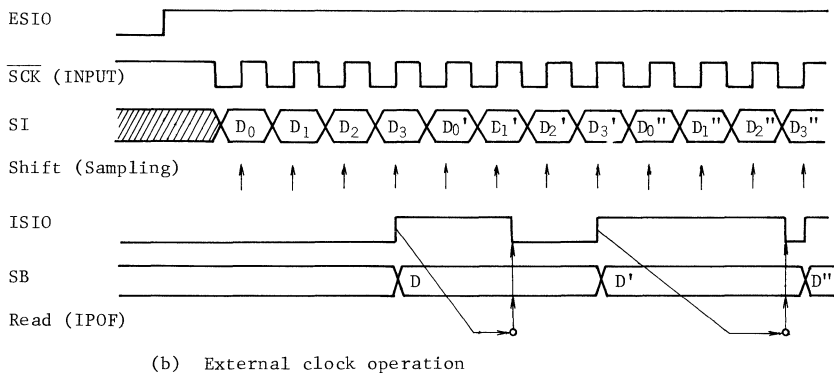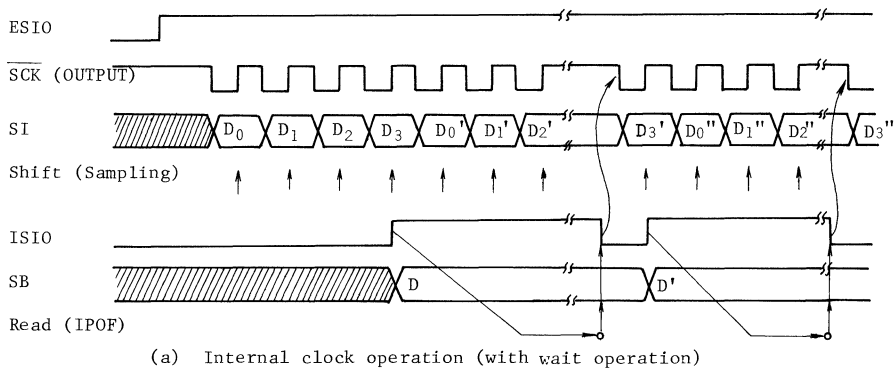
Fig. 1.11.5   Receive (trailing edge shift) Mode

304

INTEGRATED CIRCUIT

Toshiba

東芝

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

Completion of receiving

When all of the data are read, the receiving of data can be completed upon termination of the current data transfer, by resetting the ESIO to "0".

Whether or not the data transmission is terminated can be sensed in a program by the SIOF (MSB of status input).

To complete the receive operation when the synchronization is desired between the serial transfer and interrupt service program (indicates data reading or completion of receiving), there are two ways according to the speed of shift clock.

The receive/transmit mode must be maintained without switching the mode until the last data is read out even if the completion of the data transfer is indicated; otherwise the contents of the buffer register will be lost.

(a) Sufficiently slow data transfer rate (external clock operation)

If the timing, operated by the external clock, is slow enough to reset the ESIO to "0" prior to the generation of the next shift clock, the ESIO can be reset to "0" in the interrupt service program which is loaded to read out the last data. Thereafter the last data is read.

INTEGRATED CIRCUIT

Toshiba
東芝

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

ESIO

SIOF

SEF

$\overline{\text{SCK}}$

SI — D_2' D_3' D_0'' D_1'' D_2'' D_3''

ISIO

SB — D D' D''

Read (IPOF)
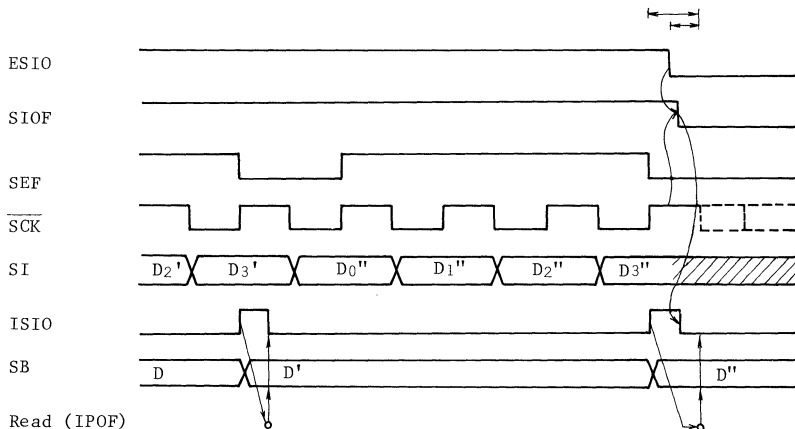
Fig. 1.11.6 Completion of Receiving (at slow transfer rate)

(b) Fast transfer rate

If the shift operation for the next data may start before the
current data is read out by receipt of the interrupt request becuase
the transfer rate is too fast, the interrupt service program which is
loaded to read out the last data but one should be used to reset the
ESIO to "0" after confirming that the SEF (bit 2 of status input) has
been set to "1".

Thereafter, the data should be read. No operation is required to
complete the data transfer in the interrupt service program for read-
ing the last data.

The method mentioned above is usually taken for the internal clock operation. In the external clock operation, however, the reset of the ESIO and the read of data must be completed before the last data is transferred to the buffer register.
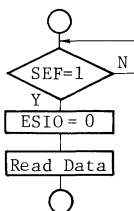
(a) Program sequence of receive end indication
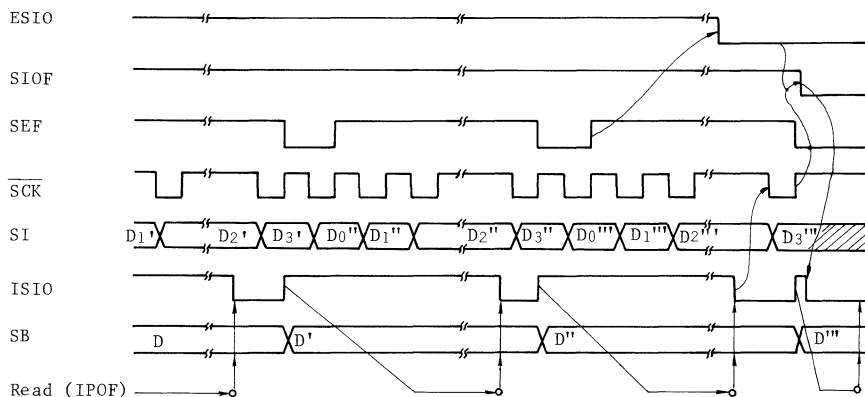
(b) Timing Chart
(in case of internal clock operation with wait operation)

Fig. 1.11.7 Completion of Receiving (at fast transfer rate)

INTEGRATED CIRCUIT

Toshiba
東芝   TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

(c) One word transfer

     The data receive operation starts after the ESIO is set to "1".
Then, the ESIO is reset to "0" after confirming that the SEF status
is set to "1". In this sequence, one interrupt casued by the buffer
full takes place; therefore, the data should be read out by the
service program.
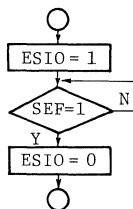
     (a)   Program sequence of receiving start/end indication
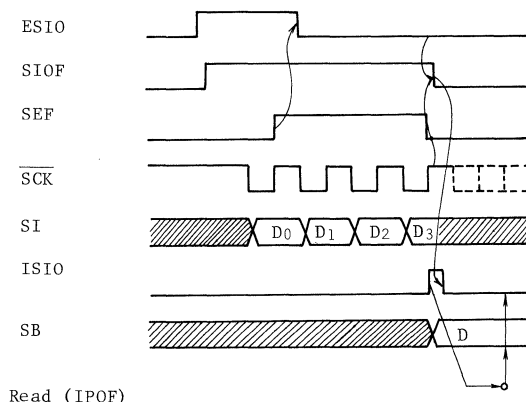
Read (IPOF)
                (b)   Timing Chart

     Fig. 1.11.8   Receiving Start/Completion (at one word transfer)

INTEGRATED CIRCUIT

東芝 Toshiba

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

Receive (leading edge shift) mode

With this mode set in the command register, the data can be received by setting the ESIO (MSB of command register) to "1".

When the four data are received from the SI pin, the 4-bit data is transferred from the shift register to the buffer register. At the same time, the interrupt (ISIO) occurs to request the data reading (buffer full). Since the shift register is transferring the data to the buffer register, the shift operation has been continued without waiting for the data being read.

When the data received from the port address IPOF is read in the interrupt service program, the interrupt request is reset. And then the next 4-bit data is transferred from the shift register to the buffer register if the buffer register has been full.

The basic operation in the receive (leading edge shift) mode is equivalent to that in the receive (trailing edge shift) mode except that the edge for the shift clock is different, and that at time of the transfer start, the first shifted data has been already input from the external circuitry before the first shift clock is applied to the data receipt. Timing charts are shown below.

ESIO

SIOF

SEF

$\overline{SCK}$(OUTPUT)

SI D_0 D_1 D_2 D_3 D_0' D_1' D_2' D_3'

Shift (Sampling)

ISIO

SB D D'

Read (IPOF)

(a) Internal clock operation (with wait operation)

ESIO

SIOF

SEF

$\overline{SCK}$(INPUT)

SI D_0 D_1 D_2 D_3 D_0' D_1' D_2' D_3' D_0'' D_1'' D_2'' D_3''

Shift (Sampling)

ISIO

SB D D' D''

Read (IPOF)
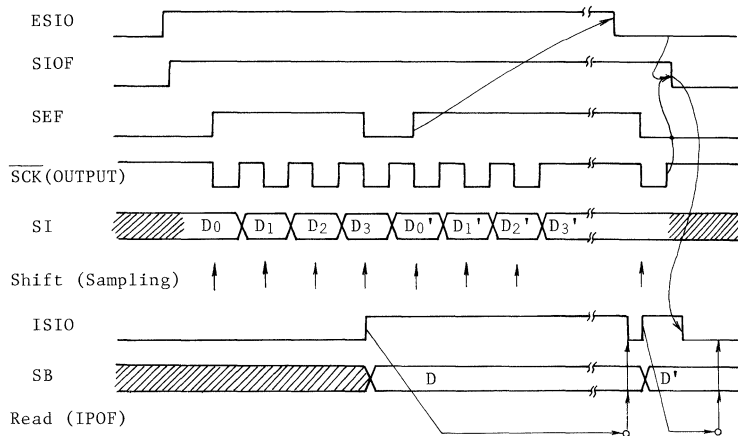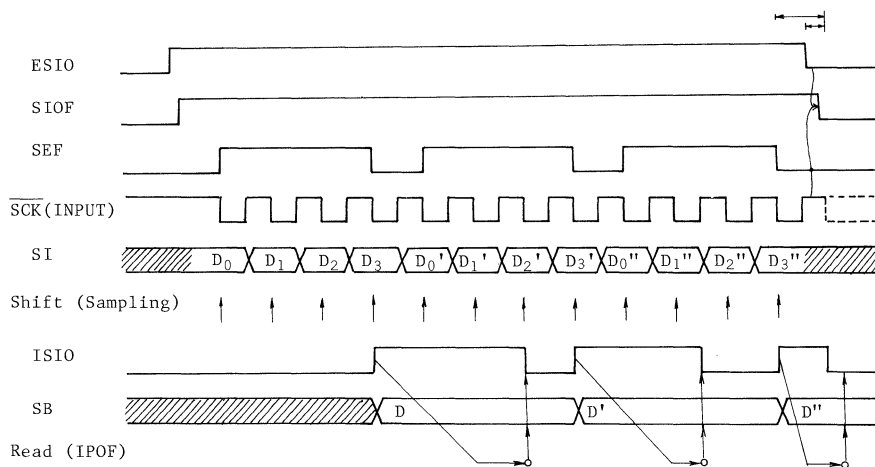
(b) External clock operation (at slow transfer rate)

