

RICOH**PRELIMINARY**

EK-030-8905

RP5C72**Digital Signal Processor**

T-49-12-09

The RP5C72 is a high-performance general-purpose digital signal processor (DSP). Its 32-bit internal operation includes a 512-word data RAM (1 word = 16 bits), 4k-word ROM (1 word = 16 bits) that can be assigned to both program and data areas, and a 32-bit multiplier (16 × 16 bits, fixed point operation). It processes at 100 ns per instruction.

The index function of the memory has two sets of hardware (index units) that contain ALUs dedicated to addressing, enabling operations required for various digital signal processing.

By changing software in the built-in program memory, the RP5C72 can be adapted to various applications.

■ FEATURES

- Fast processing: 100 ns/instruction (for fck = 40 MHz)
- 16 × 16 bits → 32-bit MPY/ACC built-in
- 32-bit width ALU built-in (Fixed decimal point operation)
- 4-stage pipeline architecture
- 512 words × 16 bits RAM built-in (for data only)
- 4 k words × 16 bits ROM built-in (mask programmable)
(for data and program)
- Repeat instruction function, HOLD function
- External interrupt/internal condition acknowledge EVENT port (2 pins)
I/O port (2 pins)
- Two EVENT registers and one TIMER register built-in (16-bit width)
- Serial I/O port (3-line × 1)
- Parallel I/O port (8-bit × 1)
- 2-layer metal CMOS process
- 28-pin DIL plastic package
- Power consumption: 630 mW (max.)

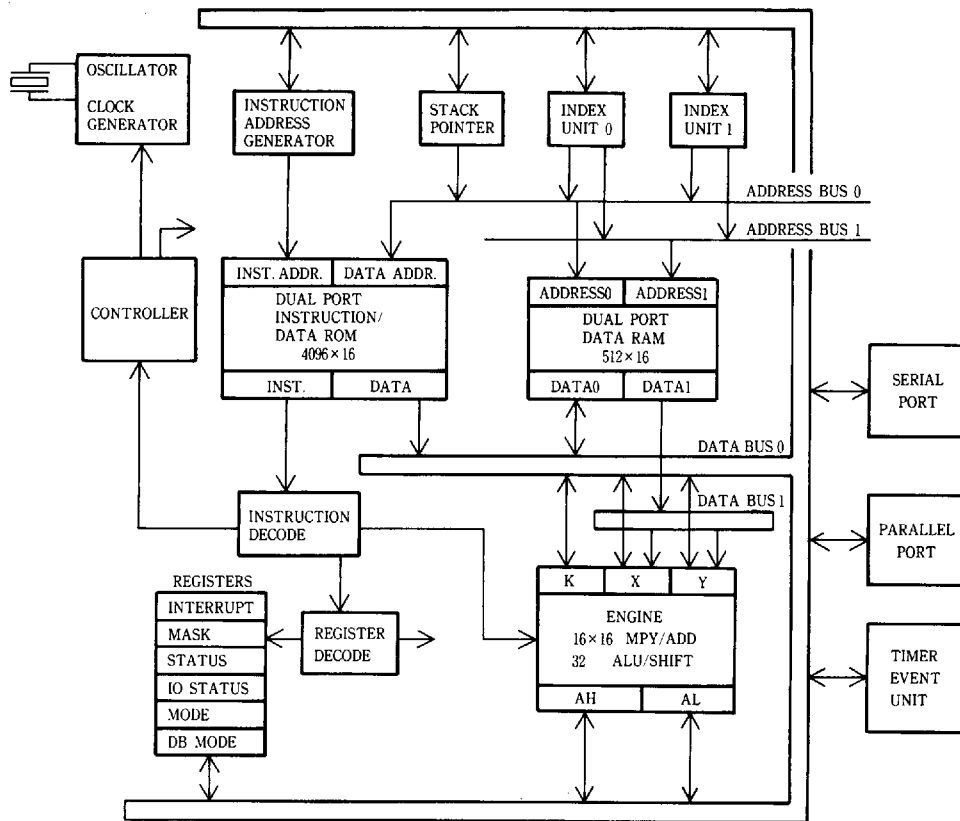
■ APPLICATION

Fast MODEM	Echo canceller
Filter bank	Motor control
Digital filters	Voice composition/recognition
ADPCM	FFT

RICOH

RP5C72

■ BLOCK DIAGRAM



■ PIN CONFIGURATION

YBCLK	□ 1	28	□ VDD
CLOCK	□ 2	27	□ PORT[0]
XSTL1	□ 3	26	□ PORT[1]
XSTL2	□ 4	25	□ PORT[2]
RESET	□ 5	24	□ PORT[3]
HOLD	□ 6	23	□ PORT[4]
TEST	□ 7	22	□ PORT[5]
SO	□ 8	21	□ PORT[6]
SI	□ 9	20	□ PORT[7]
SCLK	□ 10	19	□ PA0
I/O[1]	□ 11	18	□ PCS
I/O[0]	□ 12	17	□ PW
EVENT[1]	□ 13	16	□ POE
VSS	□ 14	15	□ EVENT[0]

■ PIN DESCRIPTION

Pin Name	Pin No.	I/O	Description
VDD	28	I	+5V power supply
VSS	14	I	GND
XSTL 1	3	I	System clock input (external clock input) Connecting a crystal oscillator between XSTL1 and XSTL2 structures an oscillation circuit. (An oscillation circuit is built-in.)
XSTL 2	4	O	
CLOCK	2	O	Internal clock output. Clock signal which is half frequency of XSTL1 is out from the CLOCK pin.
YBCLK	1	O	Internal instruction bus clock output. ¼ of the XSTL1 input frequency. When HOLD is High, the output is fixed to the Low level.
RESET	5	I (Schmit Input)	Reset input. Initializes the chip. Sets bus mode, clears the MASK, INTR, STAT, IOSTAT, MODE, and DEBMODE registers, and initializes the serial/parallel port. Starts a program from the address of the values stored in memory address' h0000.
HOLD	6	I	Hold internal clock except for serial port, parallel port and timer event unit.
EVENT [0]	15	I/O	EVENT/Compare I/O pin. This pin can be programmed to input or output. Performs sampling timing, real-time timer event, and timer value read.
EVENT [1]	13		
I/O [1]	11	I/O	Programmable I/O port pin. Input : Senses external signals and generates interrupts to DSP. Output: For transfers of data written to internal MODE register bits 2 and 3 to the outside.
I/O [0]	12		
SI	9	I	Serial port data input pin. The input data synchronized with the SCLK Low timing is an input enable signal (SIEN). The input data synchronized with the SCLK High timing is SI data.
SO	8	O	Serial port data output pin. The output data synchronized with the SCLK Low timing is an output enable signal (SOEN). The output signal synchronized with the SCLK High timing is SO data.

RP5C72

Pin Name	Pin No.	I/O	Description
SCLK	10	O	Serial clock output. Synchronized with the DSP internal clock, 1/16 of the original oscillation frequency (XSTL1) is output. Example) When fck = 40 MHz, SCLK = 2.5 MHz (400 ns cycle)
PORT 0 ~ PORT 7	27~20	I/O	8-bit parallel port data pin.
PW	17	I	Parallel port write strobe. When PW = "L" and PCS = "L", the PORT is input state.
POE	16	I	Parallel port output enable. When POE = "L" and PCS = "L", the PORT is output state.
PCS	18	I	Parallel port chip select.
PA 0	19	I	Parallel port address select. When PA0 = "L", selects the lower eight bits of the parallel port register (16-bit width) When PA0 = "H", selects the upper eight bits of the parallel port register (16-bit width)

Parallel port control pin assignment

PCS	POE	PW	PA 0	PORT[7] ~ PORT[0]	Selected register
0	0	1	0	Lower 8 bit read	Lower eight bits of parallel port output
0	0	1	1	Upper 8 bit read	Upper eight bits of parallel port output
0	1	0	0	Lower 8 bit write	Lower eight bits of parallel port input
0	1	0	1	Upper 8 bit write	Upper eight bits of parallel port input
0	0	0	x	Prohibition	
0	1	1	x	Hi - Z	No Transfer
1	x	x	x	Hi - Z	No Transfer

x : Don't Care.

