RICOH

Microelectronic Specification

RP65C02

CMOS 8-bit MICROCOMPUTER SYSTEM

■ GENERAL DESCRIPTION

The RP65C02 is 8-bit CMOS CPU. It has the instruction set and pins which are fully compatible with the NMOS 6502 CPU, and in addition, with 59 new instructions. It is provided with the features of the CMOS such as the powerdown, standby mode, etc.

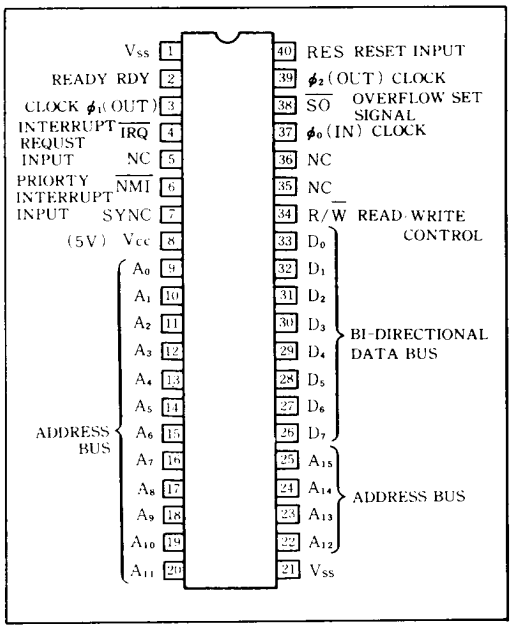
■ FEATURES

- Single power supply 5V operation
- CMOS silicon gate process
- Low power dissipation
- 8 bit bi directional data bus, parallel processing
- 68-type 210 instructions
- Powerful 13-type addressing modes
- Programmable stack pointer
- Maskable interrupt and non maskable interrupt
- 6 type internal registers
- Enable to connect the external memory with up to 64Kbytes
- Reference clock 1~4 MHz
- Executable single instruction
- Computable decimal and binary
- Bus compatible with M6800
- Pin compatible with ROCKWELL R65C02

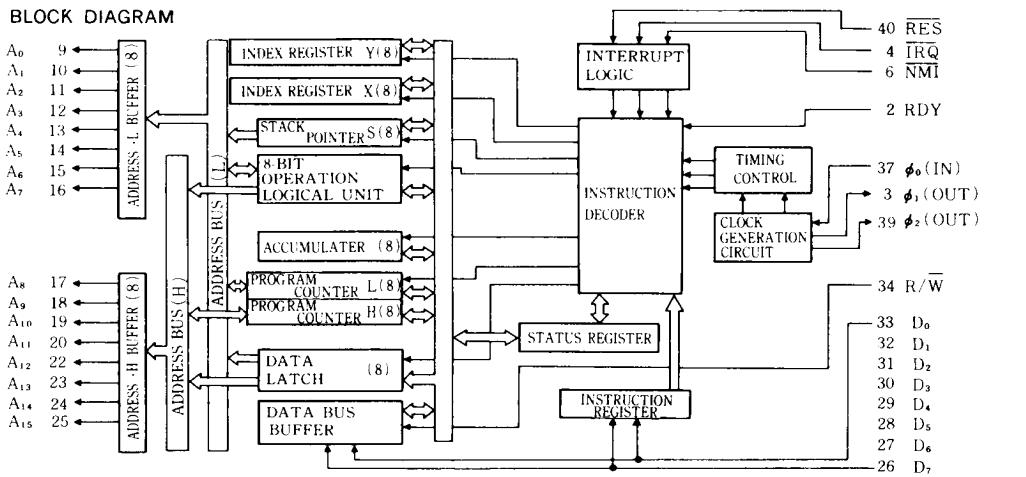
■ APPLICATIONS

- Hand held computer, etc.

■ PIN CONFIGURATION (Top view)



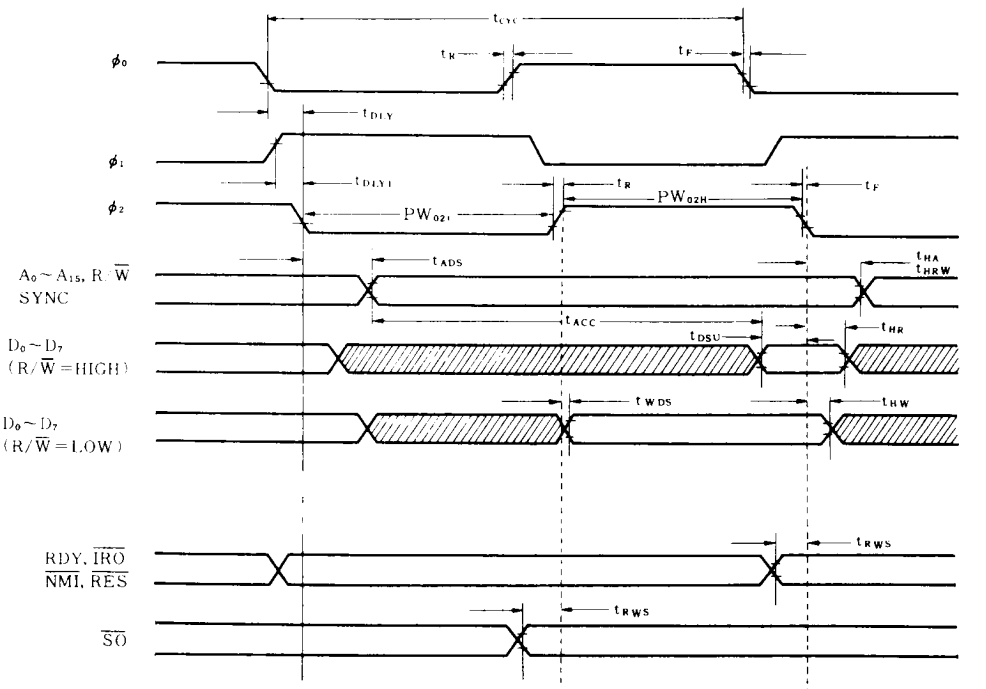
■ BLOCK DIAGRAM



● AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0±5%, Ta = 0℃ ~ 70℃, load 130pF)