ESIO

SIOF

SEF

$\overline{SCK}$(INPUT)

SI $\quad$ D_0 D_1 D_2 D_3 D_0' D_1' D_2' D_3' D_0'' D_1'' D_2'' D_3''

Shift (Sampling)

ISIO

SB $\quad$ D $\quad$ D' $\quad$ D''

Read (IPOF)
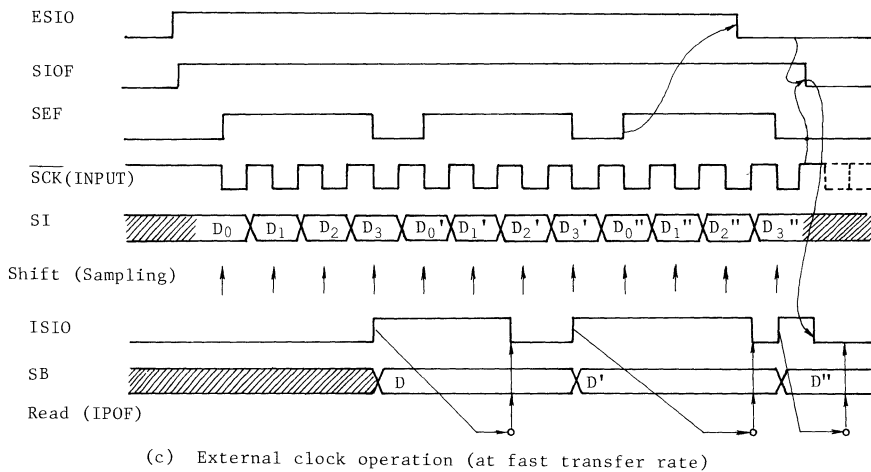
(c) External clock operation (at fast transfer rate)

ESIO

SIOF

SEF

$\overline{SCK}$

SI $\quad$ D_0 D_1 D_2 D_3

Shift (Sampling)

ISIO

SB $\quad$ D

Read (IPOF)
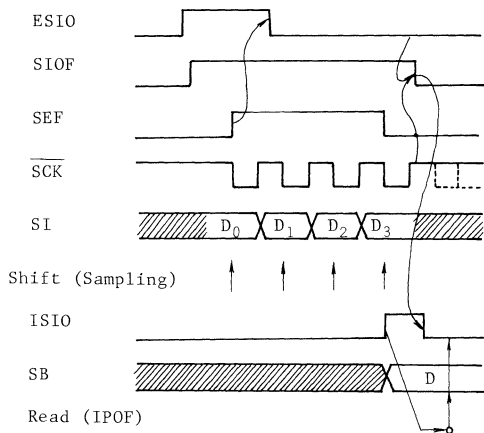
(d) One-word transfer

Fig. 1.11.9 Receive(Leading Edge Shift)Mode

311

2. Basic operation and pin operation

    1. Instruction cycle

    2. Basic clock (CP) generation

    3. Initialization operation

    4. Memory stand-by function

    5. Interrupt input

    6. Input/output port

    7. Other pins

    The timing in each basic operation, and the configuration, function, and timing of the pins are described according to the above items.

    The operation and timing with each component of the hardware are covered in detail in the description of each item of the components.

    Different input/output port circuit system can be specified according to the port. The details to specify the type of input/output port circuit are given in the descreption covering the program tape format.

2.1 Instruction cycle

    The instruction execution and the internal hardware control are synchronized with the basic clock (CP, fc Hz).

    The minimum unit of the instruction execution is called the "instruction cycle", and all instructions are executed by one or two instruction cycles, each of which is called one-cycle instruction or two-cycle instruction.

An instruction cycle consists of four machine cycles ($M_1 \sim M_4$), and each machine cycle requires two basic clock times.
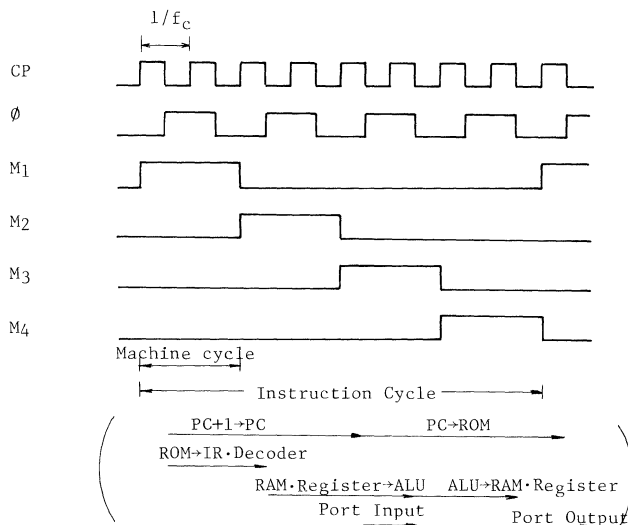
Fig. 2.1.1 Instruction Cycle

2.2 Basic clock (CP) generation

An oscillation circuit is contained, and the necessary clock is easily generated by connecting the resonator to external pins (X_{IN}, X_{OUT}). By the way, the oscillation circuit serves as schmitt circuit.

The clock generated in the oscillation circuit is called the "basic clock" with which the internal control is synchronized. The basic clock is applied to the timing generator and the control circuit of system to provide various control signals.

The following are the examples of the resonator connection.

X_{IN} X-tal X_{OUT}
Ceramic
C_{IN} C_{OUT}
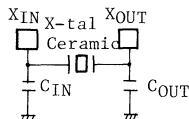
(a)   For X-tal or ceramic resonator

X_{IN} X_{OUT}
R
C

(b)   For RC

X_{IN} X_{OUT}

(Open)
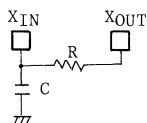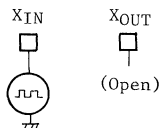
(c)   For external oscillator

Fig. 2.2.1   Resonator Connections

2.3   Initialization operation

Initialization operation is performed by keeping the $\overline{RESET}$ pin to the low level. However, the following conditions are required to put the initialization operation into practice with certainty

① The supply voltage is within the operating voltage.

② The oscillation circuit operates stably.

③ The $\overline{RESET}$ is held at the low level in at least three instruction cycle time.

Toshiba

INTEGRATED CIRCUIT

TECHNICAL DATA

東芝

TMP4740P
TMP4720P

PRELIMINARY

The following processings are performed by the initialization operation.

①   Reset the program counter to "0".

②   Set the status flag to "1".

③   Reset the interrupt enabling master F/F and the interrupt enabling register to "0", and also reset the interrupt latch to "0".

④   Reset the divider to "0".

⑤   Initialize the input/output port and command register to the fixed level.

The initialization operation is released due to the rise of the $\overline{\text{RESET}}$ pin to the high level, and the program can be executed from address 0 in sequence.

The $\overline{\text{RESET}}$ pin serves as Schmitt circuit input, and is connected with pull-up resister ($\approx 200\text{k}\Omega$ TYP., MOS-load resister).

2.4   Memory Stand-by function

Even during the cut off of the main power supply (V_{DD}), the RAM data can be held with low power dissipation by connecting the back-up power supply to the V_{HH} pin. This memory stand-by operation is performed by the following procedure.

①   Keep the $\overline{\text{RESET}}$ pin at the low level in at least three instruction cycle time before the V_{DD} power goes to the minimum operating voltage.

②   Hold the low level of the $\overline{\text{RESET}}$ pin. At the same time, the level of the V_{HH} voltage should be kept at that of more than mimimum stand-by voltage.

The operation should be started from initialization operation after the main power supply is resumed.

The power dissipation at the stand-by time can be minimized by this function.

2.5 Interrupt input

Two pins ($\overline{INT_1}$, $\overline{INT_2}$) are provided for the external interrupt input. Since these pins are common pins with R_8 port, they can be used as I/O pins (R_{82}, R_{80}) respectively, if not used as the interrupt input pins.

The interrupt via INT_2 can be inhibited at any time by the program, but the interrupt via INT_1 is not inhibited by it independently. Therefore, when this pin is used for the R_{82} port, the interrupt will always take place due to the detection of the falling edge of the signal. It is necessary to set a dummy interrrupt service program including the (RETI) instruction only, even if the INT_1 is not used.

The interrupt latch is set by the falling edge of the external inputs (INT_1, INT_2), and an interrupt request is made to the CPU. To assure that the interrupt latch is positively set or reset, and that the next interrupt request is set, both of the high and low levels should be kept for more than two instruction cycle time.

The external interrupt input is the Schmitt circuit input.
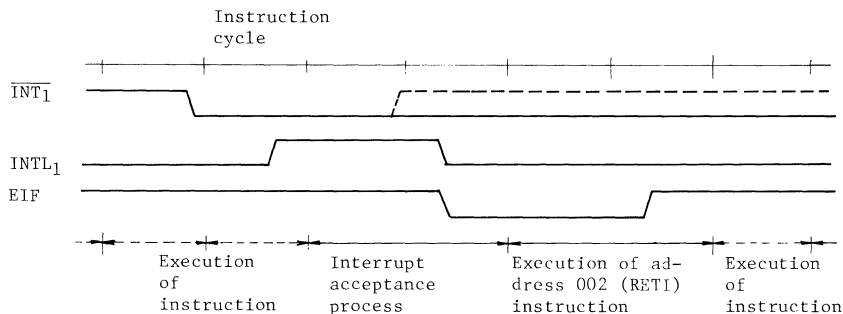
Fig. 2.5.1 Interrupt Timing (Dummy process of INT_1 interrupt)

2.6 Input/output port

(1) Input/output timing

The timing to read the external data from the input port or I/O port is in M_3 machine cycle in the second cycle of the input instruction (two-cycle instruction). Since this timing cannot be externally recognized, the transient input data should be processed by a program.

The timing to output the data to the output port or I/O port is in M_4 machine cycle in the second cycle of the output instruction (two-cycle instruction), but this timing cannot be externally recognized.
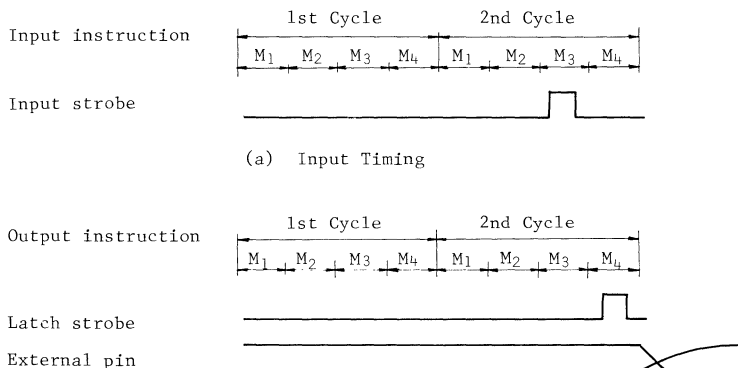
(a) Input Timing
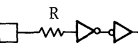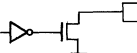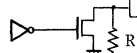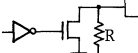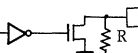
(b) Output Timing

Fig. 2.6.1 Input/Output Timing

(2)   Input/Output circuit format

The input/output circuit format of the input/output port is
shown following.