TEST	7	I (Schmit Input)	Test mode input. Using this terminal and the RESET pin starts the test mode. Set to VDD level normally.
------	---	---------------------	---

Note: Hi-Z: high impedance state

■ ABSOLUTE MAXIMUM RATING

Symbol	Parameter	Condition	Ratings	Unit
V _{cc}	Supply Voltage	V _{ss} = 0V	-0.3 ~ +7.0	V
V _i	Input Voltage		-0.3 ~ V _{cc} +0.3	V
V _o	Output Voltage		-0.3 ~ V _{cc} +0.3	V
PD	Power Consumption	T _a = 25°C	1.0	W
Topr	Operating Temperature		0 ~ 70	°C
Tstg	Storage Temperature		-40 ~ +125	°C

■ RECOMMENDED CONDITIONS (T_a = 0 ~ +70°C, V_{ss} = 0V)

Symbol	Parameter		Min.	Typ.	Max.	Unit
V _{cc}	Supply Voltage		4.75	5.0	5.25	V
V _{IH}	"H" Input Voltage	TTL Level	2.0		V _{cc} +0.3	V
		CMOS Level	3.5		V _{cc} +0.3	V
		Schmitt Input Level	2.4		V _{cc} +0.3	V
V _{IL}	"L" Input Voltage	TTL Level	-0.3		0.8	V
		CMOS Level	-0.3		1.5	V
		Schmitt Input Level	-0.3		0.6	V

■ DC ELECTRICAL CHARACTERISTICS (Ta = 0 ~ 70°C, Vcc = 5V ±5%)

Symbol	Parameter		Condition	Min.	Typ.	Max.	Unit
ILI	Input Leakage Current		V _{IN} = 0V ~ V _{CC}	-10		10	μA
ILO	Output Off Leakage Current		V _O = 0V ~ V _{CC}	-10		10	μA
V _{IH}	"H" Input Voltage	TTL Level		2.0		V _{CC} + 0.3	V
		Schmitt Input Level (RESET, TEST)		2.4			V
		Clock Input Level (XSTL 1)		4.0			V
V _{IL}	"L" Input Voltage	TTL Level		-0.3		0.8	V
		Schmitt Input Level (RESET, TEST)		-0.3		0.6	V
		Clock Input Level (XSTL 1)		-0.3		1.0	V
V _{OH}	"H" Output Voltage		V _{CC} = Min I _{OH} = -0.4 mA	2.4			V
V _{OL}	"L" Output Voltage		V _{CC} = Min I _{OL} = 2 mA			0.5	V
I _{CC}	Operating Current		V _{CC} = Max note) V _{IN} = V _{SS} or V _{CC} f _{CK} = 40 MHz			120	mA

Note 1) Output pins are measured with output unloaded
Input pins are fixed in GND or V_{CC}.

■ Capacitance (Ta = 25°C, f = 1 MHz)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
C _I	Input Capacitance	V _{IN} = 0V note 2)			5	pF
C _O	Output Capacitance	V _{OUT} = 0V note 2)			7	pF

Note 2) This parameter is periodically sampled and not 100% tested.

■ ELECTRICAL CHARACTERISTICS

1. Clock characteristics

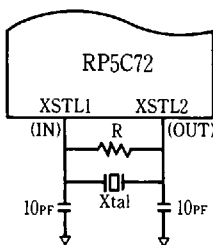
RP5C72 can use both internal oscillation circuit (Crystal oscillation) and external oscillator clock input.

1-(1) Internal clock circuit

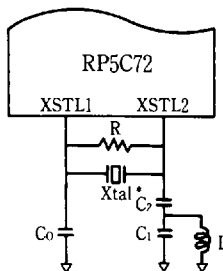
The internal clock circuit starts oscillation when inserting a crystal oscillator between XSTL 1 and XSTL 2 pins.

〈Recommended oscillation circuit〉

$f_{ck} \leq 20\text{MHz}$



$20\text{MHz} < f_{ck} \leq 40\text{MHz}$



Internal Clock Option

Condition

$$(2\pi)^2 \left(\frac{2f_{ck}}{3}\right)^2 C_1 L \approx 1$$

*) 3rd overtone crystal

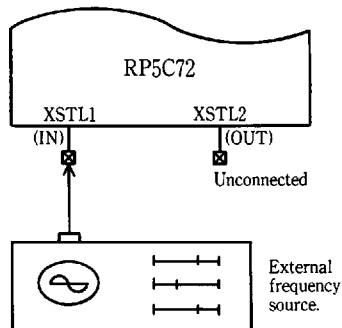
1-(2) Load Condition (Ta = 0°C ~ 70°C, Vcc = 5V ±5%)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
f _{ck}	Crystal Frequency		1		40	MHz
C ₀	Load Capacitance of XSTL 1			15		pF
C ₁	Load Capacitance of XSTL 2			15		pF
C ₂	Load Capacitance of XSTL 2			0.01		μF
L	Load Inductance of XSTL 2			2.2		μH
R	Feed Back Register			1		MΩ

RP5C72

1-(3) External Clock

As shown in the figure on the right, an external clock device can input clock directly to the XSTL 1 pin. At that time, open the XSTL2 pin.

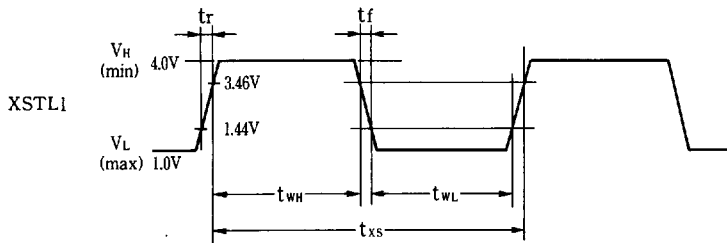


1-(4) External Clock Input Timing (Ta=0°C~70°C, Vcc=5V±5%)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t _{xs}	External Clock Input Cycle Time		25		1000	ns

Note) Duty ratio of the external clock : $40\% \leq (t_r + t_{WH})/t_{xs} \leq 60\%$

(Timing Chart)

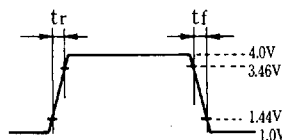


1-(5) Clock Output Switching Characteristics (Ta = 0°C ~ 70°C, Vcc = 5V ±5%)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t _{CLr}	CLOCK Output Rise Time	C _L = 100pF			8	ns
t _{CLf}	CLOCK Output Fall Time	C _L = 100pF			8	ns
t _{CLW}	CLOCK Output High Pulse Width		t _{xs} - 8	t _{xs}	t _{xs} + 8	ns
t _{YBr}	YBCLK Output Rise Time	C _L = 100pF			8	ns
t _{YBf}	YBCLK Output Fall Time	C _L = 100pF			8	ns
t _{YBW}	YBCLK Output High Pulse Width		2t _{xs} - 8	2t _{xs}	2t _{xs} + 8	ns
t _{dxor}	XSTL 1 → CLOCK ⌋ Delay Time	C _L = 100pF			25	ns
t _{dxof}	XSTL 1 → CLOCK ⌋ Delay Time	C _L = 100pF			25	ns
t _{dxyr}	XSTL 1 → YBCLK ⌋ Delay Time	C _L = 100pF			35	ns
t _{dxyf}	XSTL 1 → YBCLK ⌋ Delay Time	C _L = 100pF			35	ns