Symbol	Parameters	1 MHz		2 MHz		3 MHz		4 MHz		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{CYC}	Cycle time	1000		500		333		250		ns
PW_{02l}	ϕ_2 "Low" Clock Pulse Width	430	5000	210	5000	150	5000	100	5000	ns
PW_{02H}	ϕ_2 "High" Clock Pulse Width	450		220		160		110		ns
t_R, t_F	Clock rising time, Clock Falling Time		25		15		12		10	ns
t_{ADS}	Address Delay Time		225		140		110		90	ns
t_{HA}	Address Hold Time	20		20		15		15		ns
t_{HRW}	R/$\overline{W}$ Hold Time	30		30		30		30		ns
t_{DSU}	Read Data Setup Time	100		50		50		50		ns
t_{HR}	Read Data Hold Time	10		10		10		10		ns
t_{HW}	Write Data Hold Time	30		30		30		30		ns
t_{ACC}	Read Access Time	695		340		254		168		ns
t_{RWS}	Processor Control Setup Time (RDY, S.O, $\overline{IRQ}$, $\overline{NMI}$, $\overline{RES}$)	200		110		80		60		ns
t_{WDS}	Write Data Delay Time		175		100		75		70	ns
t_{DLY}	Delay Time ϕ_0 to ϕ_2		100		100		100		100	ns
t_{DLY1}	Delay Time ϕ_1 to ϕ_2		50		50		50		50	ns

■ TIMING CHART



※ "H" 2.4V, "L" 0.4V

■ ABSOLUTE MAXIMUM RATINGS

Symbol	Parameters	Limits	Unit
V_{CC}	Supply Voltage	−30 ~ +7.0	V
V_I	Input Voltage	−0.3 ~ +7.0	V
P_d	Power Dissipation	500	mW
T_{opr}	Operating Ambient Temperature	0 ~ +70	℃
T_{stg}	Storage Temperature	−40 ~ +125	℃

■ ELECTRICAL CHARACTERISTICS

● DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 ± 5%, V_{SS} = 0V, Ta = 0 ~ +70℃)

Symbol	Parameters		Measuring Conditions	Specified Value Min	Typ	Max	Unit
V_{IH}	Input High Voltage	ϕ_0(in)		2.4		V_{CC}+0.3	V
		Logic		2.0		V_{CC}+0.3	
V_{IL}	Input Low Voltage	ϕ_0(in)		−0.3		0.4	V
		Logic		−0.3		0.8	
I_{IL}	Input Leak Current	Logic	V_{CC} = 5.25V	−30		+10	μA
		ϕ_0(in)	V_I = 0 ~ 5.25V	−10		+10	
I_{OL}	Output Floating Leakage Current Data Line		V_I = 0.4 ~ 2.4V	−10		+10	μA
V_{OH}	Output High Voltage ϕ_1, ϕ_2, SYNC, D_0~D_7, A_0~A_{15}, R/$\overline{W}$		I_{LOAD} = −100μA V_{CC} = 4.75V	2.4			V
V_{OL}	Output Low Voltage ϕ_1, ϕ_2, SYNC, D_0~D_7, A_0~A_{15}, R/$\overline{W}$		I_{LOAD} = 1.6mA V_{CC} = 4.75V			0.4	V
P_d	Power Disspation (no-load)		I_C = 0MHz (standby)		0.1	0.15	mW
			= 1MHz		20	30	mW
			= 2MHz		40	60	mW
			= 3MHz		60	90	mW
			= 4MHz		80	120	mW
			Low Power (RDY = 0)		10	15	mW/MHz
C	Input Capacitance Logic D_0~D_7		V_I = 0V f = 1MHz			10	pF
	SYNC, A_0~A_{15}, R/$\overline{W}$					10	
	ϕ_0(in)					10	
	ϕ_1					30	
	ϕ_2					50	

Note : $\overline{IRQ}$, $\overline{NMI}$ need pull-up resistor.

■ EXPLANATION ON PIN FUNCTION

● Clock Input (ϕ_{0in})

It is the input terminal to generate the system clock in the inside and input the reference clock from the outside. The operating frequency is between 1 and 4 MHz. And when ϕ_{0in} stops at highlevel, the CPU becomes the stand-by mode.

● Clock Output (ϕ_{1out}, ϕ_{2out})

The two signal ϕ_{1out} or ϕ_{2out} output for the system-clock output. These are provided with each device for one part of control bus synchronous signal. ϕ_{1out} is address-time and the address becomes valid at ϕ_{1out} time. ϕ_{2out} is data time and the data becomes valid at ϕ_{2out} time.

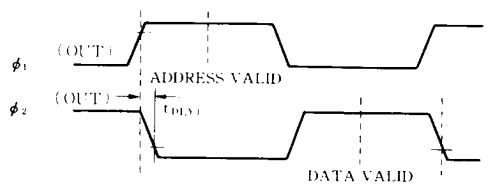


Fig.1 PHASE RELATION BY SYSTEM CLOCK ϕ_{1OUT}, ϕ_{2OUT}

● Address Bus ($A_0 \sim A_{15}$)

$A_0 \sim A_{15}$ constiute a 16-bit address bus. The address that is indicated with these bits are hexadecimal \$0000~FFFF. (decimal 0~65535)

All TTL compatible, these address bus is capable of driving one-standard TTL and 130 pF.

● Data Bus ($D_0 \sim D_7$)

$D_0 \sim D_7$ constitute the 8-bit bidirectional data bus, input or output. All TTL compatible, these address bus is capable of driving one-standard TTL and 130pF.

● Bus Direction Indicative Signal ($R/\overline{W}$)

It is the signal to decide the direction of data bus.

In reading (input data from other device to the CPU) "1" is output, and in writing (output data from the CPU to other) "0" is output. Read or write timing are as shown in Fig.2, Fig.3.

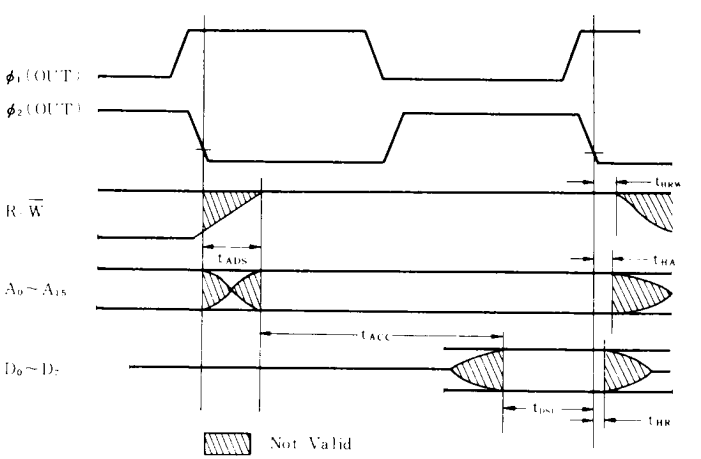


Fig.2 READ MODE TIMING

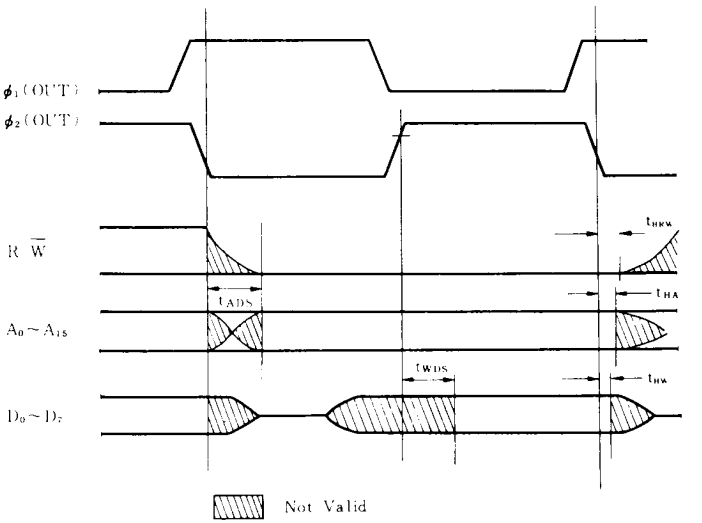


Fig.3 WRITE MODE TIMING

● Ready Signal (RDY)