For the TMP4740P and the TMP4720P, any of the input/output circuit
systems shown in the following tables can be selected.   You can
specify your input/output circuit system when requesting the program
tape.

"$\overline{IOCODE}$ AA" is employed if not specified.

Input/Output Circuit Code ($\overline{IOCODE}$) AA					
Port Circuit	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent Circuit	R = 1kΩ (TYP.)		R = 1kΩ (TYP.)	R = 1kΩ (TYP.)	R = 1kΩ (TYP.)
Remark	○ High threshold input. ○ No resistor is contained.	○ Sink open drain output. ○ High output current. ○ Output latch is initialized to the high level.	○ Sink open drain output. ○ Output latch is initialized to the high level.	○ Sink open drain output. ○ Output latch is initialized to the high level.	○ Schmitt circuit input. ○ Sink open drain output. ○ Output latch is initialized to the high level.

Note :   The input/output port of the evaluator chip TMP4700C is made up with
the circuit system equivalent to this input/output circuit system;
therefore, the system of the TMP4700C can become equivalent to that
of the TMP4740P or the TMP4720P by externally installing EPROM
(program memory) on the TMP4700C (but TMP4700C is not contained the
pull-up resistor with $\overline{RESET}$ pin and the pull-down resistor with
TEST pin.).
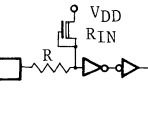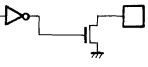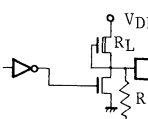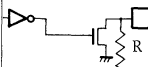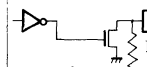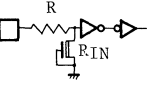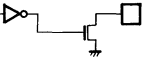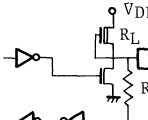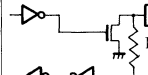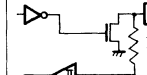
Input/Output Circuit Code   (IOCODE)   AE

Port Circuit	Input (K₀)	Output (P₁,P₂)	I/O (R₄,R₅,R₆)	I/O (R₇)	I/O (R₈,R₉)
I/O equiv- alent cir- cuit	 R_{IN}=100kΩ(TYP.) R = 1kΩ		 R_L = 5kΩ (TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o High threshold input o Pull-up resistor is contained.	o Sink open drain output. o High output current. o Output latch is initialized to the high level	o Sink open drain output. o Pull-up resistor is contained. o Output latch is initialized to the high level.	o Sink open drain output. o Output latch is initialized to the high level	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level

Input/Output Circuit Code   (IOCODE)   AF

Port Circuit	Input (K₀)	Output (P₁,P₂)	I/O (R₄,R₅,R₆)	I/O (R₇)	I/O (R₈,R₉)
I/O equiv- alent cir- cuit	 R_{IN}=100Ω (TYP.) R = 1kΩ (TYP.)		 R_L = 5kΩ (TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o High threshold input o Pull-down resistor is contained.	o Sink open drain output. o High output current. o Output latch is initialized to the high level	o Sink open drain output. o Pull-up resistor contained. o Output latch is initialized to the high level	o Sink open drain output. o Output latch is initialized to the high level	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level

319
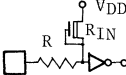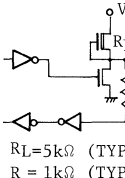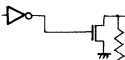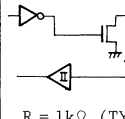
Input/Output Circuit Code (IOCODE) AH					
Port Circuit	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	$R_{IN}=100k\Omega$ (TYP.) $R = 1k\Omega$ (TYP.)		$R_L=5k\Omega$ (TYP.) $R = 1k\Omega$ (TYP.)	$R = 1k\Omega$ (TYP.)	$R = 1k\Omega$ (TYP.)
Remark	o High threshold input o Pull-up resistor is contained.	o Sink open drain output. o High output current. o Output latch is initialized to the high level	o Sink open drain output. o Pull-up resistor is contained. o Output latch is initialized to the low level	o Sink open drain output. o Output latch is initialized to the high level	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level

Input/Output Circuit Code (IOCODE) AI					
Port Circuit	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	$R_{IN}=100k\Omega$ (TYP.) $R = 1k\Omega$ (TYP.)		$R_L=5k\Omega$ (TYP.) $R = 1k\Omega$ (TYP.)	$R = 1k\Omega$ (TYP.)	$R = 1k\Omega$ (TYP.)
Remark	o High threshold input o Pull-down resistor is contained.	o Sink open drain output. o High output current. o Output latch is initialized to the high level	o Sink open drain output. o Pull-up resistor is contained. o Output latch is initialized to the low level	o Sink open drain output. o Output latch is initialized to the high level	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level

INTEGRATED CIRCUIT

Toshiba
東芝   TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

2.7 Other pins

Timer/Counter input

Two pins (T_1, T_2) are provided for the external timer/counter inputs. Since these pins are common pins with R_8 port, they can be also used as I/O pins (R_{83}, R_{81}), respectively, if not used as the timer/counter inputs.

The count latch is set by the rising edge of the external input (T_1, T_2), and a count request is made to the CPU. To assure that the count latch is positively set or reset, both of the high and low levels should be kept for more tham two instruction cycle times.

The external timer/counter input is the Schmitt circuit input.

Serial port

This port is connected to the external circuitry via three pins ($\overline{SCK}$, SO, SI), which are also used for the R9 port. These pins can be used as the pins of the R9 port (R_{92}, R_{91}, R_{90}), if not used for the serial port.

To assure that the shift operation is positively performed in the external clock mode, both of the high and low levels should be kept for more than two instruction cycle times.

The $\overline{SCK}$ input in the external clock mode and the SI input in the receive mode are Schmitt circuit inputs.

TEST pin

This pin is used for the shipment test. To operate the user system with this pin, the input should be surely set to the low level. By the way, TEST pin is connected with pull-down resistor ($\approx 70k\Omega$ TYP., MOS-load resistor).

東芝 **INTEGRATED CIRCUIT**

Toshiba

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

3. Instructions

The TLCS-47 series microcomputer is provided with 90 instructions, which are software compatible within the series. The instructions of the TLCS-47 series is consist of 1-byte instructions or 2-byte instructions. To classify them in terms of the execution time, there are 1-cycle instructions and 2-cycle instructions.

1-byte, 1-cycle instructions are mainly used in this series, and are arranged so as to improve the program efficiency.

1-byte	1-cycle instruction	40
1-byte	2-cycle instruction	11
2-byte	2-cycle instruction	39
	Total	90

(a) Classification by byte/cycle

Move instruction (Note 1)	22
Compare instruction	6
Arithmetic instruction	16
Logical instruction	9
Bit manipulation instruction	24
Input/Output instruction (Note 2)	6
Branch, subroutine instruction	6
Other instruction	1
Total	90

(Note 1) : Including ROM data referring instructions
(Note 2) : Including PLA referring instruction.

(b) Classification by function
Table 3.0.1 Classification of Instructions.

322

3.1 Description of symbols

The following symbols are used for describing the instructions in the
following explanations.

Symbol	Description
AC	Accumulator
M[x]	Data memory (Address x)
HR	H register
LR	L register
P[p]	Port (Address p)
FLAG	Flag
CF	Carry flag
ZF	Zero flag
SF	Status flag
GF	General flag
PC	Program counter
STACK[(SPW)]	Stack (Stack level is indicated by the contents of stack pointer word.)
SPW	Stack pointer word
EIF	Enable interrupt master F/F
EIR	Enable interrupt register
INTLj	Interrupt latch (j = 5 - 0)
DC	Data counter
ROM[x]	Program memory (Address x)
(ROM_H, ROM_L)	(High-order 4 bits or low-order 4 bits are expressed by suffix H/L.)
$\leftarrow$	Transfer
$\leftrightarrow$	Exchange
+	Addition
−	Substraction
$\wedge$	Logical AND of the corresponding bits
$\vee$	Logical OR of the corresponding bits
$\forall$	Exclusive OR of the corresponding bits

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

Symbol	Description
$(\overline{CF})$	Inversion of carry flag contents
null	Processed result is transferred nowhere
(AC)	Contents of accumulator
(H.L)	Contents of 8 bits coupling H register with L register
M[(H.L)]	Contents of data memory for which the contents of 8 bits coupling H register with L register is used as address.
(AC)<b>	Contents of bit assigned by b of accumulator.
(LR)<3:2>	Contents of bit 3 to bit 2 of L register
(PC)<11:6>	Contents of bit 11 to bit 6 of program counter

3.2 Description of instructions   (*): Note 1 (**): Exec.cycle ($\overset{*}{**}$): Hexadecimal

Item Class	Assembler Mnemonic	Object Code			Function	Flag(*)			(**)
		Binary		($\overset{*}{**}$)	Functional Description	CF	ZF	SF	
Move Instruction	LD A, @HL	0 0 0 0 1 1 0 0		0 C	$(AC) \leftarrow M[(H \cdot L)]$	–	Z	1	1
					Loads the contents of the data memory specified by the H and L registers in the accumulator.				
	LD A, x	0 0 1 1 1 1 0 0		3 C	$(AC) \leftarrow M[x]$	–	Z	1	2
		$x_7x_6x_5x_4$ $x_3x_2x_1x_0$		$x_H x_L$	Loads the contents of the data memory specified by the x of the instruction field in the accumulator.				
	LD HL, x	0 0 1 0 1 0 0 0		2 8	$(LR) \leftarrow M[x'], (HR) \leftarrow M[x'+1]$ $x'=x_7x_6x_5x_4x_3x_200$	–	–	1	2
		$x_7x_6x_5x_4$ $x_3x_2x_1x_0$		$x_H x_L$	Loads the consecutive two-word contents of the data memory specified by the x' (modified x) of the instruction field in the H and L registers.				
	LD A, #k	0 1 0 0 $k_3k_2k_1k_0$		4 k	$(AC) \leftarrow k$	–	Z	1	1
					Loads the immediate data k of the instruction field in the accumulator. Serves as the clear instruction when k = 0.				