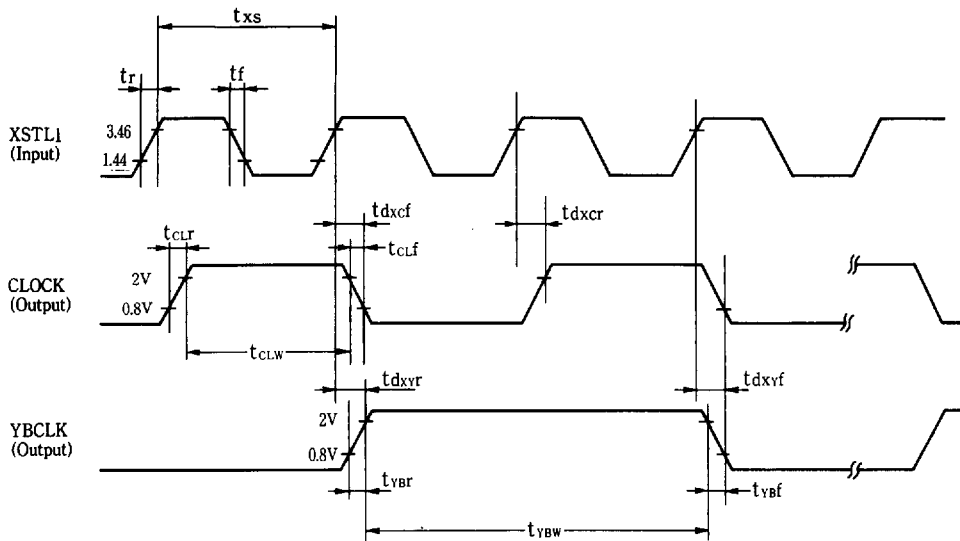
Rise/fall time of clock input signal
 The following times are recommended:

- Rise time (tr) : 5 ns max.
- Fall time (tf) : 5 ns max.



RP5C72

<Timing Chart>



Note) Timing measurements are referenced to and from a low voltage of 0.8V and a high voltage of 2.0V unless otherwise noted.

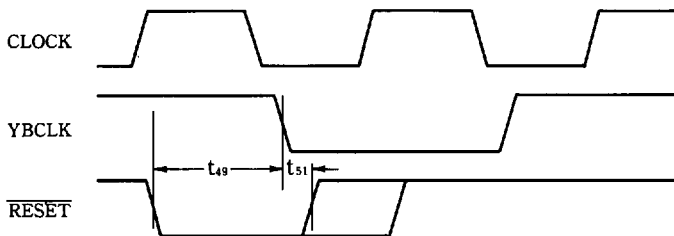
2. Control Pin

2-(1) Soft Reset (Ta = 0°C ~ 70°C, Vcc = 5V±5%)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t ₄₉	YBCLK → $\overline{\text{RESET}}$ Setup Time		40		4t _{xs} - 10	ns
t ₅₁	$\overline{\text{RESET}}$ Active Hold Time		10		4t _{xs} - 50	ns

Note) Soft reset is not performed in the timing that YBCLK stretches internally.

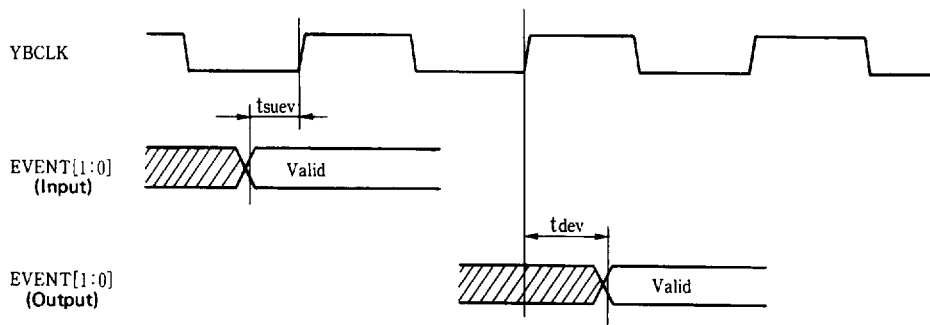
<Timing Chart>



2-(2) EVENT Pin ($T_a = 0^{\circ}\text{C} \sim 70^{\circ}\text{C}$, $V_{cc} = 5\text{V} \pm 5\%$)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t_{suev}	Setup Time of EVENT Input before YBCLK \downarrow		30		$4t_{xs} - 10$	ns
t_{dev}	EVENT Output Delay Time		0		35	ns

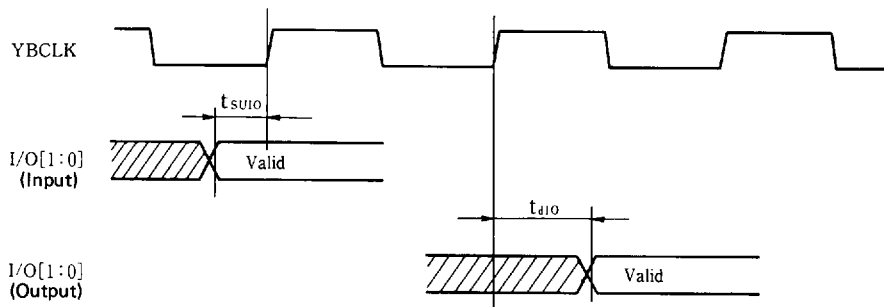
<Timing Chart>



2-(3) I/O Pin ($T_a = 0^{\circ}\text{C} \sim 70^{\circ}\text{C}$, $V_{cc} = 5\text{V} \pm 5\%$)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t_{suIO}	Setup Time of I/O Input before YBCLK \downarrow		30		$4t_{xs} - 10$	ns
t_{dIO}	I/O Output Delay Time		0		35	ns

<Timing Chart>

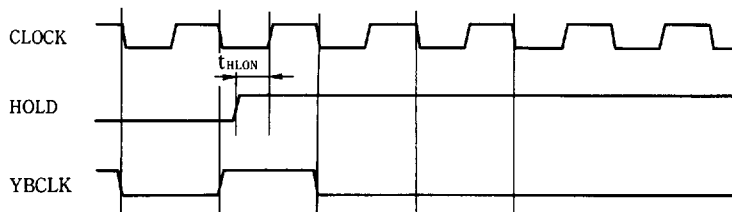


RP5C72

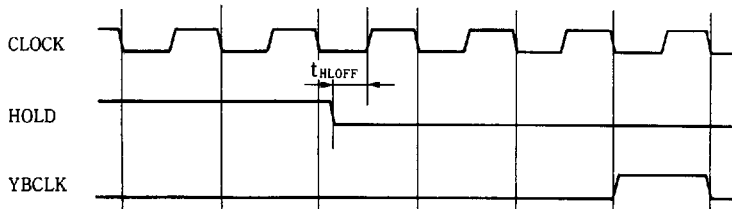
2-(3) HOLD Pin ($T_a = 0^{\circ}\text{C} \sim 70^{\circ}\text{C}$, $V_{cc} = 5\text{V} \pm 5\%$)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t_{HLON}	Setup Time HOLD \uparrow before CLOCK \uparrow		20		$2t_{xs} - 10$	ns
t_{HLOFF}	Setup Time HOLD \downarrow before CLOCK \downarrow		20		$2t_{xs} - 10$	ns