This input allows the user to single-cycle the microprocessor on all cycles including write cycles. A negative transition to the low state, during or coincident with ϕ_1, will halt the microprocessor with the output address lines reflecting the current address being fetched. This condition will remain through a subsequent ϕ_2 in which the ready signal is low. This feature allows microprocessor interfacing with low-speed memory as well as direct memory access (DMA).

● System Reset ($\overline{RES}$)

The input is used to reset the CPU in a power down state and to start. During that the input is Low level, READ/WRITE to the CPU is not all accepted. When the rising time signal of the pin is detected, the CPU becomes the reset mode at once.

After initial setting time of the 6 clock time, the interrupt mask flag is set, the CPU reads the vector address from each location (FFFC)(FFFD), and sets the program counter.

The input consists of the Schmitt trigger circuit as which power on reset is acted by only CR.

● Interrupt Request Signal ($\overline{IRQ}$, $\overline{NMI}$)

$\overline{IRQ}$ (Interrupt Request)

If the TTL compatible input is the low level, the CPU starts the interrupt operation. When the instruction in execution is finished, the CPU allows the interrupt request, but at the same time, the interrupt mask bit in the status code register is checked, and if not set, the CPU begins the execution of the interrupt sequence. The program counter and status register are loaded with stack, the interrupt mask flag is set so as not to accept any other interrupts. At the end of this cycle, the content of location FFFF load into high order 8-bit of program-counter, and the content of location FFFE load into low order 8-bit of programcounter. The program control is changed a memory vector which is stored these location.

To accept an interrupt, RDY signal should be high level. These are just same with all interruptions. When it is used to the wired OR with this pin, it must use a pullup resistor.

● $\overline{NMI}$ (Nonmaskable Interrupt)

When the falling signal is input in pin, the CPU detects this edge, and starts the nonmaskable interrupt operation.

$\overline{NMI}$ is unconditional interrupt request. When the instruction in execution becomes end, the similar

operation to IRQ is executed regardless of the state of interrupt mask flag.

In the vector address which is loaded to program counter, high order 8-bit are contents of location FFFB, and low order 8-bit are contents of location FFFA. The program counter changes to these addresses. When it uses the wired OR with this pin, it must use the pullup resistor.

$\overline{IPQ}$ and $\overline{NMI}$ are interrupt inputs of hardware which is sampled in the inside of the CPU during ϕ_2 time. After a instruction in execution comes at the end, it executes next interrupt routine from the first ϕ_1 time.

● Overflow Flag Set Signal ($\overline{SO}$)

The overflow flag bit (V) in the status code register is set by the falling edge input to this pin. As this signal is sampled by the rising edge, the input must be synchronized outside.

● Instruction Fetch Cycle Synchronous Signal (SYNC)

This output signal indicates the cycle that microprocesser fetch the instruction code.

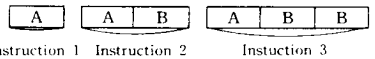
It becomes "High Level" at the ϕ_1 time that the instruction is load. During cycle time that SYNC is a high level, if RDY input is set at the low level, the CPU halts with the state until RDY becomes high level.

The single step execution is enabled by control of RDY.

Vector Address		Signal Names
MSB	LSB	
FFFF	FFFE	$\overline{IRQ}$
FFFD	FFFC	$\overline{RES}$
FFFB	FFFA	$\overline{NMI}$

■ ADDRESSING MODE

The Fig.4 shows a sample of pattern which machine language is stored in the memory. Generally the instruction consists of OP-code and operand (modifies the OP code). The operand gives the information of address. Instruction 1 consists of the OP code, and instruction 2 consists of the 1-byte OP-code and operand. Instruction 3 consists 1 byt OP-code, 2-byte operand. The CPU is informed the length of each instruction by the OP code and is fetch the operand of the number of the required bytes by this information. The OP-code have the information which shows the kind of the operand. The kind of this operand is equivalent to the addressing mode.



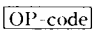
A : OP-code

B : Operand

Fig.4 A SAMPLE OF PATTERN WHICH MACHINE LANGUAGE IS STORED IN MEMORY.

● Accumulator Addressing

This type includes the addressing in the single byte and is equivalent to the execution in the accumulator.

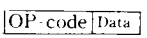


The execution on the accumulator.

Fig.5 ACCUM

● Immediate Addressing

This type is 2-byte instructions having the OP-code and operand. The operand has not information of addressing, but describes the data itself.

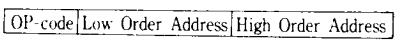


Address is nod needed.

Fig.6 IMMEDIATE

● Absolute Addressing

This type is 3-byte instruction (OP-code is 1-byte and operand is 2-byte). The 2-byte address indicates the low order , the third byte address the high order, all of 64 Kbytes is accessed.



Describes directly the execution address.

Fig.7 ABSOLUTE

● Zero Page Addressing

This type is 2-byte instruction of OP code and operand. The high oder address is automatically set "00". With addressing a low oder address, it is able to code and the short of the execution. It is able to use efficiently the memory space and execute time by using the addressing suitably.

OP-code	Low Order Address

Execution high order address becomes "00"

Fig.8 ZERO PAGE

● Implied Addressing

The instruction code is 1-byte order. Almost all of the instruction control registers which is the internal memory equipment of the CPU. and needs no addressing.

OP code

Without address.

Fig.9 IMPLIED

● Indexed Zero Page Addressing

This is 2-byte instruction of OP-code and operand. It is called as "ZERO PAGE X" or "ZERO PAGE Y" because the execution address is addressed with the indexed register (X or Y).

This is one of the zero page addressing. The high order addressing is set automaticaly "00", and low address is added with the content of the 2 byte. As the carry after the calculation is not added, the execute address does not exceed zero page.

OP code	LOW Order Address

Execution High Order Address : 00

Execution Low Order Address :

Low Order Address	+	Index register (X or Y)

Neglect the carry.

Fig.10 Z PAGE X ; ZPAGE Y

● Indxed Absolute Addressing

This is 3 byte instruction of 1 byte OP code, 2 byte operand. The execute addressing is addressed with the index (X or Y). It is called as "absolute X" or "absolute Y".