INTEGRATED CIRCUIT

Toshiba / 東芝

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

Items / Class	Assembler Mnemonic	Object Code Binary	(**)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Move Instruction	LD H, #k	1 1 0 0 k₃k₂k₁k₀	C k	$(HR) \leftarrow k$ Loads the immediate data k of the instruction field in the H register. Serves as the clear instruction when k = 0.	– – 1	1
	LD L, #k	1 1 1 0 k₃k₂k₁k₀	E k	$(LR) \leftarrow k$ Loads the immediate data k of the instruction field in the L register. Serves as the clear instruction when k = 0.	– – 1	1
	LDL A, @DC	0 0 1 1 0 0 1 1	3 3	$(AC) \leftarrow ROM_L[(DC)]$ Loads the lower-order 4 bits of the data read out of the data table of the program memory specified by the data counter, in the accumulator.	– Z 1	2
	LDH A,@DC+	0 0 1 1 0 0 1 0	3 2	$(AC) \leftarrow ROM_H[(DC)]$, $(DC) \leftarrow (DC)+1$ Loads the higher-order 4 bits of the data read out of the data table of the program memory specified by the data counter, in the accumulator, and then increments the contents of the data counter. [Note 2]	– Z 1	2
	ST A, @HL	0 0 0 0 1 1 1 1	0 F	$M[(H \cdot L)] \leftarrow AC$ Stores the contents of the accumulator in the data memory specified by the H and L registers.	– – 1	1
	ST A,@HL+	0 0 0 1 1 0 1 0	1 A	$M[(H \cdot L)] \leftarrow (AC)$, $(LR) \leftarrow (LR)+1$ Stores the contents of the accumulator in the data memory specified by the H and L registers, and then increments the contents of the L register. [Note 3]	– Z $\overline{C}$	1
	ST A,@HL-	0 0 0 1 1 0 1 1	1 B	$M[(H \cdot L)] \leftarrow (AC)$, $(LR) \leftarrow (LR)-1$ Stores the contents of the accumulator in the data memory specified by the H and L registers, and then decrements the contents of the L register. [Note 3]	– Z $\overline{B}$	1
	ST A, x	0 0 1 1 1 1 1 1 x₇x₆x₅x₄ x₃x₂x₁x₀	3 F xₕxₗ	$M[x] \leftarrow (AC)$ Stores the contents of the accumulator in the data memory specified by the x of the instruction field.	– – 1	2
	ST #k,@HL+	1 1 1 1 k₃k₂k₁k₀	F k	$M[(H \cdot L)] \leftarrow k$, $(LR) \leftarrow (LR)+1$ Stores the immediate data k of the instruction field in the data memory specified by the H and L registers, and then increments the contents of the L register. [Note 3]	– Z $\overline{C}$	1

325

Items / Class	Assembler Mnemonic	Object Code Binary	(***)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Move Instruction	ST #k, y	0010 1101 $k_3k_2k_1k_0$ $y_3y_2y_1y_0$	2 D k y	M[y]←k	– – 1	2
				Stores the immediate value k of the instruction field in the data memory specified by y (page 0) of the instruction field. Serves as the clear instruction when k = 0.		
	MOV H, A	0001 0000	1 0	(AC)←(HR)	– Z 1	1
				Loads the contents of the H register in the accumulator.		
	MOV L, A	0001 0001	1 1	(AC)←(LR)	– Z 1	1
				Loads the contents of the L register in the accumulator.		
	XCH A, H	0011 0000	3 0	(HR)⇄(AC)	– Z 1	2
				Exchanges the contents of the accumulator for those of the H register. [Note 2]		
	XCH A, L	0011 0001	3 1	(LR)⇄(AC)	– Z 1	2
				Exchanges the contents of the accumulator for those of the L register. [Note 2]		
	XCH A,EIR	0001 0011	1 3	(EIR)⇄(AC)	– – 1	1
				Exchanges the contents of the accumulator for those of the interrupt enable register.		
	XCH A,@HL	0000 1101	0 D	M[(H·L)]⇄(AC)	– Z 1	1
				Exchanges the contents of the accumulator for those of the data memory specified by the H and L registers. [Note 2]		
	XCH A, x	0011 1101 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	3 D x_Hx_L	M[x]⇄(AC)	– Z 1	2
				Exchanges the contents of the accumulator for those of the data memory specified by the x of the instruction field. [Note 2]		
	XCH HL, x	0010 1001 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	2 9 x_Hx_L	M[x']⇄(LR),M[x'+1]⇄(HR) x'=$x_7x_6x_5x_4x_3x_2$00	– – 1	2
				Exchanges the contents of the H and L registers for consecutive two-word contents of the data memory specified by the x' (modified x) of the instruction field.		

Items / Class	Assembler Mnemonic	Object Code Binary	(**)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Compare Instruction	CMPR A,@HL	0 0 0 1　0 1 1 0	1 6	null←M[(H·L)]−(AC) / Compares the contents of the data memory specified by the H and L registers with those of the accumulator.	B̄ Z Z̄	1
	CMPR A, x	0 0 1 1　1 1 1 0 / $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	3 E / x_Hx_L	null←M[x]−(AC) / Compares the contents of the data memory specified by the x of the instruction field with those of the accumulator.	B̄ Z Z̄	2
	CMPR A,#k	1 1 0 1　$k_3k_2k_1k_0$	D k	null←k−(AC) / Compares the immediate data k of the instruction field with the contents of the accumulator. Serves as the accumulator test instruction when k = 0.	B Z Z̄	1
	CMPR H,#k	0 0 1 1　1 0 0 0 / 1 1 0 1　$k_3k_2k_1k_0$	3 8 / D k	null←k−(HR) / Compares the immediate data k of the instruction field with the contents of the H register. Serves as the H register test instruction when k = 0.	− Z B̄	2
	CMPR L,#k	0 0 1 1　1 0 0 0 / 1 0 0 1　$k_3k_2k_1k_0$	3 8 / 9 k	null←k−(LR) / Compares the immediate data k of the instruction field with the contents of the L register. Serves as the L register test instruction when k = 0.	− Z B̄	2
	CMPR y,#k	0 0 1 0　1 1 1 0 / $k_3k_2k_1k_0$ $y_3y_2y_1y_0$	2 E / k y	null←k−M[y] / Compares the immediate data k of the instruction field with the contents of the data memory specified by the y (page 0) of the instruction field. Serves as the data memory test instruction when k = 0.	B̄ Z Z̄	2
Arithmetic Instruction	INC A	0 0 0 0　1 0 0 0	0 8	(AC)←(AC)+1 / Increments the contents of the accumulator.	− Z C̄	1
	INC L	0 0 0 1　1 0 0 0	1 8	(LR)←(LR)+1 / Increments the contents of the L register.	− Z C̄	1

327

Items Class	Assembler Mnemonic	Object Code		Function	Flag(*)	(**)
		Binary	(*)	Functional Description	CF ZF SF	
Arithmetic Instruction	INC @HL	0 0 0 0 1 0 1 0	0 A	M[(H·L)]←M[(H·L)]+1	– Z $\overline{C}$	1
				Increments the contents of the data memory specified by the H and L registers.		
	DEC A	0 0 0 0 1 0 0 1	0 9	(AC)←(AC)–1	– Z $\overline{B}$	1
				Decrements the contents of the accumulator.		
	DEC L	0 0 0 1 1 0 0 1	1 9	(LR)←(LR)–1	– Z $\overline{B}$	1
				Decrements the contents of the L register.		
	DEC @HL	0 0 0 0 1 0 1 1	0 B	M[(H·L)]←M[(H·L)]–1	– Z $\overline{B}$	1
				Decrements the contents of the data memory specified by the H and L registers.		
	ADDC A,@HL	0 0 0 1 0 1 0 1	1 5	(AC)←(AC)+M[(H·L)]+(CF)	C Z $\overline{C}$	1
				Adds the contents of the data memory specified by the H and L registers as well as those of the carry flag to those of the accumulator, and places the result in the accumulator.		
	ADD A,@HL	0 0 0 1 0 1 1 1	1 7	(AC)←(AC)+M[(H·L)]	– Z $\overline{C}$	1
				Adds the contents of the data memory specified by the H and L registers to those of the accumulator, and places the result in the accumulator.		
	ADD A, #k	0 0 1 1 1 0 0 0	3 8	(AC)←(AC)+k	– Z $\overline{C}$	2
		0 0 0 0 $k_3k_2k_1k_0$	0 k	Adds the immediate data k of the instruction field to the contents of the accumulator, and places the result in the accumulator. Serves as the correction instruction for decimal addition and subtraction when k = 6 or A.		
	ADD H, #k	0 0 1 1 1 0 0 0	3 8	(HR)←(HR)+k	– Z $\overline{C}$	2
		1 1 0 0 $k_3k_2k_1k_0$	C k	Adds the immediate data k of the instruction field to the contents of the H register, and places the result in the H register. Serves as the H register increment instruction or the decrement instruction when k = 1 or F, respectively.		

INTEGRATED CIRCUIT

Toshiba
東芝

TECHNICAL DATA

TMP4740P
TMP4720P

RELIMINARY

Items Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	(**)	Functional Description		
Arithmetic Instruction	ADD L,#k	0 0 1 1 1 0 0 0 1 0 0 0 $k_3k_2k_1k_0$	3 8 8 k	$(LR) \leftarrow (LR)+k$ Adds the immediate data k of the instruction field to the contents of the L register, and places the result in the L register.	– Z $\overline{C}$	2
	ADD @HL,#k	0 0 1 1 1 0 0 0 0 1 0 0 $k_3k_2k_1k_0$	3 8 4 k	$M[(H \cdot L)] \leftarrow M[(H \cdot L)]+k$ Adds the immediate data k of the instruction field to the contents of the data memory specified by the H and L register, and places the result in the data memory. Serves as the correction instruction for the decimal addition and subtraction when k = 6 or A.	– Z $\overline{C}$	2
	ADD y,#k	0 0 1 0 1 1 1 1 $k_3k_2k_1k_0$ $y_3y_2y_1y_0$	2 F k y	$M[y] \leftarrow M[y]+k$ Adds the immediate data k of the instruction field to contents of the data memory specified by the y (page 0) of the instruction field, and places the result in the data memory. Serves as the correction instruction for decimal addition and subtraction when k = 6 or A.	– Z $\overline{C}$	2
	SUBRC A,@HL	0 0 0 1 0 1 0 0	1 4	$(AC) \leftarrow M[(H \cdot L)]-(AC)-(\overline{CF})$ Subtracts the contents of the accumulator and the inverse contents of the carry flag from the contents of the data memory specified by the H and L registers, and places the result in the accumulator.	$\overline{B}$ Z $\overline{B}$	1
	SUBR A,#k	0 0 1 1 1 0 0 0 0 0 0 1 $k_3k_2k_1k_0$	3 8 1 k	$(AC) \leftarrow k-(AC)$ Subtracts the contents of the accumulator from the immediate data k of the instruction field, and places the result in the accumulator. Serves as the accumulator 2's complement instruction or the data inversion (1's complement) instruction when k = 0 or F, respectively.	– Z $\overline{B}$	2

329
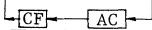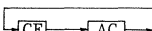
Items / Class	Assembler Mnemonic	Object Code Binary	Object Code ($\overset{*}{**}$)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Arithmetic Instruction	SUBR@HL,#k	0 0 1 1 1 0 0 0 0 1 0 1 $k_3k_2k_1k_0$	3 8 5 k	$M[(H \cdot L)] \leftarrow k - M[(H \cdot L)]$ Subtracts the contents of the data memory specified by the H and L registers from the immediate data k of the instruction field, and places the result in the data memory. Serves as the data memory 2's complement instruction or the data inversion (1's complement) instruction when k = 0 or F, respectively.	– Z $\overline{B}$	2
Logical Instruction	ROLC A	0 0 0 0 0 1 0 1	0 5	[CF]←[AC] (rotate left by 1 bit) Rotates the contents of the accumulator and carry flag to the left by one bit. [Note 4]	C Z $\overline{C}$	1
	RORC A	0 0 0 0 0 1 1 1	0 7	[CF]→[AC] (rotate right by 1 bit) Rotates the contents of the accumulator and carry flag to the right by one bit.[Note 4]	C Z $\overline{C}$	1
	AND A,@HL	0 0 0 1 1 1 1 0	1 E	$(AC) \leftarrow (AC) \wedge M[(H \cdot L)]$ Carries out the logical AND of the corresponding bits with the contents of the accumulator and those of the data memory specified by the H and L register, and places the result in the accumulator.	– Z $\overline{Z}$	1
	AND A,#k	0 0 1 1 1 0 0 0 0 0 1 1 $k_3k_2k_1k_0$	3 8 3 k	$(AC) \leftarrow (AC) \wedge k$ Carries out the logical AND of the corresponding bits with the contents of the accumulator and the immediate data k of the instruction field, and places the result in the accumulator.	– Z $\overline{Z}$	2
	AND @HL,#k	0 0 1 1 1 0 0 0 0 1 1 1 $k_3k_2k_1k_0$	3 8 7 k	$M[(H \cdot L)] \leftarrow M[(H \cdot L)] \wedge k$ Carries out the logical AND of the corresponding bits with the contents of the data memory specified by the H and L registers and the immediate data k of the instruction field, and places the result in the data memory.	– Z $\overline{Z}$	2