<Timing Chart HOLD enable>



<Timing Chart HOLD disable>



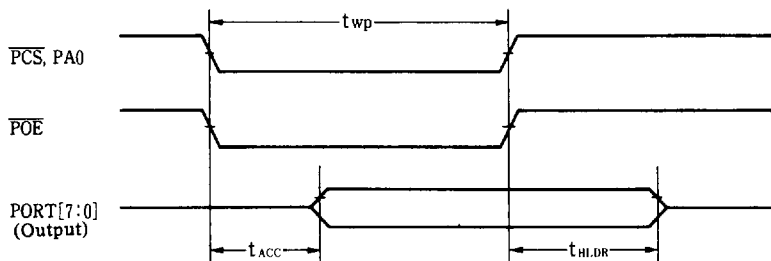
3. I/O Port

3-(1) Parallel Port ($T_a = 0^\circ\text{C} \sim 70^\circ\text{C}$, $V_{CC} = 5\text{V} \pm 5\%$)

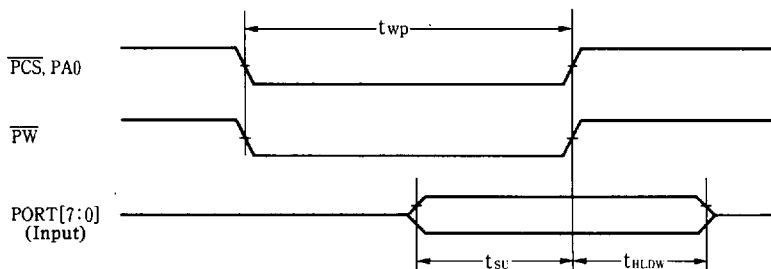
Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t_{ACC}	Parallel Port Output Access Time		0		75	ns
t_{HLDR}	Parallel Port Output Read Hold Time		5			ns
t_{SU}	Parallel Port Input Setup Time		25			ns
t_{HLDW}	Parallel Port Input Write Hold Time		10			ns
t_{WP}	\overline{PCS} , \overline{PAO} , \overline{POE} , \overline{PW} Pulse Width		$4t_{XS} + 50$			ns

(Timing Chart)

Parallel Port Output Timing (External Read)



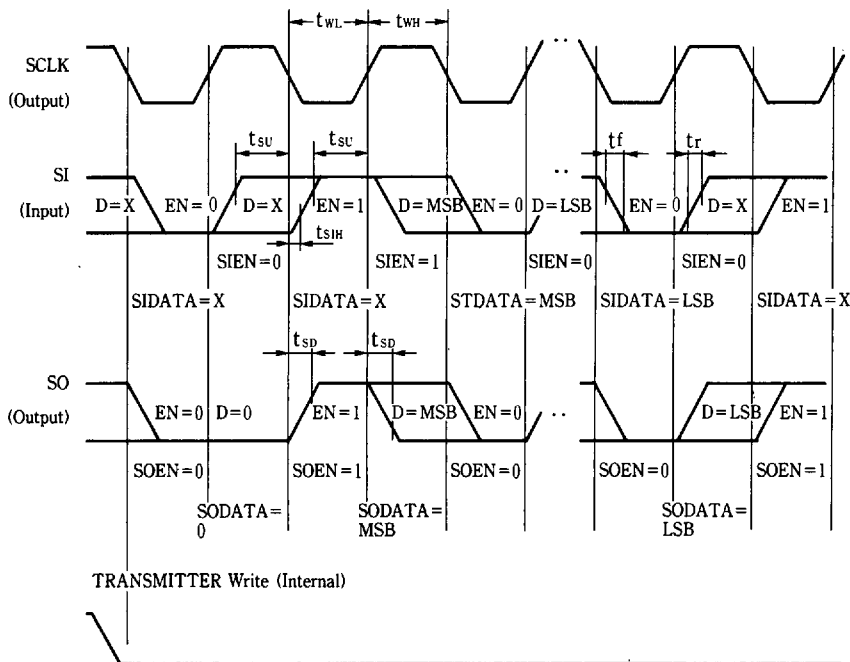
Parallel Port Input Timing (External Write)



RP5C72

3-(2) Serial Poat ($T_a = 0^{\circ}\text{C} \sim 70^{\circ}\text{C}$, $V_{cc} = 5\text{V} \pm 5\%$)

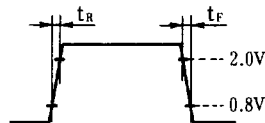
Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
$tw_{L(H)}$	SCLK 'LOW' ('High') Pulse Width		200	8 t_{xs}		ns
t_{su}	SI Input Setup Time		50			ns
t_{sH}	SI Input Hold Time		0			ns
t_{sd}	SO Output Delay Time		0		25	ns



Rise/fall time of serial input signal (SI)
The following times are recommended:

Rise time (t_r) : 5 ns (Max.)

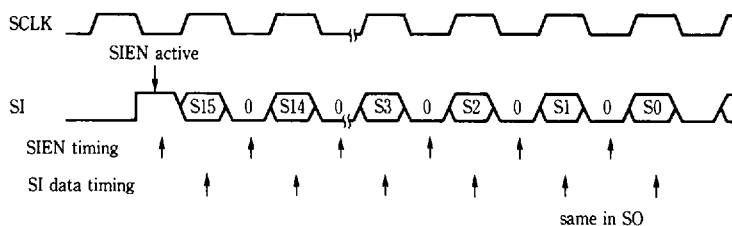
Fall time (t_f) : 5 ns (Max.)



3-(3)

Notes for Serial Port Timing

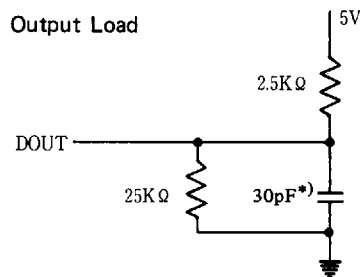
- (1) D = X and SIDATA = X in the timing chart mean the data is "don't care".
- (2) SIEN and SI are time-divided and input to the same input pin (SI).
 SIEN is synchronized with the rising edge of SCLK and input.
 SI data is synchronized with the falling edge of SCLK and input.
- (3) SOEN and SO is time-divided and output from the same output pin (SO).
 SOEN is synchronized with the falling edge of SCLK and output.
 SO data is synchronized with the rising edge of SCLK and output.
- (4) To transfer data, make the enable signal active first (1), then fetch 16-bit data as shown in the figure below:



- (5) SCLK is 1/16 of the XSTL1 frequency (original oscillation frequency).
- (6) The SCLK clock is always output from an external pin.
- (7) SO does not become Hi-Z state.

4. AC Test Condition

Input Pulse Levels	GND to 3.0V
Input Rise and Fall times	10ns
Input Timing Levels	1.5V
Output Reference Levels	1.5V
Output Load	See Figure



*) Includes scope and jig.

RP5C72

■ INSTRUCTIONS

● Description of Symbols

Symbol	Meaning	Symbol	Meaning
A	Accumulator (32-bit width)	SX(XX)	Sign-extends MSB.
AH	Upper 16-bits of accumulator	1 (a)	Increase in the instruction cycle (case a). The case that the instruction cycle is increased by one in the edual (external dual) mode.
AL	Lower 16-bits of accumulator		
X, Y, K	Input registers for ENGINE (MPY, ALU etc.)		
X @ Y	Configures a 32-bit width making the upper 16-bits the X register and lower 16-bits the Y register.	1 (b)	Increase in the instruction cycle (case b). The case that the instruction cycle is increased by one when transferring data to the register in the INDEX UNIT (r, i, p, t) and the stack pointer.
X @ 0	Makes the upper 16-bits the X register and makes the lower 16-bits all zeros, and configures a 32-bit width.	1 (c)	Increase in the instruction cycle (case c). The case that the instruction cycle is increased by one when no jump and call occur by the control instruction (JMPRNZ CALLRNZ) of one instruction cycle.
SP	Stack pointer (16-bit width)		
PC	Program counter (16-bit width)		
(aa)	Absolute address		
(ra)	Relative address		
freq	Fast register (for types, see the internal register list (P25).)		
hxxxx	Hexadecimal data display	XX, ?	Don't care

● Description of Operand Symbols

The operands are in the same form as the symbols for assembler description.