This is one of the absolute addressing. Execute address is added with the content of index register. The count of index and content of count are stored in the index register. And it is able to address the base address by the OP-code. It is able to modify the plural areas by using some base address and index, and the code and exceution time can be shortened.

OP-code	Low Order Address	High Order Address

Execution address :

High Order Address	Low Order Address	+	Index Register (X or Y)

Fig.11 ABS, X ; ABS, Y

● Relative Addressing

This is 2-byte instruction of OP-code and operand. It is used only for the jump OP-code, and appoints the jump address, 2-byte oder of OP-code is called as "offset" and is added with the content of offset to the low oder 8 bit of program-counter set to the location of next instruction.

It has the range of -128 to +127-byte. The range of the branch is -128 to 127 byte from the head address of next instruction.

OP-code	Offset

Offset value is 128(80H)~ +127(7FH)

Fig.12 RELATIVE

● Indexed Indirect Addressing

This is 2-byte instruction of OP-code and operand. As execute address is indirectly addressed, it is called as "Indirect X".

The execute address is added with 2 byte of instruction and content of X-register, and the carry is neglected. When the content of calculation is the address of Zero page, the content stored in the address becomes the low order 8-bit of effective address, and the content of next address becomes the high order. The address of stored memory (high and low order) that appoints the effective address must be in the zero page.

The content is stored in the address is low oder 8 bit of efffective address.

OP-code	Low Order Address

Execution Address Low Order :

00	Low Order Address + X register	content of address

Execution Address High Order :

00	Low Order Address + X register + 1	content of the address

Fig.13 (IND, X)

● Indirect Indexed Addressing

This is 2 byte instruction of OP code and

operand. It is called as "Indirect, Y" as it appoints indirectly effective address. The 2-byte of OP-code shows the address of Zero page. The content of Y register is added with the content of memory, and the result becomes the low 8-bit of effective address. Carry is added to the content of next memory in the zero-page and it becomes the high oder 8-bit of effective address.

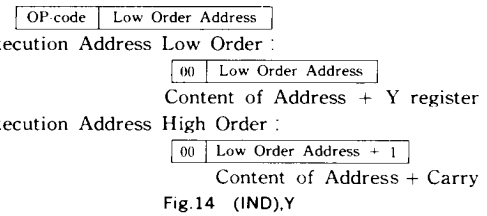


Fig.14 (IND),Y

● Indirect Addressing

This instruction is 2-byte of OP-code and operand. Execution address is address of Zero page.

Contents of this address becomes low oder 8-bit of execution address, and contents of the next address becomes high order 8-bit of execution address. This is the same operation as in the ease of X being zero in "indirect, X".

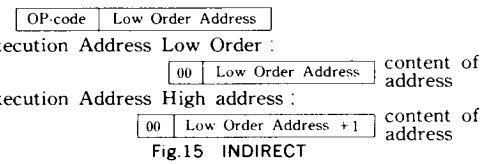
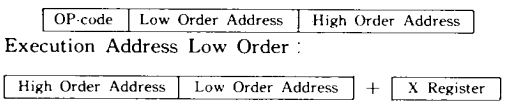


Fig.15 INDIRECT

● Indexed Absolute Indirect Addressing

This is 3-byte instruction (OP-code is 1-byte, operand is 2-byte). The result which adds the content of 2-byte or 3-byte to content of X register becomes the memory address that stores information of execution low order address 8-bit. The content of next address becomes high order 8-bit ,of the execution address.



Execution Address High Order :

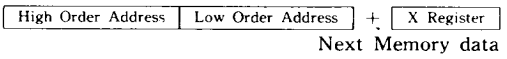


Fig.16 JMP (IND), X

● Absolute Indirect Addressing

The 2-byte of the instruction contains the low order 8-bit of a memory location. The high order 8-bit of that memory location are contained in the 3-byte of the instruction.

The contents of the fully specified memory location are the low order byte of the effective address. The next memory location contains the high order byte of effective address, which is loaded into the 16-bit of the program counter.

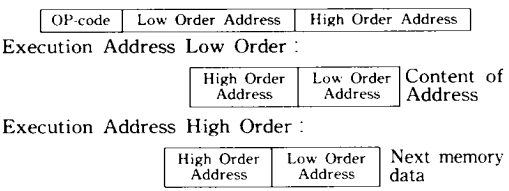


Fig.17 JMP (IND)

● Bit Addressing

In the instruction set (BBR, BBS, RMB, SMB), OP-code corresponds to bit OP-code.

(BBR, BBS) This is 3-byte instruction (OP-code is 1-byte, operand is 2-byte). Execution address is zero page. Low order address is 2-byte of instruction.

3-byte of-instruction is offset content which points the address of branching.

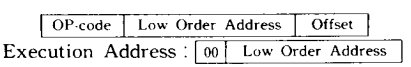


Fig.18 BBR, BBS

(RMB, SMB) This is 2-byte instruction of OP-code, operand. Execution address is zero page, low order address is 2-byte of instruction.

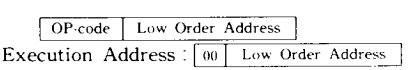
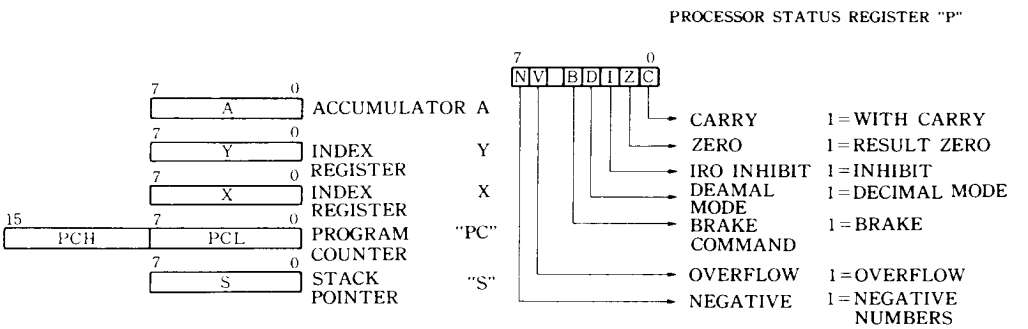


Fig.19 RMB, SMB

■ INTERNAL REGISTER



■ INSTRUCTION SET(Alphabetical order)