Items Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	(**)	Functional Description		
Logical Instruction	OR A, @HL	0 0 0 1 1 1 0 1	1 D	$(AC) \leftarrow (AC) \vee M[(H \cdot L)]$ Carries out the logical OR of the corresponding bits with the contents of the accumulator and those of the data memory specified by the H and L registers, and places the result in the accumulator.	– Z $\overline{Z}$	1
	OR A, #k	0 0 1 1 1 0 0 0 0 0 1 0 $k_3k_2k_1k_0$	3 8 2 k	$(AC) \leftarrow (AC) \vee k$ Carries out the logical OR of the corresponding bits with the contents of the accumulator and the immediate data k of the istruction field, and places the result in the accumulator.	– Z $\overline{Z}$	2
	OR @HL,#k	0 0 1 1 1 0 0 0 0 1 1 0 $k_3k_2k_1k_0$	3 8 6 k	$M[(H \cdot L)] \leftarrow M[(H \cdot L)] \vee k$ Carries out the logical OR of the corresponding bits with the contents of the data memory specified by the H and L registers and the immediate data k of the instruction field, and places the result in the data memory.	– Z $\overline{Z}$	2
	XOR A, @HL	0 0 0 1 1 1 1 1	1 F	$(AC) \leftarrow (AC) \forall M[(H \cdot L)]$ Carries out the logical exclusive OR of the corresponding bits with the contents of the accumulator and those of data memory specified by the H and L registers. and places the result in the accumulator.	– Z $\overline{Z}$	1
Bit Manipulation Instruction	TEST CF	0 0 0 0 0 1 1 0	0 6	$(SF) \leftarrow (\overline{CF})$, $(CF) \leftarrow 0$ Places the inverse contents of the carry flag in the status flag, and then resets the carry flag to "0".	0 – *	1
	TEST A, b	0 1 0 1 1 1 b_1b_0	5 C+b	$(SF) \leftarrow (\overline{AC}) <b>$ Places the inverse contents of the bit, which is specified by the b of the instruction field, of the accumulator, in the status flag.	– – *	1
	TEST @HL,b	0 1 0 1 1 0 b_1b_0	5 8+b	$(SF) \leftarrow \overline{M[(H \cdot L)]} <b>$ Places the inverse contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, in the status flag.	– – *	1

INTEGRATED CIRCUIT

TECHNICAL DATA

Toshiba
東芝

TMP4740P
TMP4720P

PRELIMINARY

Items Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	(※※)	Functional Description		
Bit Manipulation Instruction	TEST y, b	0 0 1 1 1 0 0 1 1 0 b_1b_0 $y_3y_2y_1y_0$	3 9 8+b y	$(SF) \leftarrow \overline{M[y]<b>}$ Places the inverse contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, in the status flag.	– – *	2
	TEST %P,b	0 0 1 1 1 0 1 1 1 0 b_1b_0 $p_3p_2p_1p_0$	3 B 8+b P	$(SF) \leftarrow \overline{P[p]<b>}$ Places the inverse contents of the bit, which is specified by the b of the instruction field, of the port (port register in the output port, and pin input in the input and I/O port) specified by the p of the instruction field, in the status flag.	– – *	2
	TEST @L	0 0 1 1 0 1 1 1	3 7	$(SF) \leftarrow \overline{P[(LR)<3:2>+4]<(LR)<1:0>>}$ Places the inverse contents of the bit, which is specified by the lower-order two bits of the L register, of the ports $R_4 - R_7$ (pin input) specified by the higher two bits of the L register, in the status flag.	– – *	2
	TESTP CF	0 0 0 0 0 1 0 0	0 4	$(SF) \leftarrow (CF)$, $(CF) \leftarrow 1$ Places the contents of the carry flag in the status flag, and then sets the carry flag to "1".	1 – *	1
	TESTP ZF	0 0 0 0 1 1 1 0	0 E	$(SF) \leftarrow (ZF)$ Places the contents of the zero flag in the status flag.	– – *	1
	TESTP GF	0 0 0 0 0 0 0 1	0 1	$(SF) \leftarrow (GF)$ Places the contents of the general flag in the status flag.	– – *	1
	TESTP y,b	0 0 1 1 1 0 0 1 1 1 b_1b_0 $y_3y_2y_1y_0$	3 9 C+b y	$(SF) \leftarrow M[y]<b>$ Places the contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, in the status flag.	– – *	2

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

Toshiba
東芝

Items / Class	Assembler Mnemonic	Object Code Binary	($**$)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Bit Manipulation Instruction	TESTP %P,b	0 0 1 1 1 0 1 1 1 1 $b_1 b_0$ $p_3 p_2 p_1 p_0$	3 B C+b P	$(SF) \leftarrow P[p]<b>$ Places the contents of the bit, which is specified by the b of the instruction field, of the port (port register for the output port, and pin input for the input or I/O ports), which is specified by the p of the instruction field, in the status flag.	− − *	2
	SET GF	0 0 0 0 0 0 1 1	0 3	$(GF) \leftarrow 1$ Sets the general flag to "1".	− − 1	1
	SET @HL, b	0 1 0 1 0 0 $b_1 b_0$	5 b	$M[(H \cdot L)]<b> \leftarrow 1$ Sets the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, to "1".	− − 1	1
	SET y, b	0 0 1 1 1 0 0 1 0 0 $b_1 b_0$ $y_3 y_2 y_1 y_0$	3 9 b y	$M[y]<b> \leftarrow 1$ Sets the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, to "1".	− − 1	2
	SET %p, b	0 0 1 1 1 0 1 1 0 0 $b_1 b_0$ $p_3 p_2 p_1 p_0$	3 B b y	$P[p]<b> \leftarrow 1$ Sets the bit, which is specified by the b of the instruction field, of the port specified by the p of the instruction field, to "1".	− − 1	2
	SET @L	0 0 1 1 0 1 0 0	3 4	$P[(LR)<3:2>+4]<(LR)<1:0>> \leftarrow 1$ Sets the bit, which is specified by the lower-order two bits of the L register, of the ports $R_4 - R_7$ specified by the higher-order two bits of the L register, to "1".	− − 1	2
	CLR GF	0 0 0 0 0 0 1 0	0 2	$(GF) \leftarrow 0$ Clears the general flag to "0".	− − 1	1
	CLR @HL, b	0 1 0 1 0 1 $b_1 b_0$	5 4+b	$M[(H \cdot L)]<b> \leftarrow 0$ Clears the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L register, to "0".	− − 1	1

Items Class	Assembler Mnemonic	Object Code Binary	(**)	Function Functional Description	Flag(*) CF ZF SF	(**)
Bit Manipulation Instruction	CLR y, b	0 0 1 1 1 0 0 1 0 1 $b_1 b_0$ $y_3 y_2 y_1 y_0$	3 9 4+b y	$M[y]<b> \leftarrow 0$ Clears the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, to "0".	− − 1	2
	CLR %P, b	0 0 1 1 1 0 1 1 0 1 $b_1 b_0$ $p_3 p_2 p_1 p_0$	3 B 4+b p	$P[p]<b> \leftarrow 0$ Clears the bit, which is specified by the b of the instruction field, of the port specified by the p of the instruction field, to "0".	− − 1	2
	CLR @L	0 0 1 1 0 1 0 1	3. 5	$P[(LR)<3:2>+4]<(LR)<1:0>> \leftarrow 0$ Clears the bit, which is specified by the lower-order two bits of the L register, of the ports $R_4 - R_7$ specified by the higher-order two bits of the L register, to "0".	− − 1	2
	CLR IL, r	0 0 1 1 0 1 1 0 1 1 $r_5 r_4$ $r_3 r_2 r_1 r_0$	3 6 C+$r_H r_L$	$(INTL)<5:0> \leftarrow (INTL)<5:0> \wedge r<5:0>$ Resets the interrupt latch INTLj when the rj of the instruction field is "0". (j = 5 − 0)	− − 1	2
	EICLR IL,r	0 0 1 1 0 1 1 0 0 1 $r_5 r_4$ $r_3 r_2 r_1 r_0$	3 6 4+$r_H r_L$	$(EIF) \leftarrow 1,$ $(INTL)<5:0> \leftarrow (INTL)<5:0> \wedge r<5:0>$ Sets the interrupt enable master F/F to "1". Interrupt latch INTLj is reset when the rj of the instruction field is "0". (j = 5 − 0)	− − 1	2
	DICLR IL,r	0 0 1 1 0 1 1 0 1 0 $r_5 r_4$ $r_3 r_2 r_1 r_0$	3 6 8+$r_H r_L$	$(EIF) \leftarrow 0,$ $(INTL)<5:0> \leftarrow (INTL)<5:0> \wedge r<5:0>$ Resets the interrupt enable master F/F to "0". Interrupt latch INTLj is reset when the rj of the instruction field is "0". (j = 5 − 0)	− − 1	2
Input Instruction	IN %P, A	0 0 1 1 1 0 1 0 0 0 1 0 $p_3 p_2 p_1 p_0$	3 A 2 P	$(AC) \leftarrow P[p]$ Places the input data from the port specified by the p of the instruction field in the accumulator.	− Z Z̄	2