Symbol	Field size	Meaning
(eidx)	11 bits	eidx mode bit (*1)
(dir. ix)	8 bits	Direct indexing mode bit (*1)
(const n)	n bit	n-bit-constant
(freq)	4 bits	Fast register address
(reg)	5 bits	Address of all registers (fast and slow registers)
(src)	5 bits	Source register code of data transfer instruction (XFR)
(dst)	5 bits	Destination register code of data transfer instruction (XFR)
(cond)	5 bits	Condition code (*2)
(dest)	n bit	Jump destination value of jump and call instructions (n bits)
(imm n)	n bit	Immediate data value

*1: See the INDEXING DEFINITION TABLE. (P24)

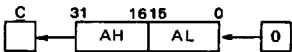
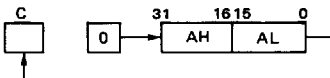
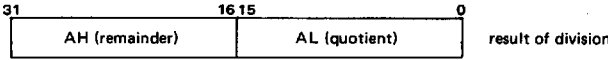
*2: For details, see the CONDITION TABLE. (P26)

RP5C72

13.	ILLOP1	Illegal op-code 1	1/1	D group
	Operation :	Sets bit 15 (ILLOP) of the IOSTAT register. (PC) + 1 → PC	FLAG :	No change
14.	ILLOP2	Illegal op-code 2	1/1	D group
	Operation :	Sets bit 15 (ILLOP) of the IOSTAT register. (PC) + 1 → PC	FLAG :	No change
15.	ILLOP3	Illegal op-code 3	1/1	D group
	Operation :	Sets bit 15 (ILLOP) of the IOSTAT register. (PC) + 1 → PC	FLAG :	No change
16.	ILLOP4	Illegal op-code 4	1/1	D group
	Operation :	Sets bit 15 (ILLOP) of the IOSTAT register. (PC) + 1 → PC	FLAG :	No change
17.	ILLOP5	Illegal op-code 5	1/1	D group
	Operation :	Sets bit 15 (ILLOP) of the IOSTAT register. (PC) + 1 → PC	FLAG :	No change
18.	ILLOP6	Illegal op-code	1/1	D group
	Operation :	Sets bit 15 (ILLOP) of the IOSTAT register. (PC) + 1 → PC	FLAG :	No change
19.	ILLOP7	Illegal op-code	1/1	D group
	Operation :	Sets bit 15 (ILLOP) of the IOSTAT register. (PC) + 1 → PC	FLAG :	No change
20.	JMPA	Conditional absolute address jump instruction	2/2	C group
	Operation :	Condition (see the condition table.) When true : (aa) → PC else : (PC) + 2 → PC	FLAG :	No change
21.	JMPR	Conditional relative address (-64 to +63) jump instruction	1/1+1(c)	C group
	Operation :	Condition (see the condition table.) When true : (PC) + SX(ra) → PC else : (PC) + 1 → PC	FLAG :	No change
22.	LDB	One-byte data load instruction (fast register only)	1/1+1(b)	B group
	Operation :	Sign-extends 8-bit data, and loads it to the fast register as 16-bit data (see the register classification table). (PC) + 1 → PC SX (8-bit data) → freg	FLAG :	No change
23.	LDIX	Index load instruction (fast register only)	1/1+1(b)	B group
	Operation :	(1) ((r0) + SX(ra)) → freg, (PC) + 1 → PC (2) ((r1) + SX(ra)) → freg, (PC) + 1 → PC	FLAG :	No change

24.	LDW	1-word (16-bit) data load instruction (all registers) Operation : (PC) + 2 → PC 16-bit data → reg (fast/slow register)	2/2+1(b) FLAG : No change	B group
25.	LPC	Conditional loop instruction Operation : While the condition is true, repeats execution of the next instruction. (PC) + 1 → PC After that, if the condition is : True : (PC) → PC else : (PC) + 1 → PC (For conditions, see the condition table.)	1/1+n+1 FLAG : No change	C group
26.	LPI	Immediate loop instruction Operation : Repeats the execution of the next instruction as many times as indicated by the lower 7-bits of the instruction. (PC) + 1 → PC After that, if n > 0, (PC) → PC, n-1 → n if n = 0, (PC) + 1 → PC n : lower 7-bits of the instruction code (0 ≤ n ≤ 127)	1/1+n+1 FLAG : No change	C group
27.	MAC	Adds the accumulator and the result of multiplication of the X and Y registers. Operation : (A) + (X) × (Y) → A, (PC) + 1 → PC	1/1+1(a) FLAG : V, C, N, Z	A group
28.	MACAX	Adds the X@0 register (32-bit) and the result of multiplication of the upper 16-bits of the accumulator and the Y register. Operation : (X @ 0) + (AH) × (Y) → A, (PC) + 1 → PC	1/1+1(a) FLAG : V, C, N, Z	A group
29.	MACK	Adds the X@0 register and the result of multiplication of the K and Y registers, and outputs to the accumulator. Operation : (X @ 0) + (K) × (Y) → A, (PC) + 1 → PC	1/1+1(a) FLAG : V, C, N, Z	A group
30.	MSB	Subtracts the result of multiplication of the X and Y registers from the accumulator. Operation : (A) - (X) × (Y) → A, (PC) + 1 → PC	1/1+1(a) FLAG : V, C, N, Z,	A group
31.	MUL	Outputs the result of the multiplication of the X and Y registers to the accumulator. Operation : (X) × (Y) → A, (PC) + 1 → PC	1/1+1(a) FLAG : V, N, Z	A group
32.	OR	Logical add of the accumulator and the X@Y register. Operation : (A) OR (X @ Y) → A, (PC) + 1 → PC	1/1+1(a) FLAG : N, Z, B	A group
33.	ORXY	Logical add of the X@0 register and the Y@0 register. Operation : (X @ 0) OR (Y @ 0) → A, (PC) + 1 → PC	1/1+1(a) FLAG : N, Z, B	A group
34.	POP	Draws data from the stack to the register. Operation : (SP) + 1 → SP, (SP) → reg (All registers) (PC) + 1 → PC	1/1+(b) FLAG : No change	B group
35.	PSH	Stores the register value in the stack. Operation : (reg) → (SP), (SP) - 1 → SP (PC) + 1 → PC	1/1+1(b) FLAG : No change	B group
36.	RTI	Return from the interrupt routine. Operation : (SP) + 1 → SP (SP) → PC Return from the interrupt routine by storing the value again from the stack pointer to the program counter.	1/3 FLAG : No change	C group

RP5C72

37.	RTS	Return from subroutine Operation : (SP) + 1 → SP (SP) → PC Return from the subroutine by storing the value again from the stack pointer to the program counter.	1/3 FLAG : No change	C group
38.	SHL	Accumulator value logical shift instruction (1-bit to the left) Operation : (A) shift left 1-bit → A, (PC) + 1 → PC	1/1+1(a) FLAG : C, N, Z, B	A group
				
39.	SHR	Accumulator value logical shift instruction (1-bit to the right) Operation : (A) shift right 1-bit → A, (PC) + 1 → PC	1/1+1(a) FLAG : C, N, Z, B	A group
				
40.	STIX	Index store from the fast register to the memory Operation : (freg) → (r0) + sx(ra) (freg) → (r1) + sx(ra)	1/1 FLAG : No change	B group
41.	SUB	Subtracts the X@Y register value from the accumulator. Operation : (A) - (X@Y) → A, (PC) + 1 → PC	1/1+1(a) FLAG : V, C, N, Z, B	A group
42.	SUBC	Conditional subtract instruction Operation : If (A) - (X@Y) < 0, (A) × 2 → A, LSB = 0 If (A) - (X@Y) ≥ 0, [(A) - (X@Y)] × 2 → A LSB = 1 (Division) Using the loop instruction and SUBC in combination, the division results (quotient and remainder) can be obtained by the accumulator.	1/1 FLAG : V, C, N, Z	A group
				