(2) ADC Add memory and accumulator with carry
(2) AND Logical AND memory and accumulator
ASL One bit left shift (memory or accumulator)
(1) BBR Branch if bit reset
(1) BBS Branch if bit is set
BCC Branch if carry is cleared
BSC Branch if carry is set
BEQ Branch if result is zero
(2) BIT Test memory bit with accumulator
BMI Branch if result is negative
BNE Branch if result is not zero
BPL Branch if result is positive
(1) BRA Unconditional branch
BRK Forced break
BVC Branch if overflow is cleared
BVS Branch if overflow is set
CLC Clear carry flag
CLD Clear decimal mode
CLI Clear disable interrupt
CLV Clear overflow flag
(2) CMP Compare memory with accumulator
CPX Compare memory with index registerX
CPY Compare memory with index registerY
(2) DEC Decrement memory
DEX Decrement index register X
DEY Decrement index register Y
(2) EOR Exclusive OR memory or accumulator
(2) INC Increment memory
INX Increment index register X
INY Increment index register Y
(2) JMP Jump to new location
JSR Jump to new location, hold return address
(2) LDA Load memory into accumulator
LDX Load memory into index register X
LDY Load memory into index register Y
LSR One bit right shift (memory or accumulater)
NOP No operation
(2) ORA Logical OR memory and accumulator
PHA Push accumulator on stack
PHP Push processer status on stack
(1) PHX Push X register on stack
(1) PHY Push Y register on stack
PLA Pull accumulator from stack
PLP Pull processer status from stack
(1) PLX Pull X register from stack
(1) PLY Pull Y register from stack
(1) RMB Reset memory bit
ROL Rotate left circular of one bit (memory or accumulator)
ROR Rotate right circular of one bit (memory or accumulator)
RTI Return from interrupt
RTS Return from subroutine
(2) SBC Subtract memory and borrow from accumulator
SEC Set carry flag
SED Set decimal
SEI Set disable interrupt status
(1) SMB Set memory bit
(2) STA Store accumulator to memory
STX Store index register X to memory
STY Store index register Y to memory
(1) STZ Zero store
TAX Transfer accumulator to index register X
TAY Transfer accumulator to index registor Y
(1) TRB Test or reset bit
TSB Test or set bit
TSX Transfer stack pointer to accumulator
TXA Transfer index register to accumulator
TXS Transfer index register to stack pointer
TYA Transfer index register to accumulator

Note (1): the instructions are newly designed in 65C02
(2): the instructions are added addressing in 65C02

■ INSTRUCTION SET (Matrix map)

BRK Implied 1 7	– Operation Code – Addressing Mode – Byte : Cycles

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	BRK Implied 1 7	ORA IND,X 2 6			TSB ZP 2 5	ORA ZP 2 3	ASL ZP 2 5	RMB0 ZP 2 5	PHP Implied 1 3	ORA IMM 2 2	ASL Accum 1 2		TSB ABS 3 6	ORA ABS 3 4	ASL ABS 3 6	BBR0 ZP 3 5**
1	BPL Relative 2 2**	ORA IND ,Y 2 5*	ORA IND 2 5		TRB ZP 2 5	ORA ZP,X 2 4	ASL ZP,X 2 6	RMB1 ZP 2 5	CLC Implied 1 2	ORA ABS,Y 3 4*	INC Instruction 1 2		TRB ABS 3 6	ORA ABS,X 3 4*	SAL ABS,X 3 7	BBR1 ZP 3 5**
2	JSR Absolute	AND IND,X 2 6			BIT ZP 2 3	AND ZP 2 3	ROL ZP 2 5	RMB2 ZP 2 5	PLP Implied 1 4	AND IMM 2 2	ROL Instruction 1 2		BIT ABS 3 4	AND ABS 3 4	ROL ABS 3 6	BBR2 ZP 3 5**
3	BMI Relative 2 2**	AND IND ,Y 2 5*	AND IND 2 5		BIT ZP,X 2 4	AND ZP,X 2 4	ROL ZP,X 2 6	RMB3 ZP 2 5	SEC Implied 1 2	AND ABS,Y 3 4*	DEC Instruction 1 2		BIT ABS,X 3 4*	AND ABS,X 3 4*	ROL ABS,X 3 7	BBR3 ZP 3 5**
4	RTI Implied 1 6	EOR IND,X 2 6				EOR ZP 2 3	LSR ZP 2 5	RMB4 ZP 2 5	PHA Implied 1 3	EOR IMM 2 2	LSR Instruction 1 2		JMP ABS 3 3	EOR ABS 3 4	LSR ABS 3 6	BBR4 ZP 3 5**
5	BVC Relative 2 2**	EOR IND ,Y 2 5*	EOR IND 2 5			EOR ZP,X 2 4	LSR ZP,X 2 6	RMB5 ZP 2 5	CLI Implied 1 2	EOR ABS,Y 3 4*	PHY Implied 1 3			EOR ABS,X 3 4*	LSR ABS,X 3 7	BBR5 ZP 3 5**
6	RTS Implied 1 6	ADC IND,X 2 6†			STZ ZP 2 3	ADC ZP 2 3†	ROR ZP 2 5	RMB6 ZP 2 5	PLA Implied 1 4	ADC IMM 2 2†	ROR Instruction 1 2		JMP Indiect 3 5	ADC ABS 3 4†	ROR ABS 3 6	BBR6 ZP 3 5**
7	BVS Relative 2 2**	ADC IND ,Y 2 5*†	ADC IND 2 5†		STZ ZP,X 2 4	ADC ZP,X 2 4†	ROR ZP,X 2 6	RMB7 ZP 2 5	SEI Implied 1 2	ADC ABS,Y 3 4*†	PLY Implied 1 4		JMP IND ,X 3 6	ADC ABS,X 3 4*†	ROR ABS,X 3 7	BBR7 ZP 3 5**
8	BRA Relative 2 3	STA IND,X 2 6			STY ZP 2 3	STA ZP 2 3	STX ZP 2 3	SMB0 ZP 2 5	DEY Implied 1 2	BIT IMM 2 2	TXA Implied 1 2		STY ABS 3 4	STA ABS 3 4	STX ABS 3 4	BBS0 ZP 3 5**
9	BCC Relative 2 2**	STA IND ,Y 2 6	STA IND 2 5		STY ZP,X 2 4	STA ZP,X 2 4	STX ZP,Y 2 4	SMB1 ZP 2 5	TYA Implied 1 2	STA ABS,Y 3 5	TXS Implied 1 2		STZ ABS 3 4	STA ABS,X 3 5	STZ ABS,X 3 5	BBS1 ZP 3 5**
A	LDY IMM 2 2	LDA IND,X 2 6	LDX IMM 2 2		LDY ZP 2 3	LDA ZP 2 3	LDX ZP 2 3	SMB2 ZP 2 5	TAY Implied 1 2	LDA IMM 2 2	TAX Implied 1 2		LDY ABS 3 4	LDA ABS 3 4	LDX ABS 3 4	BBS2 ZP 3 5**
B	BCS Relative 2 2**	LDA IND ,Y 2 5*	LDA IND 2 5		LDY ZP,X 2 4	LDA ZP,X 2 4	LDX ZP,Y 2 4	SMB3 ZP 2 5	CLV Implied 1 2	LDA ABS,Y 3 4*	TSX Implied 1 2		LDY ABS,X 3 4*	LDA ABS,X 3 4*	LDX ABS,Y 3 4*	BBS3 ZP 3 5**
C	CPY IMM 2 2	CMP IND,X 2 6			CPY ZP 2 3	CMP ZP 2 3	DEC ZP 2 5	SMB4 ZP 2 5	INY Implied 1 2	CMP IMM 2 2	DEX Implied 1 2	CPY ABS 3 4	CMP ABS 3 4	DEC ABS 3 6	BBS4 ZP 3 5**	
D	BNE Relative 2 2**	CMP IND ,Y 2 5*	CMP IND 2 5			CMP ZP,X 2 4	DEC ZP,X 2 6	SMB5 ZP 2 5	CLD Implied 1 2	CMP ABS,Y 3 4*	PHX Implied 1 3			CMP ABS,X 3 4*	DEC ABS,X 3 7	BBS5 ZP 3 5**
E	CPX IMM 2 2	SBC IND,X 2 6†			CPX ZP 2 3	SBC ZP 2 3†	INC ZP 2 5	SMB6 ZP 2 5	INX Implied 1 2	SBC IMM 2 2†	NOP Implied 1 2		CPX ABS 3 4	SBC ABS 3 4†	INC ABS 3 6	BBS6 ZP 3 5**
F	BEQ Relative 2 2**	SBC IND ,Y 2 5*†	SBC IND 2 5†			SBC ZP,X 2 4†	INC ZP,X 2 6	SMB7 ZP 2 5	SED Implied 1 2	SBC ABS,Y 3 4*†	PLX Implied 1 4			SBC ABS,X 3 4*†	INC ABS,X 3 7	BBS7 ZP 3 5**
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