Items \ Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	$\binom{*}{**}$	Functional Description		
Input/Output Instruction	IN %P, @HL	0 0 1 1 1 0 1 0 0 1 1 0 $p_3p_2p_1p_0$	3 A 6 P	$M[(H \cdot L)] \leftarrow P[p]$ Places the input data from the port specified by the p of the instruction field in the data memory specified by the H and L registers.	$- - \overline{Z}$	2
	OUT A, %P	0 0 1 1 1 0 1 0 1 0 $\overline{p_4}$ 0 $p_3p_2p_1p_0$	3 A $8+2\overline{p_4}$p	$P[p] \leftarrow (AC)$, $P = P_4P_3P_2P_1P_0$ Outputs the contents of the accumulator to the port specified by the p of the instruction field. $(0 \leq p \leq 31)$	$- - 1$	2
	OUT @HL,%P	0 0 1 1 1 0 1 0 1 1 $\overline{p_4}$ 0 $p_3p_2p_1p_0$	3 A $C+2\overline{p_4}$p	$P[p] \leftarrow M[(H \cdot L)]$, $P = P_4P_3P_2P_1P_0$ Outputs the contents of the data memory specified by the H and L registers to the port specified by the p of the instruction field. $(0 \leq p \leq 31)$	$- - 1$	2
	OUT #k,%P	0 0 1 0 1 1 0 0 $k_3k_2k_1k_0$ $p_3p_2p_1p_0$	2 C k P	$P[p] \leftarrow k$ Outputs the immediate data k of the instruction field to the port specified by the p of the instruction field. Serves as the clear instruction when k = 0.	$- - 1$	2
	OUTB @HL	0 0 0 1 0 0 1 0	1 2	$P[2] \cdot P[1] \leftarrow ROM[F \cdot (E+(CF)) \cdot M[(H \cdot L)]]$ Outputs the data (eight bits) of the program memory located in addresses FE0 - FFF, which use a five-bit data connecting the contents of the data memory specified by the H and L registers and those of the carry flag, as lower-order five-bit addresses, to the P_2 - P_1 ports.	$- - 1$	2
Branch· Subroutine Instruction	BS a	0 1 1 0 $a_{11}a_{10}a_9a_8$ $a_7a_6a_5a_4$ $a_3a_2a_1a_0$	6 a_H a_M a_L	If SF=1 then (PC)←a else null· Places the immediate data a of the instruction field in the program counter if the status flag is at "1". If the status flag is at "0", sets the status flag only to "1", and moves to the next address.	$- - 1$	2

INTEGRATED CIRCUIT

Toshiba 東芝

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

Items / Class	Assembler Mnemonic	Object Code Binary	Object Code ($**$)	Function / Functional Description	Flag(*) CF|ZF|SF	(**)
Branch·Subroutine Instruction	BSS a	1 0 $d_5 d_4$ $d_3 d_2 d_1 d_0$	$8+d_H d_L$	If SF=1 then (PC)←a else null, a=(PC)<11·6 >·d	– – 1	1
				Carries out the branch within a page (64-byte) if the status flag is at "1"; brings the immediate value d of the instruction field into the lower-order six bits of the program counter. Since the updated value remains in the higher-order six bits, if this instruction is specified in the last address in the page, branching is carried out to the next page. If the status flag is at "0", it sets the status flag only to "1", and moves to the next address. [Note 5]		
	CALL a	0 0 1 0 0 $a_{10} a_9 a_8$ $a_7 a_6 a_5 a_4$ $a_3 a_2 a_1 a_0$	2 aH a_M a_L	STACK[(SPW)]←(PC),(SPW)←(SPW)-1 (PC)←a, 0 ≤ a ≤ 2,047	– – –	2
				Carries out the subroutine call; saves the contents of the program counter in the stack, and decrements the stack pointer word, and then places the immediate data a of the instruction field in the program counter. However, the call address of the subroutine must be in the addresses 000 -7FF. [Note 5]		
	CALLS a	0 1 1 1 $n_3 n_2 n_1 n_0$	7 n	STACK[(SPW)]←(PC),(SPW)←(SPW)-1 (PC)←a, a=8n+6(n≠0), 134(n=0)	– – –	2
				Carries out the short form subroutine call. The operation is the same as that of the "CALL" instruction except that the value to be set in the program counter is automatically defined by the n of the instruction field. [Note 5]		
	RET	0 0 1 0 1 0 1 1	2 A	(SPW)←(SPW)+1,(PC)←STACK[(SPW)]	– – –	2
				Returns from the subroutine to the previous program; increments the stack pointer word, and restores the data of the return address from the stack to the program counter.		

Item / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binaty	(***)	Functional Description		
Branch·Subroutine Instruction	RETI	0 0 1 0 1 0 1 1	2 B	(SPW)←(SPW)+1, (FLAG·PC)←STACK[(SPW)], (EIF)←1	* * *	2
				Returns from the interrupt processing routine; increments the stack pointer word, and restores the data of the return address from the stack and the data of the flag, to the program counter and the flag, respectively. And then, it sets the interrupt enable master F/F to "1".		
Other Instruction	NOP	0 0 0 0 0 0 0 0	0 0	no operation	– – –	1
				Moves to the next instruction without performing any operation.		

Note 1. <u>Setting Condition of Flag.</u>

"C" indicates the carry output from the most significant position in the addition operation, and "B" indicates the borrow output from the most significant position in the subtraction operation.

"Z" indicates the zero detection signal to which "1" is applied only when either the ALU output of the processing result or all four bits of the data transferred to the accumulator are zero.

The flag is set to "C", "C̄", "B̄", "Z", "Z̄", "1", or "0" according to the data processing rusult. The value specified by the function is set to the flag with the mark "*", and the mark "−" denotes no change in the state of the flag.

Note 2. The zero flag is set according to the data set in the accumulator.

Note 3. The flags (ZF, SF) are set according to the result of increment or decrement of the L register.

Note 4. The carry is the data shifted out from the accumulator.

Note 5. The contents of the program counter indicate the next address of the instruction to be executed.

INTEGRATED CIRCUIT

Toshiba 東芝

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS ($V_{SS}=0V$)

SYMBOL	ITEM	RATING	UNITS
V_{DD}	Supply Voltage	$-0.5 \sim 7$	V
V_{HH}			
V_{IN}	Input Voltage	$-0.5 \sim 7$	V
V_{OUT1}	Output Voltage (Except Open Drain Port)	$-0.5 \sim 7$	V
V_{OUT2}	Output Voltage (Open Drain Port)	$-0.5 \sim 10$	
I_{OUT}	Output Current (P_1, P_2)	30	mA
P_D	Power Dissipation ($T_{opr}=70°C$)	850	mW
T_{sol}	Soldering Temperature · Time	260 (10 sec)	
T_{stg}	Storage Temperature	$-55 \sim 125$	°C
T_{opr}	Operating Temperature	$-30 \sim 70$	

RECOMMENDED OPERATING CONDITIONS ($V_{SS}=0V$)

SYMBOL	ITEM	CONDITION	MIN.	MAX.	UNITS
T_{opr}	Operating Temperature		-30	70	°C
V_{DD}	Supply Voltage		4.5	5.5	V
V_{HH}					
V_{HH1}	Supply Voltage (Memory Stand-by)		3.5	5.5	
V_{IH1}	High Level Input Voltage ($R_4 \sim R_7$)		2.2	V_{DD}	V
V_{IH2}	High Level Input Voltage(Except $R_4 \sim R_7$)		3	V_{DD}	
V_{IL1}	Low Level Input Voltage (Except K_0)		0	0.8	
V_{IL2}	Low Level Input Voltage (K_0)		0	1.2	
f_C	Clock Frequency		0.4	4.2	MHz
t_{WCH}	High Level Clock Pulse Width (Note 1)	$V_{IN}=V_{IH}$	80	$-$	nS
t_{WCL}	Low Level Clock Pulse Width (Note 1)	$V_{IN}=V_{IL}$	80	$-$	

(Note 1) For external clock operation.

DC CHARACTERISTICS (V_{SS}=0V, V_{DD}=V_{HH}=5V±10%, T_{opr}=−30 ∿ 70°C)

SYMBOL	PARAMETER	CONDITION	MIN.	(Note 1) TYP.	MAX.	UNITS
V_{HS}	Hysteresis Voltage (Schmitt Circuit Input)		−	0.5	−	V
I_{IN1}	Input Current (Note 2) (K_0, $\overline{RESET}$, TEST)	V_{DD}=V_{HH}=5.5V, V_{IN}=5.5V	−	−	20	μA
I_{IN2}	Input Current (Open Drain R Port)	V_{DD}=5.5V, V_{IN}=5.5V	−	−	20	μA
I_{IL}	Low Level Input Current (R Port with Pull-up Resistor)	V_{DD}=5.5V, V_{IN}=0.4V	−	−	−2	mA
I_{LO}	Output Leak Current (Open Drain P, R Port)	V_{DD}=5.5V, V_{OUT}=5.5V	−	−	20	μA
V_{OH}	High Level Output Voltage (R Port with Pull-up Resistor)	V_{DD}=4.5V, I_{OH}=−200μA	2.4	−	−	V
V_{OL}	Low Level Output Voltage (Except X_{OUT})	V_{DD}=4.5V, I_{OL}=1.6mA	−	−	0.4	V
I_{OL}	Low Level Output Current (P_1, P_2)	V_{DD}=5V, V_{OL}=1V	−	20	−	mA
I_{DD}+I_{HH}	Supply Current	V_{DD}=V_{HH}=5.5V	−	50	100	mA
I_{HH1}	Supply Current(Memory Stand-by)	V_{DD}=V_{SS}, V_{HH}=3.5V	−	5	10	mA

(Note 1) Typical values are at T_{opr}=25°C, V_{DD}=V_{HH}=5V.

(Note 2) When an input resistor is built in the device, the input current through the resistor is eliminated.

AC CHARACTERISTICS (V_{SS}=0V, V_{DD}=V_{HH}=5V±10%, T_{opr}=−30 ∿ 70°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
t_{cy}	Instruction Cycle Time		1.9	−	20	μS
t_{SDH}	Shift data hold time	(Note 1)	0.5t_{cy}−300	−	−	nS

AC TIMING CHART

• Serial Port (Completion of transmission)
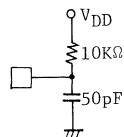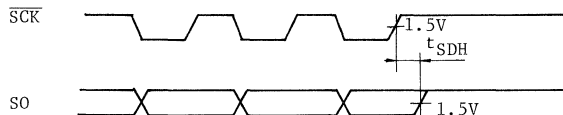
(Note 1) External circuit for serial ports $\overline{SCK}$ and SO

RELIMINARY

EXTERNAL DIMENSION VIEW

Unit in mm
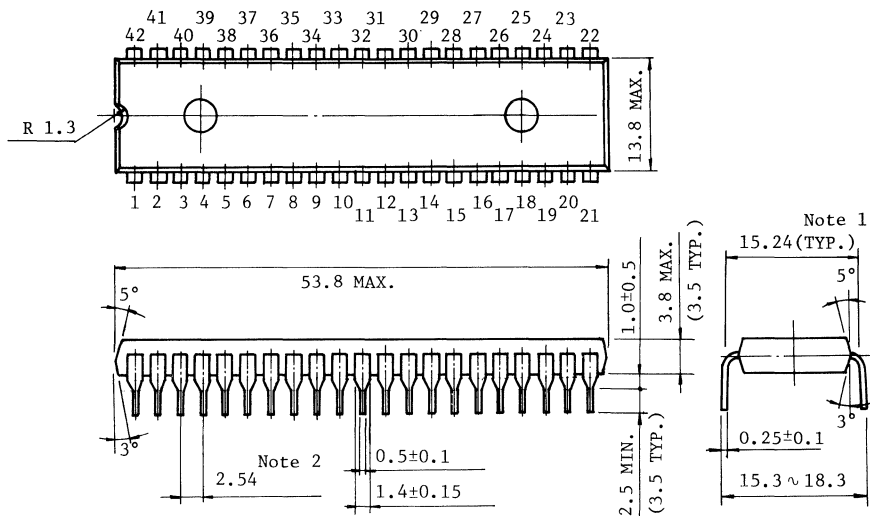
Weight 5.7g (TYP.)

Note 1. This dimension is measured at the center of bending point of leads.

Note 2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoretical positions with respect to No.1 and No.42 leads.

Specification of program tape and input/output circuit format

Engineering Samples(ES) of the TMP4740P and TMP4720P will be made if you specify the program data and input/output circuit format by use of a paper tape.