43.	SUBI	Subtracts the accumulator value from the X@Y register. Operation : (X @ Y) - (A) → A, (PC) + 1 → PC	1/1+1(a) FLAG : V, C, N, Z, B	A group
44.	SUBXY	Subtracts the Y@0 register value from the X@0 register. Operation : (X @ 0) - (Y @ 0) → A, (PC) + 1 → PC	1/1+1(a) FLAG : V, C, N, Z, B	A group
45.	SUBYX	Subtracts the X@0 register value from the Y@0 register. Operation : (Y @ 0) - (X @ 0) → A, (PC) + 1 → PC	1/1+1(a) FLAG : V, C, N, Z, B	A group
46.	XFR	Data transfer instruction between any registers Operation : (reg) → reg, (PC) + 1 → PC reg is all registers (16-bit width).	1/1+1(b) FLAG : No change	B group
47.	XOR	Exclusive logical add of the accumulator and the X@Y register Operation : (A) XOR(X @ Y) → A, (PC) + 1 → PC	1/1+1(a) FLAG : N, Z, B	A group
48.	XORXY	Exclusive logical add of the X@0 register and the Y@0 register Operation : (X @ 0) XOR (Y @ 0) → A, (PC) + 1 → PC	1/1+1(a) FLAG : N, Z, B	A group

■ INSTRUCTION SET SUMMARY

[A] Arithmetic/Logic Instruction					Instruction Bit Pattern															
MNEMONIC	WORDS	CYCLES	DESCRIPTION	OPERANDS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENOP	1	1+1(a)	Indexing only	(eidx)	0	0	0	0	0	← eidx →										
ADD	1	1+1(a)	A ← (A) + (X @ Y)	(eidx)	0	0	0	0	1	← eidx →										
ADDXY	1	1+1(a)	A ← (X @ 0) + (Y @ 0)	(eidx)	0	0	0	1	0	← eidx →										
SUB	1	1+1(a)	A ← (A) - (X @ Y)	(eidx)	0	0	0	1	1	← eidx →										
SUBXY	1	1+1(a)	A ← (X @ 0) - (Y @ 0)	(eidx)	0	0	1	0	0	← eidx →										
SUBI	1	1+1(a)	A ← (X @ Y) - (A)	(eidx)	0	0	1	0	1	← eidx →										
SUBYX	1	1+1(a)	A ← (Y @ 0) - (X @ 0)	(eidx)	0	0	1	1	0	← eidx →										
SUBC	1	1	If (A)-(X@Y) < 0 then (A)*2- A.LSB=0 (A)-(X@Y) < 0 then (A-X@Y)*2 → A, LSB=1 The Result AH = the remainder AL = the quotient	— — —	1	1	0	1	0	1	0	1	1	1	0	0	0	0	1	0
MUL	1	1+1(a)	A ← (X) * (Y)	(eidx)	0	0	1	1	1	← eidx →										
MAC	1	1+1(a)	A ← (A) + (X) * (Y)	(eidx)	0	1	0	0	0	← eidx →										
MACAX	1	1+1(a)	A ← (X @ 0) + (AH) * (Y)	(eidx)	0	1	0	0	1	← eidx →										
MACK	1	1+1(a)	A ← (X @ 0) + (K * Y)	(eidx)	0	1	0	1	0	← eidx →										
MSB	1	1+1(a)	A ← (A) - (X) * (Y)	(eidx)	0	1	0	1	1	← eidx →										
AND	1	1+1(a)	A ← (A) AND (X @ Y)	(eidx)	0	1	1	0	0	← eidx →										
ANDXY	1	1+1(a)	A ← (X @ 0) AND (Y @ 0)	(eidx)	0	1	1	0	1	← eidx →										
OR	1	1+1(a)	A ← (A) OR (X @ Y)	(eidx)	0	1	1	1	0	← eidx →										
ORXY	1	1+1(a)	A ← (X @ 0) OR (Y @ 0)	(eidx)	0	1	1	1	1	← eidx →										
XOR	1	1+1(a)	A ← (A) XOR (X @ Y)	(eidx)	1	0	0	0	0	← eidx →										
XORXY	1	1+1(a)	A ← (X @ 0) XOR (Y @ 0)	(eidx)	1	0	0	0	1	← eidx →										
SHR	1	1+1(a)	A ← (A) SHR 1 (right)	(eidx)	1	0	0	1	0	← eidx →										
SHL	1	1+1(a)	A ← (A) SHL 1 (left)	(eidx)	1	0	0	1	1	← eidx →										

[B] Load/Store Instruction					Instruction Bit Pattern																
MNEMONIC	WORDS	CYCLES	DESCRIPTION	OPERANDS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
LDW	2	2+1(b)	Load 16 bit immediate value into register.	(const 16) (reg)	1	1	0	1	0	1	0	1	0	1	0	← reg →					
LDB	1	1+1(b)	Load 8 bit immediate value into fast register with sign extend.	(const 8) (freg)	← imm 16 →										← imm 8 →						
LDIX	1	1+1(b)	Indexed load to fast register.	(dir. ix) (freg)	1	0	1	0	← freg →	← idx →											
STIX	1	1	Indexed store to fast register.	(freg) (dir. ix)	1	0	1	1	← freg →	← idx →											
XFR	1	1+1(b)	Register transfer, any registers. src -> dst (src and dst are reg)	(src) (dst)	1	1	0	1	0	0	← src →	← dst →									
PSH	1	1+1(b)	Push reg. onto stack.	(reg)	1	1	0	1	0	1	0	1	0	0	1	← reg →					
POP	1	1+1(b)	Pop reg. from stack	(reg)	1	1	0	1	0	1	0	1	0	0	0	← reg →					

RP5C72

[C] Control Flow Instruction					Instruction Bit Pattern															
MNEMONIC	WORDS	CYCLES	DESCRIPTION	OPERANDS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPC	1	1+n+1	Loop conditional	(cond)	1	1	0	1	0	1	0	0	0	?	?	?	?	?	?	?
LPI	1	1+n+1	Loop immediate	(const 7)	1	1	0	1	0	1	0	0	1	←	imm 7	→				
JMPA	2	2	Conditional absolute jump	(cond) (dest)	1	1	0	1	0	1	0	1	1	0	0	←	cond	→		
CALLA	2	2	Conditional absolute call	(cond) (dest)	1	1	0	1	0	1	0	1	1	0	1	←	cond	→		
CALLS	1	2	Call location 16	(eidx)	1	1	0	1	1	←	eidx	→								
JNPR	1	1+1(c)	Conditional jump relative +63/-64	(cond) (dest)	1	1	1	0	←	cond	→	←	imm 7	→						
CALLRNZ	1	1+1(c)	Not zero call relative +1023/-1024	(dest)	1	1	1	1	0	←	imm 11	→								
CALLRT	1	1	Unconditional call relative +1023/-1024	(dest)	1	1	1	1	1	←	imm 11	→								
RTS	1	3	Return from subroutine by restoring PC from the SP.	—	1	1	0	1	0	1	0	1	1	1	0	0	0	0	1	1
RTI	1	3	Return from interrupt service	—	1	1	0	1	0	1	0	1	1	1	0	0	0	1	1	1
IEN	1	1	Enable interrupts	—	1	1	0	1	0	1	0	1	1	1	0	0	0	0	0	0
IDS	1	1	Disable interrupts (used after IEN)	—	1	1	0	1	0	1	0	1	1	1	0	0	0	0	0	1