† Cycles Add 1 when decimal mode

* Cycles Add 1 when page crossing occurs

** Cycles Add 1 when branch in same page

LSD 0 0 Add 2 when branch in different page

Newly Designed Instruction

■ INSTRUCTION SET

ADDRESSING

Mnemomc	Operation	Immediate			Absolute			Zeropage			Accum			Implied			(IND,X)			(IND),Y		
		op	n	0	op	n	0	op	n	0	op	n	0	op	n	0	op	n	0	op	n	0
ADC	A + MIC→A (1)(5)	69	2	2	6D	4	3	65	3	2							81	6	2	71	5	2
AND	AAM→A A (1)	29	2	2	2D	4	3	25	3	2							21	6	2	31	5	2
ASL	C				0E	6	3	06	5	2	9A	2	1									
BBR(#0-n)	Branch on Mb = 0								5	3												
BBS(#0-n)	Branch on Mb = 1								5	3												
BCC	Branch on C = 0 (2)																					
BCS	Branch on C = 1 (2)																					
BEQ	Branch on Z = 1 (2)																					
BIT	AAM (b)	89	2	2	2C	4	3	24	3	2												
BMI	Branch on N = 1 (2)																					
BNE	Branch on Z = 0 (2)																					
BPL	Branch on N = 0 (2)																					
BRA	Branch Alloys (2)																					
BRK	Broak																					
BVC	Branch on V = 0 (2)													00	7	1		6				
BVS	Branch on V = 1 (2)																					
CLC	0 - C																					
CLD	0 - D													18	2	1						
CLI	0 - I													D8	2	1						
CLV	0 V													58	2	1						
CMP	A - M (1)	C9	2	2	CD	4	3	C5	3	2				B8	2	1						
CPX	X - M	E0	2	2	EC	4	3	E4	3	2							C1	6	2	D1	5	2
CPY	Y - M	C0	2	2	CC	4	3	C4	3	2												
DEC	M - 1 →M				CE	6	3	C6	5	2	3A	2	1									
DEX	X - 1 →X													CA	2	1						
DEY	Y - 1→Y													88	2	1						
EOR	AYM •A (1)	49	2	2	4D	4	3	45	3	2							41	6	2	51	5	
INC	M + 1 →M				EE	6	3	E6	5	2	1A	2	1									
INX	X + 1 →X													E8	2	1						
INY	Y + 1 →Y													C8	2	1						
JMP	Jumo to New Loc				4C	3	3										7C	6	3			
JSR	Jump Sub				20	6	3															
LDA	M - A (1)	A9	2	2	AD	4	3	A5	3	2							A1	6	2	B1	5	2
LDX	M - X (1)	A2	2	2	AE	4	3	A6	3	2												
LDY	M - Y (1)	A0	2	2	AC	4	3	A4	3	2												
LSR	0 - - C				4E	6	3	A6	5	2	4A	2	1									
NOP	No Operation													EA	2	1						
ORA	AVM - A (1)	09	2	2	0D	4	3	05	3	2							01	6	2	11		2
PHA	A - Ms S - 1→S													48	3	1						
PHP	P - Ms S - 1→S													08	3	1						
PHX	X - Ms S - 1→S													DA	3	1						
PHY	Y Ms S 1→S													5A	3	1						
PLA	S + 1 →S Ms→A													68	4	1						
PLP	S + 1 →S Ms→P													28	4	1						
PLX	S + 1 →S Ms→X													FA	4	1						
PLY	S + 1 Ms→Y													7A	4	1						
RMB(#0-n)	0→Mb (1)								5	2												
ROL					2E	6	3	26	5	2	2A	2	1									
ROR					6E	6	3	66	5	2	6A	2	1									
RTI	Rtrn Int Isoo Frg 11																					
RTS	Rtrn Sub Isee.Frg 21													40	6	1						
SBC	A - M - C →A (1)(5)	E9	2	2	ED	4	4	E5	3	2				60	6	1	E1	6	2	F1	5	2
SEC	I→C																					
SED	I→D													38	2	1						
SEI	I→I													F8	2	1						
SMB(#0-n)	I→Mg (4)								5	2				78	2	1						
STA	A→M				8D	4	3	85	3	2							81	6	2	91	6	2
STX	X→M				8E	4	3	86	3	2												
STY	Y→M				8C	4	3	84	3	2												
STZ	0→M				9C	4	3	64	3	2												
TAX	A →X																					
TAY	A →Y													AA	2	1						
TRB	A ^M→M				1C	6	3	14	5	2				A8	2	1						
TSB	AVM→M				0C	6	3	04	5	2												
TSX	S →X																					
TXA	X→A													BA	2	1						
TXS	X→S													8A	2	1						
TYA	Y •A													9A	2	1						
														98	2	1						

(Symbol Description)

X : Index X
Y : Index Y
A : Accumulator
M : Designated Memory with Effective Address
Ms : Designated Memory with Stack pointer
Mb : Zero page Memory Bit
M_7 : Memory Bit 7
M_6 : Memory Bit 6
+ : Add
- : Subtract
^ : Logical AND
. : Logical OR
≠ : Exclusive OR
n : Machine Cycles
: Bytes