The paper tape format is equivalent to the Hex. format of Intel Co. (Format I).

The program data should be specified within the address space corresponding to the built-in ROM capacity; the addresses 000 - 7FF denote the address range in the TMP4720P. Accordingly, if the PLA data conversion table (addresses FE0 - FFF) is used, the table data must be assigned as the data located in addresses 7E0 - 7FF.

1. Specification of input/output circuit format

The paper tape of Format I starts recording the program data after record mark ":", but the input/output circuit code should be specified just before the first record mark.

The "IOCODE XX" format is used to define the input/output circuit code. XX denotes the proper input/output circuit code (two alphabets).

(Note)   If the input/output circuit code is not specified, "IOCODE AA" is employed. It should be noted that if the specified format is different from the standard one, and if the specified input/output circuit code is illegal, such specifications may be considered to have not been made.

(Example of tape list)

```
TOSHIBA MICROCOMPUTER TLCS-47
IOCODE AA
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67E31F5D8ABA6DF292F113F5C1
:100020004FF1F
                 :
                 :
                 :
:1007E000B53D42E0EC32546025B7308CDD52063D1D
:1007F000B4BE9E9E345B6138060B20BC372BF60BD6
:00000001FF
```

2. Program tape format (Format I)

.... Leaders   More than 50 characters of "NULL"

...... Comment

record mark ":" and the input/output circuit format are not included.

Specification of the input/output circuit format.

Not specially required.

... Specification of the input/output circuit format IOCODE XX (XX denotes the input/output circuit code) (It does not matter if there is "NULL" from immediately after ℂℝ, 🄻🄵 to immediately before record marking.)

It is considered to be no specification, if not specified.

...... Record mark (start of record)

.... Record length

the number of program data within one record is represented by two-digit hexadecimal notation. "00" denotes the end of data.

.... Load address

the first address of the record is represented by four-digit hexadecimal notation.

... Record type

"00" ..... Normal record
"01" ..... End of file record

.... Data

1-byte data is represented by two-digit hexadecimal notation.

... Checksum

two-digit hexadecimal notation is used to represent the data of the lower-order eight bits in the value which the data covering from the record length up to immediately before the checksum is subtracted from the initial value 0.

...... Record mark (the same procedure is repeated hereinafter.)

... Trailers   More than 50 characters of "NULL"

東芝 Toshiba

LIST OF INSTRUCTIONS

Classification	Mnemonic	Binary 1st Byte	Binary 2nd Byte	Hex 1st Byte	Hex 2nd Byte	Function	CF	ZF	SF	Exec. Cycle	Note
Move	LD A ,@HL	00 00 11 00		0 C		$(AC) \leftarrow M[(H \cdot L)]$	—	Z	1	1	
	LD A , x	00 11 11 00	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 C	$x_H x_L$	$(AC) \leftarrow M[x]$	—	Z	1	2	
	LD HL, x	00 10 10 00	$x_7x_6x_5x_4x_3x_2x_1x_0$	2 8	$x_H x_L$	$(LR) \leftarrow M[x'],(HR) \leftarrow M[x'+1], x'=x_7x_6x_5x_4x_3x_2|0$	—	—	1	2	
	LD A ,#k	01 00 $k_3k_2k_1k_0$		4 k		$(AC) \leftarrow k$	—	Z	1	1	
	LD H ,#k	11 00 $k_3k_2k_1k_0$		C k		$(HR) \leftarrow k$	—	—	1	1	
	LD L ,#k	11 10 $k_3k_2k_1k_0$		E k		$(LR) \leftarrow k$	—	—	1	1	
	LDL A ,@DC	00 11 00 11		3 3		$(AC) \leftarrow ROM_L[(DC)]$	—	Z	1	2	
	LDH A ,@DC+	00 11 00 10		3 2		$(AC) \leftarrow ROM_H[(DC)],(DC) \leftarrow (DC)+1$	—	Z	1	2	*2
	ST A ,@HL	00 00 11 11		0 F		$M[(H \cdot L)] \leftarrow (AC)$	—	—	1	1	
	ST A ,@HL+	00 01 10 10		1 A		$M[(H \cdot L)] \leftarrow (AC),(LR) \leftarrow (LR)+1$	—	Z	$\bar{C}$	1	*3
	ST A ,@HL-	00 01 10 11		1 B		$M[(H \cdot L)] \leftarrow (AC),(LR) \leftarrow (LR)-1$	—	Z	$\bar{B}$	1	*3
	ST A , x	00 11 11 11	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 F	$x_H x_L$	$M[x] \leftarrow (AC)$	—	—	1	2	
	ST #k ,@HL+	11 11 $k_3k_2k_1k_0$		F k		$M[(H \cdot L)] \leftarrow k,(LR) \leftarrow (LR)+1$	—	Z	$\bar{C}$	1	*3
	ST #k , y	00 10 01 11	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 D	k v	$M[y] \leftarrow k$	—	—	1	2	
	MOV H , A	00 01 00 00		1 0		$(AC) \leftarrow (HR)$	—	Z	1	1	
	MOV L , A	00 01 00 01		1 1		$(AC) \leftarrow (LR)$	—	Z	1	1	
	XCH A , H	00 11 00 00		3 0		$(HR) \rightleftarrows (AC)$	—	Z	1	2	*2
	XCH A , L	00 11 00 01		3 1		$(LR) \rightleftarrows (AC)$	—	Z	1	2	*2
	XCH A ,EIR	00 01 00 11		1 3		$(EIR) \rightleftarrows (AC)$	—	—	1	1	
	XCH A ,@HL	00 01 11 01		0 D		$M[(H \cdot L)] \rightleftarrows (AC)$	—	Z	1	1	*2
	XCH A , x	00 11 11 01	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 D	$x_H x_L$	$M[x] \rightleftarrows (AC)$	—	Z	1	2	*2
	XCH HL, x	00 10 10 01	$x_7x_6x_5x_4x_3x_2x_1x_0$	2 9	$x_H x_L$	$M[x'] \rightleftarrows (LR),M[x'+1] \rightleftarrows (HR), x'=x_7x_6x_5x_4x_3x_200$	—	—	1	2	
Compare	CMPR A ,@HL	00 01 10 00		1 6		$null \leftarrow M[(H \cdot L)]-(AC)$	$\bar{B}$	Z	$\bar{Z}$	1	
	CMPR A , x	00 11 11 10	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 E	$x_H x_L$	$null \leftarrow M[x]-(AC)$	$\bar{B}$	Z	$\bar{Z}$	2	
	CMPR A ,#k	11 01 $k_3k_2k_1k_0$		D k		$null \leftarrow k-(AC)$	$\bar{B}$	Z	$\bar{Z}$	1	
	CMPR H ,#k	00 11 10 00	11 01 $k_3k_2k_1k_0$	3 8	D k	$null \leftarrow k-(HR)$	—	Z	$\bar{B}$	2	
	CMPR L ,#k	00 11 10 00	10 01 $k_3k_2k_1k_0$	3 8	9 k	$null \leftarrow k-(LR)$	—	Z	$\bar{B}$	2	
	CMPR y ,#k	00 10 11 10	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 E	k y	$null \leftarrow k-M[y]$	$\bar{B}$	Z	$\bar{Z}$	2	
Arithmetic	INC A	00 00 10 00		0 8		$(AC) \leftarrow (AC)+1$	—	Z	$\bar{C}$	1	
	INC L	00 01 10 00		1 8		$(LR) \leftarrow (LR)+1$	—	Z	$\bar{C}$	1	
	INC @HL	00 00 10 10		0 A		$M[(H \cdot L)] \leftarrow M[(H \cdot L)]+1$	—	Z	$\bar{C}$	1	
	DEC A	00 00 10 01		0 9		$(AC) \leftarrow (AC)-1$	—	Z	$\bar{B}$	1	
	DEC L	00 01 10 01		1 9		$(LR) \leftarrow (LR)-1$	—	Z	$\bar{B}$	1	
	DEC @HL	00 00 10 11		0 B		$M[(H \cdot L)] \leftarrow M[(H \cdot L)]-1$	—	Z	$\bar{B}$	1	
	ADDC A ,@HL	00 01 01 01		1 5		$(AC) \leftarrow (AC)+M[(H \cdot L)]+(CF)$	C	Z	$\bar{C}$	1	
	ADD A ,@HL	00 01 01 11		1 7		$(AC) \leftarrow (AC)+M[(H \cdot L)]$	—	Z	$\bar{C}$	1	
	ADD A ,#k	00 11 10 00	00 00 $k_3k_2k_1k_0$	3 8	0 k	$(AC) \leftarrow (AC)+k$	—	Z	$\bar{C}$	2	
	ADD H ,#k	00 11 10 00	11 00 $k_3k_2k_1k_0$	3 8	C k	$(HR) \leftarrow (HR)+k$	—	Z	$\bar{C}$	2	
	ADD L ,#k	00 11 10 00	10 00 $k_3k_2k_1k_0$	3 8	8 k	$(LR) \leftarrow (LR)+k$	—	Z	$\bar{C}$	2	
	ADD @HL,#k	00 11 10 00	01 00 $k_3k_2k_1k_0$	3 8	4 k	$M[(H \cdot L)] \leftarrow M[(H \cdot L)]+k$	—	Z	$\bar{C}$	2	
	ADD y ,#k	00 10 01 11	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 F	k y	$M[y] \leftarrow M[y]+k$	—	Z	$\bar{C}$	2	
	SUBRC A ,@HL	00 01 01 00		1 4		$(AC) \leftarrow M[(H \cdot L)]-(AC)-\overline{(CF)}$	$\bar{B}$	Z	$\bar{B}$	1	
	SUBR A ,#k	00 11 10 00	00 01 $k_3k_2k_1k_0$	3 8	1 k	$(AC) \leftarrow k-(AC)$	—	Z	$\bar{B}$	2	
	SUBR @HL,#k	00 11 10 00	01 01 $k_3k_2k_1k_0$	3 8	5 k	$M[(H \cdot L)] \leftarrow k-M[(H \cdot L)]$	—	Z	$\bar{B}$	2	
Logical	ROLC A	00 00 01 01		0 5		$\boxed{CF} \leftarrow \boxed{AC}$ (rotate left by 1 bit)	C	Z	$\bar{C}$	1	*4
	RORC A	00 00 01 11		0 7		$\boxed{AC} \rightarrow \boxed{CF}$ (rotate right by 1 bit)	C	Z	$\bar{C}$	1	*4
	AND A ,@HL	00 01 11 10		1 E		$(AC) \leftarrow (AC) \wedge M[(H \cdot L)]$	—	Z	$\bar{Z}$	1	
	AND A ,#k	00 11 10 00	00 11 $k_3k_2k_1k_0$	3 8	3 k	$(AC) \leftarrow (AC) \wedge k$	—	Z	$\bar{Z}$	2	
	AND @HL,#k	00 11 10 00	01 11 $k_3k_2k_1k_0$	3 9	7 k	$M[(H \cdot L)] \leftarrow M[(H \cdot L)] \wedge k$	—	Z	$\bar{Z}$	2	
	OR A ,@HL	00 01 11 01		1 D		$(AC) \leftarrow (AC) \vee M[(H \cdot L)]$	—	Z	$\bar{Z}$	1	
	OR A ,#k	00 11 10 00	00 10 $k_3k_2k_1k_0$	3 8	2 k	$(AC) \leftarrow (AC) \vee k$	—	Z	$\bar{Z}$	2	
	OR @HL,#k	00 11 10 00	01 10 $k_3k_2k_1k_0$	3 8	6 k	$M[(H \cdot L)] \leftarrow M[(H \cdot L)] \vee k$	—	Z	$\bar{Z}$	2	
	XOR A ,@HL	00 01 11 11		1 F		$(AC) \leftarrow (AC) \veebar M[(H \cdot L)]$	—	Z	$\bar{Z}$	1	