[D] Illegal Opcode Instruction					Instruction Bit Pattern															
MNEMONIC	WORDS	CYCLES	DESCRIPTION	OPERANDS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ILLOP0	1	1	Illegal opcode 0	(const 5)	1	1	0	1	0	1	0	1	0	1	1	←	imm 5	→		
ILLOP1	1	1	Illegal opcode 1	(const 5)	1	1	0	1	0	1	0	1	1	1	1	←	imm 5	→		
ILLOP2	1	1	Illegal opcode 2	(const 4)	1	1	0	1	0	1	0	1	1	0	1	←	imm 4	→		
ILLOP3	1	1	Illegal opcode 3	(const 3)	1	1	0	1	0	1	0	1	1	1	0	0	←	imm 3	→	
ILLOP4	1	1	Illegal opcode 4	—	1	1	0	1	0	1	0	1	1	1	0	0	0	1	0	0
ILLOP5	1	1	Illegal opcode 5	—	1	1	0	1	0	1	0	1	1	1	0	0	0	1	0	1
ILLOP6	1	1	Illegal opcode 6	—	1	1	0	1	0	1	0	1	1	1	0	0	0	1	1	0
ILLOP7	1	1	Illegal opcode 7	(const 9)	1	1	0	1	0	1	1	←	imm 9	→						

■ INSTRUCTION CYCLE

	[A] Arithmetic/ Logic	[B] Load/Store	[C] Control Flow	[D] Illegal Opcode
1 W 1 C [Single word Single cycle]	ENOP ADD ADDXY SUB SUBXY SUBI SUBYX SUBC MUL MAC MACAX MACK MSB AND ANDXY OR ORXY XOR XORXY SHR SHL	LDB LDIX STIX XFR PSH POP	JMPR (taken) CALLRT CALLRNZ (taken)	ILLOP0 ILLOP1 ILLOP2 ILLOP3 ILLOP4 ILLOP5 ILLOP6 ILLOP7
1 W 2 C [Single word 2 cycles]	ENOP AND ADDXY SUB SUBXY SUBI SUBYX MUL MAC MACAX MACK MSB AND ANDXY OR ORXY XOR XORXY SHR SHL [external dual (EXDUAL) mode]	LDB n IU LDIX (rx, n) IU POP IU POP SP XFR reg IU XFR reg SP [IU: p0, p1, i0 i1, t0, t1 r0, r1]	JMPR (not taken) CALLRNZ (not taken) CALLS	
1 W 3 C [Single word 3 cycles]			RTS RTI	
2 W 2 C [2 words 2 cycles]		LDW n reg (except IU/SP)	JMPA CALLA	
2 W 3 C [2 words 3 cycles]		LDW n IU LDW n SP		
MISC.			LPC --- 1+n+1 LPI --- 1+n+1	

RP5C72

INDEXING DEFINITION

MODE	Assembler Syntax	Bit Pattern								Descriptions			
		10	9	8	7	6	5	4	3		2	1	0
(EIDX)	(eidx)	e	e	e	e	e	e	e	e	e	e	e	
	edual	0	y	y	y	y	x	x	x	x	x	x	Two fields of five bits. X ← sx(imm8) Y ← sx(imm8) X ← (r0 + sx(idx7)) Y ← (r1 + sx(idx7)) (r0 + sx(idx7)) ← AH (r1 + sx(idx7)) ← AH
	eimm	1	0	0	1	1	1	1	1	1	1	1	
		1	0	1	1	1	1	1	1	1	1	1	
	edx	1	1	0	0	1	1	1	1	1	1	1	
		1	1	0	1	1	1	1	1	1	1	1	
	1	1	1	0	1	1	1	1	1	1	1		
(DIRECT INDEXING)	(direct ix)	u	u	u	u	u	u	u	u	u	u		
	idx	0	i	i	i	i	i	i	i	i	i	freg ← (r0 + sx(idx7)) : LDIX (r0 + sx(idx7)) ← freg : STIX freg ← (r1 + sx(idx7)) (r1 + sx(idx7)) ← freg	
		0	i	i	i	i	i	i	i	i	i		
		1	i	i	i	i	i	i	i	i	i		
	1	i	i	i	i	i	i	i	i	i			

= N.B. = sx -> Sign extend

EDUAL (engine dual) INDEXING DESCRIPTION

Mnemonic	Mnemonic	Pattern	Description
(y index)	(x index)	YYYYY XXXXX	Description for (x index)
nopl	nop0	00000	no operation for IU0
0y	0x	00010	x ← 0
-ly	-lx	00110	x ← -1
!p1+	!p0+	00001	p0 ← p0 + 1
!p1-	!p0-	00111	p0 ← p0 - 1
!p1+i	!p0+i	00011	p0 ← p0 + i0
!p1+r	!p0+r	00100	p0 ← p0 + r0
!pic	!p0c	00101	if p0 = t0 then p0 ← r0, else p0 ← p0+1
ah(p1)	ah(p0)	01000	(p0) ← ah
ah(r1)	ah(r0)	01001	(r0) ← ah
ah(t1)	ah(t0)	01010	(t0) ← ah
ah(p1+i)	ah(p0+i)	01011	(p0 + i0) ← ah
ah(p1+r)	ah(p0+r)	01100	(p0 + r0) ← ah
ah(!pic)	ah(!p0c)	01101	if p0 = t0 then p0 ← r0, else p0 ← p0 + 1, (p0) ← ah
ah(!p1+i)	ah(!p0+i)	01110	p0 ← p0 + i0, (p0) ← ah
ah(!p1+r)	ah(!p0+r)	01111	p0 ← p0 + r0, (p0) ← ah
(p1)y	(p0)x	10000	x ← (p0)
(t1)y	(t0)x	10101	x ← (t0)
(r1)y	(r0)x	11000	x ← (r0)
(p1+)y	(p0+)x	10001	x ← (p0 + 1)
(p1++)y	(p0++)x	10010	x ← (p0 + 2)
(p1-)y	(p0-)x	10111	x ← (p0 - 1)
(p1--)y	(p0--)x	10110	x ← (p0 - 2)
(p1+i)y	(p0+i)x	10011	x ← (p0 + i0)
(p1+r)y	(p0+r)x	10100	x ← (p0 + r0)
(!p1+)y	(!p0+)x	11001	p0 ← p0 + 1, x ← (p0)
(!p1++)y	(!p0++)x	11010	p0 ← p0 + 2, x ← (p0)
(!p1-)y	(!p0-)x	11111	p0 ← p0 - 1, x ← (p0)
(!p1--)y	(!p0--)x	11110	p0 ← p0 - 2, x ← (p0)
(!p1+i)y	(!p0+i)x	11011	p0 ← p0 + i0, x ← (p0)
(!p1+r)y	(!p0+r)x	11100	p0 ← p0 + r0, x ← (p0)
(!pic)y	(!p0c)x	11101	if p0 = t0 then p0 = r0, else p0 ← p0 + 1 : x ← (p0)

(Y index) is same in operation as (X index).