RICOH

MODE

Zpage. X			ABS.X			ABS.Y			Relative			Indirect			Zpage. Y			Bit Addressing (OP by Bit 0)								Processor Status Codes							
op	n	0	op	n	0	op	n	0	op	n	0	op	n	0	op	n	0	0	1	2	3	4	5	6	7	7 N	6 V	5 ·	4 B	3 D	2 I	1 Z	0 C
75	4	2	7D	4	3	79	4	3				72	5	2																			
35	4	2	3D	4	3	39	4	3				32	5	2													N					Z	C
16	6	2	1E	7	3																						N					Z	C
																		0F	1F	2F	3F	4F	5F	6F	7F							
																		8F	9F	AF	BF	CF	DF	EF	FF							
									90	2	2																					
									B0	2	2																					
									F0	2	2																					
34	4	2	3C	4	3																						M7	M6				Z	
									30	2	2																					
									D0	2	2																					
									10	2	2																					
									8ø	3	2																					
									50	2	2																					
									70	2	2																					0
																														0		
																												0				
D5	4	2	DD	4	3	D9	4	3				D2	5	2													N	V				Z	C
																											N					Z	C
D6	6	2	DE	7	3																						N					Z	C
																											N					Z	
55	4	2	5D	4	3	59	4	3				52	5	2													N					Z	
F6	6	2	FE	7	3																						N					Z	
																											N					Z	
												6C	5	3													N					Z	
B5	4	2	BD	4	3	B9	4	3				B2	5	2													N					Z	
						BE	4	3							B6	4	2										N					Z	
B4	4	2	BC	4	3																						N					Z	
56	6	2	5E	7	3																						O					Z	C
																											N					Z	
15	4	2	1D	4	3	19	4	3				12	5	2													N					Z	
																											N					Z	
																														(Restored)			
																											N					Z	
																											N					Z	
																		07	17	27	37	47	57	67	77							
36	6	2	3E	7	3																						N					Z	C
76	6	2	7E	7	3																						N					Z	C
																														(Restored)			
F5	4	2	FD	4	3																												
						F9	4	3				F2	5	2													N	V				Z	(3)
																																1
																													1			
95	4	2	9D	3	3													87	97	A7	B7	C7	D7	E7	F7							
						99	5	3				92	5	2	96	4	2															
94	4	2																														
74	4	2	9E	3	3																											
																											N					Z	
																											N					Z	
																											N					Z	

NOTE (1) Add 1 to "N" when machine cycles or zero page crossing
(2) Add 1 to "N" when branch in same page
Add 2 to "N" when branch in different page
(3) Not carry = borrow
(4) Effect when zero page address
(5) Add 1 when decimal mode
(6) When the BIT (IMM mode), not related to M_7 or M_6 bit (N, V flag) of register P

■ Instruction Description (Alphabetical Order)

Descrption of symbol using in list

A	Accumulator	−	Subtract
X,Y	Index Register	∀	Exclusive OR
M	Memory	→	Transfer
P	Processor Status Register	←	Transfer
S	Stack Pointer	∨	Logical OR
Ms	Stack Memory	PCH	Program Counter High
Mb	Memory Bit	PCL	Program Counter Low
+	Add		
∧	Logical AND		

A D C Add with carry of memory and accumulator.
Operation : A+M+C → A "P" Register : N,V,Z,C

A N D Logical AND of memory and accumulator. The result is stored in accumulator.
Operation : A∧M → A "P" Register : N,Z

A S L One bit left shift. LSB is placed "0". Contents of MSB is placed C.
Operation : C ← 7 6 5 4 3 2 1 0 ← "0" "P" Register : N,Z,C

B B R If specific bit of zero page is a reset state, branch relatively.

OP-Code	Low Order Address	Offset

3-byte instruction

If the specific bit (a bit is decided on the instruction code) of effective address 00 Low Order Address is a reset state, relative branch by the Offset value on the basis of lead address of next instruction.
Operation : branch when $M_b = 0$ "P" Register : not affected

B B S If specific bit of zero page is a set state, branch relatively.

OP-code	Low Order Address	Offset

3-byte instruction

If the specific bit (a bit is decided on the instruction code) of effective address 00 Low Order Address is a set state, relative branch by the Offset value to base with lead address of next instruction.
Operation : branch when $M_b = 1$ "P" Register : Not affected

B C C Branch if the carry is reset.
Operation : branch when C = 0 "P" Register : Not affected

B C S Branch if the carry is set.
Operation : branch when C = 1 "P" Register : Not affected

B E Q Branch if the zero flag is set.
Operation : branch when Z = 1 "P" Register : Not affected

B I T Test the memory bit by the accumulator.
Operation : $A \wedge M, M_7 \rightarrow N, M_6 \rightarrow V$
The bit 6 and bit 7 of the memory are transferred to "P" Register.
If the result of A∧M is zero, Z = 1 "P" Register : N, V, Z
$(M_7)(M_6)$

B M I Branch if result is negative.
Operation : branch when N = 1 "P" Register : Not affected

BNE Branch if result is not zero.
Operation : branch when Z = 0 "P" Register : Not affected

BPL Branch if result is positive.
Operation : branch when N ≒ 0 "P" Register : Not affected

BRA Unconditional branch. "P" Register : Not affected

BRK Forced break
Operation: Execute the interrupt. In this instruction, a lead address (2-byte) of next instruction is stored in the stack. At the same time, it is stored into contents of "P" Register. Program-counter (FFFE) → PCL, (FFFF) → PCH, and Execution of program is same vector address with IRQ. The difference from the IRQ interrupt is that in the BKK operation, the B flag of "P" register is set "1" and can't mask by the I flag.
"P" Register : $\frac{B}{1}$

BVC Branch if the overflow flag is reset.
Operation : branch when V = 0 "P" Register : Not affected

BVS Branch if the overflow flag is set.
Operation : branch when V = 1 "P" Register : Not affected

CLC Clear the carry flag (C)
Operation : 0 → C "P" Register : $\frac{C}{0}$

CLD Clear the decimal mode.
Operation : 0 → C "P" Register : $\frac{D}{0}$

CLI Clear the interrupt disenable flag (I).
Operation : 0 → I "P" Register : $\frac{V}{0}$

CLV Clear overflow flag.
Operation : 0 → V "P" Register : $\frac{V}{0}$

CMP Compare memory with accumulator.
Óperation : A-M
The result is not stored. If it is nagative, N flag is set 1. If it is positive, C flag is set 1.
And if it is zero, Z and C flags are respectively 1. "P" Register : N, Z, C

CPX Compare memory with the index register X
Operation : Y-M
Flag condition of "P" Register is the same as CMP.
"P" Register : N, Z, C

CPY Compare memory with the index register Y.
Operation : Y-M
Flag condition of "P" Register is the same as CMP.
"P" Register : N, Z, C

DEC Decrement the contents of memory.
Operation : M − 1 → M "P" Register : N, Z

DEX Decrement the contents of index register X.
Operation : X − 1 → X "P" Register : N, Z

DEY Decrement the contents of index register Y.
Operation : Y − 1 → Y "P" Register : N, Z

EOR Execute the exclusive OR of memory and accumulator.