343

(continued)

Classification	Assembler Mnemonic	Object Code Binary 1st Byte	2nd Byte	Hex 1st Byte	Hex 2nd Byte	Function	CF	ZF	SF	Execution Cycle	*1
Bit Manipulation	TEST CF	00 00 01 10		0	6	$(SF)\leftarrow(\overline{CF}),(CF)\leftarrow0$	0	—	*	1	
	TEST A,b	01 01 11 b_1b_0		5	C+b	$(SF)\leftarrow(\overline{AC})\langle b\rangle$	—	—	*	1	
	TEST @HL,b	01 01 10 b_1b_0		5	8+b	$(SF)\leftarrow\overline{M[(H\cdot L)]}\langle b\rangle$	—	—	*	1	
	TEST y,b	00 11 10 01 10	$b_1b_0y_3y_2y_1y_0$	3 9 8+b	y	$(SF)\leftarrow\overline{M[y]}\langle b\rangle$	—	—	*	2	
	TEST %p,b	00 11 10 11 10	$b_1b_0p_3p_2p_1p_0$	3 B 8+b	p	$(SF)\leftarrow\overline{P[p]}\langle b\rangle$	—	—	*	2	
	TEST @L	00 11 01 11		3	7	$(SF)\leftarrow\overline{P[(LR)\langle3:2\rangle+4]\langle(LR)\langle1:0\rangle\rangle}$	—	—	*	2	
	TESTP CF	00 00 01 00		0	4	$(SF)\leftarrow(CF),(CF)\leftarrow1$	1	—	*	1	
	TESTP ZF	00 00 11 10		0	E	$(SF)\leftarrow(ZF)$	—	—	*	1	
	TESTP GF	00 00 00 01		0	1	$(SF)\leftarrow(GF)$	—	—	*	1	
	TESTP y,b	00 11 10 01 11	$b_1b_0y_3y_2y_1y_0$	3 9 C+b	y	$(SF)\leftarrow M[y]\langle b\rangle$	—	—	*	2	
	TESTP %p,b	00 11 10 11 11	$b_1b_0p_3p_2p_1p_0$	3 B C+b	p	$(SF)\leftarrow P[p]\langle b\rangle$	—	—	*	2	
	SET GF	00 00 00 11		0	3	$(GF)\leftarrow1$	—	—	1	1	
	SET @HL,b	01 01 00 b_1b_0		5	b	$M[(H\cdot L)]\langle b\rangle\leftarrow1$	—	—	1	1	
	SET y,b	00 11 10 01 00	$b_1b_0y_3y_2y_1y_0$	3 9 b	y	$M[y]\langle b\rangle\leftarrow1$	—	—	1	2	
	SET %p,b	00 11 10 11 00	$b_1b_0p_3p_2p_1p_0$	3 B b	p	$P[p]\langle b\rangle\leftarrow1$	—	—	1	2	
	SET @L	00 11 01 00		3	4	$P[(LR)\langle3:2\rangle+4]\langle(LR)\langle1:0\rangle\rangle\leftarrow1$	—	—	1	2	
	CLR GF	00 00 00 10		0	2	$(GF)\leftarrow0$	—	—	1	1	
	CLR @HL,b	01 01 01 b_1b_0		5	4+b	$M[(H\cdot L)]\langle b\rangle\leftarrow0$	—	—	1	1	
	CLR y,b	00 11 10 01 01	$b_1b_0y_3y_2y_1y_0$	3 9 4+b	y	$M[y]\langle b\rangle\leftarrow0$	—	—	1	2	
	CLR %p,b	00 11 10 11 01	$b_1b_0p_3p_2p_1p_0$	3 B 4+b	p	$P[p]\langle b\rangle\leftarrow0$	—	—	1	2	
	CLR @L	00 11 01 01		3	5	$P[(LR)\langle3:2\rangle+4]\langle(LR)\langle1:0\rangle\rangle\leftarrow0$	—	—	1	2	
	CLR IL,r	00 11 01 01 11	$r_5r_4r_3r_2r_1r_0$	3 6 C+r_Hr_L		$(INTL)\langle5:0\rangle\leftarrow(INTL)\langle5:0\rangle\wedge r\langle5:0\rangle$	—	—	1	2	
	EICLR IL,r	00 11 01 01 01	$r_5r_4r_3r_2r_1r_0$	3 6 4+r_Hr_L		$(EIF)\leftarrow1,(INTL)\langle5:0\rangle\leftarrow(INTL)\langle5:0\rangle\wedge r\langle5:0\rangle$	—	—	1	2	
	DICLR IL,r	00 11 01 01 10	$r_5r_4r_3r_2r_1r_0$	3 6 8+r_Hr_L		$(EIF)\leftarrow0,(INTL)\langle5:0\rangle\leftarrow(INTL)\langle5:0\rangle\wedge r\langle5:0\rangle$	—	—	1	2	
Input/Output	IN %p,A	00 11 10 10 00 10	$p_3p_2p_1p_0$	3 A 2	p	$(AC)\leftarrow P[p]$	—	Z	$\overline{Z}$	2	
	IN %p,@HL	00 11 10 10 00 11	$p_3p_2p_1p_0$	3 A 6	p	$M[(H\cdot L)]\leftarrow P[p]$	—	—	$\overline{Z}$	2	
	OUT A,%p	00 11 10 10 10 $\overline{P_4}$0	$p_3p_2p_1p_0$	3 A 8+$2\overline{P_4}$	p	$P[p]\leftarrow(AC),\quad p=P_4P_3P_2P_1P_0$	—	—	1	2	
	OUT @HL,%p	00 11 10 10 11 $\overline{P_4}$0	$p_3p_2p_1p_0$	3 A C+$2\overline{P_4}$	p	$P[p]\leftarrow M[(H\cdot L)],\quad p=P_4P_3P_2P_1P_0$	—	—	1	2	
	OUT #k,%p	00 10 11 00	$k_3k_2k_1k_0p_3p_2p_1p_0$	2 C k	p	$P[p]\leftarrow k$	—	—	1	2	
	OUTB @HL	00 01 00 10		1	2	$P[2]\cdot P[1]\leftarrow ROM[F\cdot(E+(CF))\cdot M[(H\cdot L)]]$	—	—	1	2	
Branch·Subroutine	BS a	01 10 $a_{11}a_{10}a_9a_8a_7a_6a_5a_4a_3a_2a_1a_0$		6 a_H a_M	a_L	If SF=1 then(PC)←a else null.	—	—	—	2	*5
	BSS a	10 $d_5d_4d_3d_2d_1d_0$		8+d_Hd_L		If SF=1 then(PC)←a else null,a=(PC)⟨11:6⟩·d	—	—	—	1	*5
	CALL a	00 10 0 $a_{10}a_9a_8a_7a_6a_5a_4a_3a_2a_1a_0$		2 a_H a_M	a_L	STACK[(SPW)]←(PC),(SPW)←(SPW)-1, (PC)←a,0≤a≤2,047	—	—	—	2	*5
	CALLS a	01 11 $n_3n_2n_1n_0$		7	n	STACK[(SPW)]←(PC),(SPW)←(SPW)-1, (PC)←a,a=6n+6(n≠0),134(n=0)	—	—	—	2	*5
	RET	00 10 10 10		2	A	$(SPW)\leftarrow(SPW)+1,(PC)\leftarrow STACK[(SPW)]$	—	—	—	2	
	RETI	00 10 10 11		2	B	$(SPW)\leftarrow(SPW)+1,(FLAG\cdot PC)\leftarrow STACK[(SPW)],(EIF)\leftarrow1$	*	*	*	2	
Other	NOP	00 00 00 00		0	0	no operation	—	—	—	1	

Note 1. Setting Condition of Flag.
"C" indicates the carry output from the most significant position in the addition operation, and "B" indicates the borrow output from the most significant position in the subtraction operation.
"Z" indicates the zero detection signal to which "1" is applied only when either the ALU output of the processing result or all four bits of the data transferred to the accumulator are zero.
The flag is set to "C","$\overline{C}$","$\overline{B}$","Z","$\overline{Z}$","1",or"0" according to the data processing ruesult. The value specified by the function is set to the flag with the mark "*",and the mark "-" denotes no change in the state of the flag.

Note 2. The zero flag is set according to the data set in the accumulator.

Note 3. The flags(ZF,SF)are set according to the result of increment or decrement of the L register.

NOte 4. The carry is the data shifted out from the accumulator.

Note 5. The contents of the program counter indicate the next address of the instruction to be executed.

東芝 Toshiba

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

Operation Code Map

lower / higher	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	TESTP CF	CLR CF	SET CF	TESTP CF	ROLC A	TEST CF	RORC A	INC A	DEC A	INC @HL	DEC @HL	LD A,@HL	XCH A,@HL	TESTP ZF	ST A,@HL
1	MOV H,A	MOV L,A	OUTB @HL	XCH A,EIR	SUBRC A,@HL	ADDC A,@HL	CMPR A,@HL	ADD A,@HL	INC L	DEC L	ST A,@HL+	ST A,@HL-	OR A,@HL	AND A,@HL	XOR A,@HL	
2			CALL a					LD HL,x	XCH HL,x	RET	RETI	OUT #k,%p	ST #k,y	LD y,#k	XCH y,#k	CMPR y,#k	ADD y,#k
3	XCH A,H	XCH A,L	LDH A,@DC+	LDL A,@DC	SET @L	CLR @L	(OP36)	TEST @L	(OP38)	(OP39)	(OP3A)	(OP3B)	LD A,x	XCH A,x	CMPR A,x	ST A,x
4	LD A,#k															
5	SET @HL,b				CLR @HL,b				TEST @HL,b				TEST A,b			
6	BS a															
7	CALLS a															
8	BSS a															
9																
A																
B																
C	LD H,#k															
D	CMPR A,#k															
E	LD L,#k															
F	ST #k,@HL+															

(continued)

1st byte code / 2nd byte code (higher)	OP36 — 36	OP38 — 38	OP39 — 39	OP3A — 3A	OP3B — 3B
0		ADD A,#k			
1		SUBR A,#k			
2		OR A,#k	SET • y,b	IN %p,A	SET %p,b
3		AND A,#k			
4		ADD @HL,#k			
5	EICIR IL,r	SUBR @HL,#k	CLR y,b	IN %p,@HL	CLR %p,b
6		OR @HL,#k			
7		AND @HL,#k			
8		ADD L,#k		OUT A,%p	
9	DICIR IL,r	CMPR L,#k	TEST y,b		TEST %p,b
A				OUT A,%p	
B					
C		ADD H,#k		OUT @HL,%p	
D	CLR IL,r	CMPR H,#k	TESTP y,b		TESTP %p,b
E				OUT @HL,%p	
F					

(caution)

1. Blank codes are reserved.

2. ▭◺ is indicated 2-byte instruction.

345