INTERNAL REGISTER SUMMARY (27)

FAST REGISTERS

Reg #	Mnemonic	Description
00	P0	pointer index test } Index Unit 0
01	I0	
02	T0	
03	R0	reload pointer index } Index Unit 1
04	P1	
05	I1	
06	T1	test reload Least significant accumulator word
07	R1	
08	AL	
09	AH	Most significant accumulator word
0A	AZ	LOAD: load AH, zero AL STORE: store AH
0B	X	} Engine registers
0C	Y	
0D	K	
0E	PARPORT	
0F	SERPORT	Parallel Port data register Serial Port data register

SLOW REGISTERS

Reg #	Mnemonic	Description
10	MASK	Interrupt mask
11	INTR	Interrupt
12	STAT	Execution status
13	IOSTAT	Serial, Parallel & debug status
14	MODE	Mode select
15		Reserved
16	DEBMODE	Debug and test mode
17	SP	Stack pointer
18	E0	Event/Compare 0
19	E1	Event/Compare 1
1A		Reserved
1B		Reserved
1C	TIMER	Timer
1D		Reserved
1E	PROGMAT	Program address compare
1F		Reserved

FLAG CONDITION

FLAG CONDITION

Mnemonic	V	C	N	Z	B
ENOP	-	-	-	-	-
ADD	m	m	m	m	m
ADDXY	m	m	m	m	m
SUB	m	m	m	m	m
SUBXY	m	m	m	m	m
SUBI	m	m	m	m	m
SUBYX	m	m	m	m	m
MUL	m	-	m	m	-
MAC	m	m	m	m	-
MACAX	m	m	m	m	-
MACK	m	m	m	m	-
MSB	m	m	m	m	-
SUBC	m	m	m	m	-
AND	-	-	m	m	m
ANDXY	-	-	m	m	m
OR	-	-	m	m	m
ORXY	-	-	m	m	m
XOR	-	-	m	m	m
XORXY	-	-	m	m	m
SHL	-	m	m	m	m
SHR	-	m	m	m	m

Note) The meaning of the symbol are following this ...

V : Over flow flag

C : Carry flag

N : Negative sign flag

Z : Zero flag

B : Bitwise XOR set flag

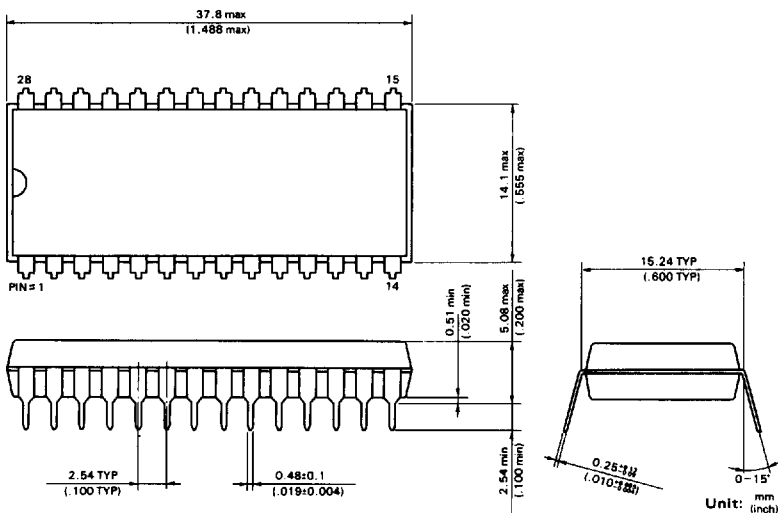
m : Flag bit change

CONDITIONED INSTRUCTION TABLE

Alias (cond)	Pattern cccc	Logic	Description
)s	00000	$N \& V \& Z \mid N \& V \& Z$	Signed strictly greater than
(=s	00001	$(N \& V \& Z \mid N \& V \& Z)$	Signed less than or equal
)=s	00010	$N \& V \mid N \& V$	Signed greater than or equal
(s	00011	$(N \& V \mid N \& V)$	Signed strictly less than
m	00100	N	Sign bit set (minus)
p	00101	\bar{N}	Sign bit reset (plus)
)u	00110	$C \& Z$	Unsigned strictly greater than
(=u	00111	$(C \& Z)$	Unsigned less than or equal
)=u	01000	C	Unsigned greater than or equal
(u	01001	\bar{C}	Unsigned strictly less than
v	01010	V	Overflow
!v	01011	\bar{V}	Not overflow
!iu0	01100	IU0	P0 not equal to T0
!iu1	01101	IU1	P1 not equal to T1
=	01110	Z	Equal
!=	01111	\bar{Z}	Not equal
t	10000	1	Always
!t	10001	0	Never
ah7	10010	AH[7]	AH register bit 7 (AH[7]) set
ah6	10011	AH[6]	AH register bit 6 (AH[6]) set
ah5	10100	AH[5]	AH register bit 5 (AH[5]) set
ah4	10101	AH[4]	AH register bit 4 (AH[4]) set
ah3	10110	AH[3]	AH register bit 3 (AH[3]) set
ah2	10111	AH[2]	AH register bit 2 (AH[2]) set
ah1	11000	AH[1]	AH register bit 1 (AH[1]) set
ah0	11001	AH[0]	AH register bit 0 (AH[0]) set
e0	11010	EVENT[0]	Sense of EVENT[0] pin
e1	11011	EVENT[1]	Sense of EVENT[1] pin
io0	11100	I/O[0]	Sense of I/O[0] pin
io1	11101	I/O[1]	Sense of I/O[1] pin
b	11110	B	Bitwise XOR of AH
!b	11111	\bar{B}	Bitwise XNOR of AH

PACKAGE DIMENSION

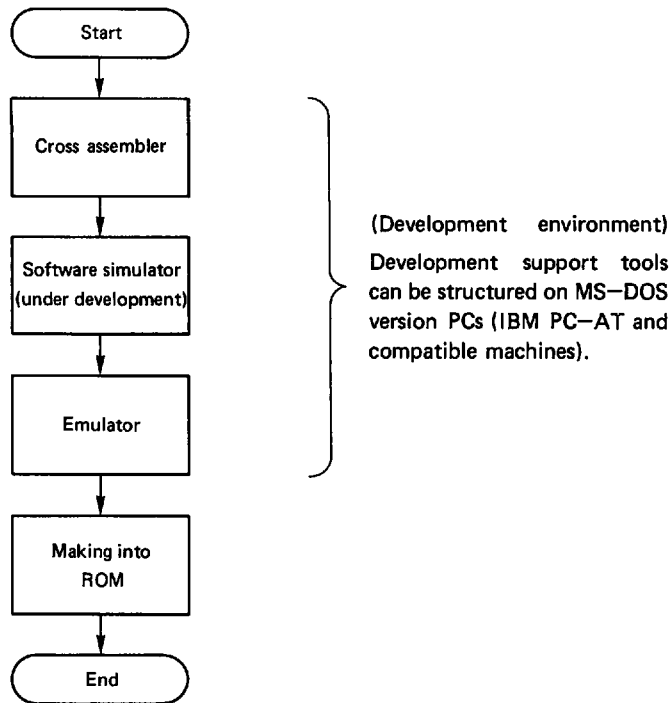
28 pin DIP



■ DEVELOPMENT TOOL

A cross assembler and an emulator are available to efficiently develop application algorithms and software and to debug target systems.

(RP5C72 program development procedure)



● Cross assembler

The cross assembler (XAS71) converts a source code generated by the RP5C72 assembly language into an executable object code.

The source file is generated by an ordinary ASCII file, using an ordinary editor to indicate the instruction in mnemonic. (Use .X71 for the extension of source file)

The object file of the output file is output in Motorola-S format, and can be executed by the emulator or the processor.

The cross assembler (XAS71) can operate on VAX and SUN3 UNIXs and on MS-DOS OSs.

RP5C72

● Emulator

The emulator of the RP5C72 is a support tool for program debugging using actual DSP chips and target systems. It has various functions required for a basic real-time in-circuit emulator.

The main functions are as follows:

- Built-in 12-bit A/D converter for input signals
- Two built-in 16-bit D/A converters for output signals
- Processes the same signals as those given to the RP5C72, with the 28-pin socket.
- Break pointer setting
- Reverse assembler
- Line assembler
- Operation commands for various memories (block transfer and replacement of files and data)
- Develops programs of up to 8 K words.
- Single step function
- A 40-MHz crystal oscillator is built-in. It is also possible to input clocks from an external oscillator.

MS-DOS version PCs (IBM PC-AT and compatible machines) are used as host computers.