Operation : $A \forall M \rightarrow A$ "P" Register : N, Z

INC Increment the contents of memory.
Operation : $M+1 \rightarrow M$ "P" Register : N, Z

INX Increment the contents of index register X.
Operation : $X+1 \rightarrow X$ "P" Register : N, Z

INY Increment the contents of index register Y.
Operation : $Y+1 \rightarrow Y$ "P" Register : N, Z

JMP Execution of program jumps to designation address.

Operation : | OP-Code | Operand | Operand | "P"Register : Not affected

The designation address by operands with 2-bytes is placed in PCL and PCH.

JSR The execution of program jumps to designation address.
Operation : When jump to designation address, return address (lead address of next instruction) is stored into stack. The return is executed by RTS.

| OP-Code | Operand | Operand | "P"Register : Not affected

The disignation address by operands with 2 bytes is stored into the PCL and PCH.
Lead address of next instruction(2-byte)——M_S, $S-1 \rightarrow S$
└——M_S, $S-1 \rightarrow S$

LDA Load the contents of memory to the accumulator.
Operation : $M \rightarrow A$ "P" Register : N, Z

LDX Load the contents of memory to index register X.
Operation : $M \rightarrow X$ "P" Register : N, Z

LDY Load the contents of memory to index register Y.
Operation : $M \rightarrow Y$ "P" Register : N, Z

LSR One bit right shift. MSB (7bit) is placed to 0, LSB (0 bit) is loaded the C.
Operation : $0 \rightarrow$ [7][6][5][4][3][2][1][0] $\rightarrow C$ "P" Register : $\overset{N}{0}$, Z, C

NOP No-operation
Operation : No operation "P" Register : Not affected

ORA Logical OR of memory and accumulator. The result is stored into the accumulator.
Operation : $A \vee M \rightarrow A$ "P" Register : N, Z

PHA Store the contents of the accumulator into the memory stack.
Operation : $A \rightarrow Ms$, $S-1 \rightarrow S$ "P" Register : Not affected

PHP Store the contents of the register P into the stack.
Operation : $P \rightarrow Ms$, $S-1 \rightarrow S$ "P" Register : Not affected

PHX Store the contents of the index register X into the stack.
Operation : $X \rightarrow Ms$, $S-1 \rightarrow S$ "P" Register : Not affected

PHY Store the contents of the index register Y into the stack.
Operation : $Y \rightarrow Ms$, $S-1 \rightarrow S$ "P" Register : Not affected

PLA Pull accumulator from stack.
Operation : $Ms \rightarrow A$, $S+1 \rightarrow S$. "P" Register : N, Z

PLP Pull processer status from stack.
Operation : $Ms \rightarrow P$, $S+1 \rightarrow S$ "P" Register : Restore

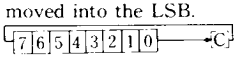
PLX Pull X register from stack.
Operation : $Ms \rightarrow X$, $S+1 \rightarrow S$ "P" Register : N, Z

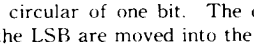
PLY Pull Y register from stack.
Operation : $Ms \rightarrow Y$, $S+1 \rightarrow S$ "P" Register : N, Z

RMB Reset the specific bit in the zero page address.

OP-Code	Low Order Bit	2-byte instruction

The specific bit (a bit is decided by the instruction code) of execution address
00 | Low Order Address is reset.
Operation : $0 \rightarrow M_b$ "P" Register : Not affected

ROL Rotate left circular of one bit. The contents of the MSB are moved into the C, the contents of the C are moved into the LSB.
Operation :  "P" Register : N, Z, C

ROR Rotate right circular of one bit. The contents of the C are moved into the MSB, the contents of the LSB are moved into the C.
Operation :  "P" Register : N, Z, C

RTI Return from interrupt. The return address in stack is loaded into the program counter, and it becomes the lead address of a next instruction of the interrupt.
Operation : $M_S \rightarrow P$, $S+1 \rightarrow S$ "P" Register : Restore
$M_S \rightarrow PCL$, $S+1 \rightarrow S$
$M_S \rightarrow PCH$, $S+1 \rightarrow S$

RTS Return from subroutine.
The return address in stack is loaded into the program counter. It becomes the lead address of a next instruction of the JSR.
Operation : $M_S \rightarrow PCL$, $S+1 \rightarrow S$ "P" Register : Not affected
$M_S \rightarrow PCH$, $S+1 \rightarrow S$

SBC Subtract memory and borrow from accumulator, and the result is stored into the accumulator.
Operation : $A - M - \overline{C} \rightarrow A$ "P" Register : N, V, Z, C
$\overline{C}$: borrow

SEC Set carry flag.
Operation : $1 \rightarrow C$ "P" Register : $\overset{C}{1}$

SED Set decimal flag.
Operation : $1 \rightarrow D$ "P" Register : $\overset{D}{1}$

SEI Set disable interrupt status.
Operation : $1 \rightarrow I$ "P" Register : $\overset{I}{1}$

SMB Set the specific bit of zero page address.

OP-Code	Low Order Bit	2-byte instruction

It sets the specific bit (a bit is decided on the instruction code) of effective address
00 | Low Order Address
Operation : $1 \rightarrow M_b$ "P" Register : Not affected

STA Store the contents of the accumulator into the memory.
Operation : $A \rightarrow M$ "P" Register : Not affected

STX Store the contents of the index register X into the memory.
Operation : X → M "P" Register : Not affected

STY Store the contents of the index register Y into the memory.
Operation : Y → M "P" Register : Not affected

STZ Clear the contents of memory.
Operation : 0 → M "P" Register : Not affected

TAX Transfer the contents of the accumulator to the index register X.
Operation : A → M "P" Register : N, Z

TAY Transfer the contents of the accumulator to the index register Y.
Operation : A → Y "P" Register : N, Z

TRB Reset the contents of memory by accumulator, and test at the same time.
Operation : A∧M → M
If the result is zero, Z flag=1 "P" Register : Z

TSB Set the contents of memory by accumulator, and test at the same time.
Operation : A∨M → M
If the result is zero, Z flag = 1 "P" Register : Z

TSX Transfer stack pointer to the index register X.
Operation : S → X "P" Register : N, Z

TXA Transfer the contents of the index register X to the accumulator.
Operation : X → A "P" Register : N, Z

TXS Transfer the contents of the index register X to stack pointer.
Operation : X → S "P" Register : Not affected

TYA Transfer the contents of the index register Y to the accumulator.
Operation : Y → A "P" Register : N, Z

■ 40-PIN DUAL-IN-LINE PACKAGE (UNIT : mm)