T-49-17-06

RICOH

EK-011-9007

CMOS 8bit MPU

RP65C02 G/G-06

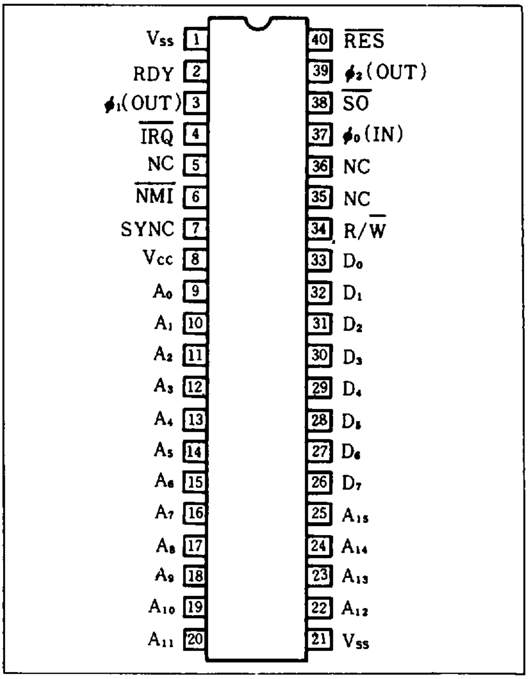
■ GENERAL DESCRIPTION

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

■ FEATURES

- 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
- 8-bit bi-directional data bus, parallel processing
- Clock RP65C02G .... 4 MHz
  RP65C02G-06 .... 6 MHz
- Computable decimal and binary
- Bus compatible with M6800
- Pin compatible with ROCKWELL R65C02
- Single power supply 5V operation
- Low power dissipation

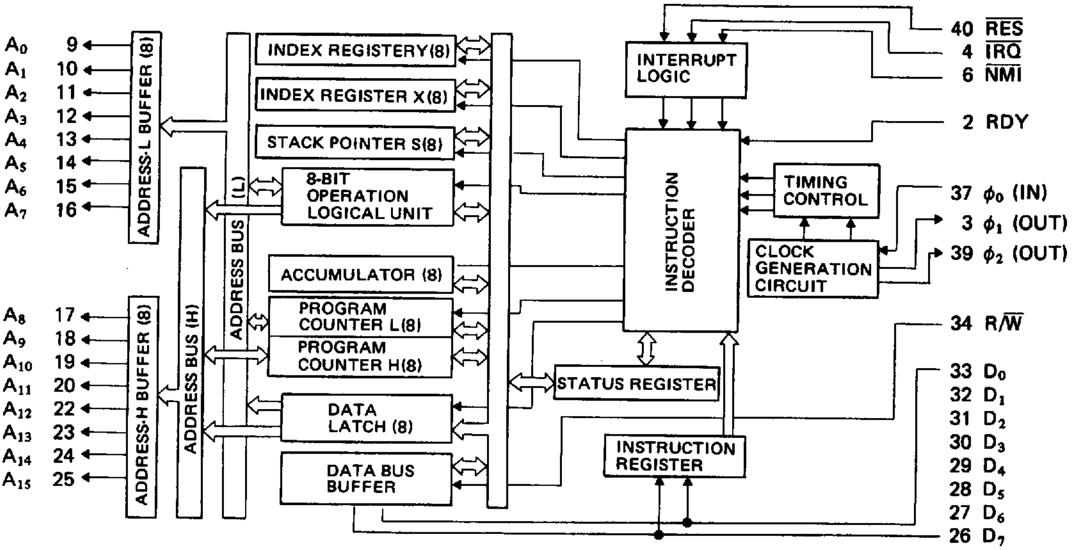
■ PIN CONFIGURATION (TOP VIEW)



■ PIN DESCRIPTION

PIN NAME	FUNCTION	PIN NAME	FUNCTION
Vss	Internal Logic Ground	Vcc	+5V Power Supply
RDY	Ready	$A_0 \sim A_{15}$	Address Bus
ϕ_1 (OUT)	Clock 1 Out	$\overline{RES}$	Reset
$\overline{IRQ}$	Interrupt Request	ϕ_2 (OUT)	Clock 2 Out
NC	No Connection	$\overline{SO}$	Set Overflow
$\overline{NMI}$	Non-Maskable Interrupt	ϕ_0 (IN)	Clock 0 In
SYNC	Synchronize	R/$\overline{W}$	Read/Write
		$D_0 \sim D_7$	Data Bus

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Symbol	Parameters	Limits	Unit
V_{CC}	Supply Voltage	−0.3 ~ +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	°C
T_{stg}	Storage Temperature	−40 ~ +125	°C

■ ELECTRICAL CHARACTERISTICS

● DC ELECTRICAL CHARACTERISTICS (Vcc = 5.0 ± 5%, Ta = 0 ~ +70°C)

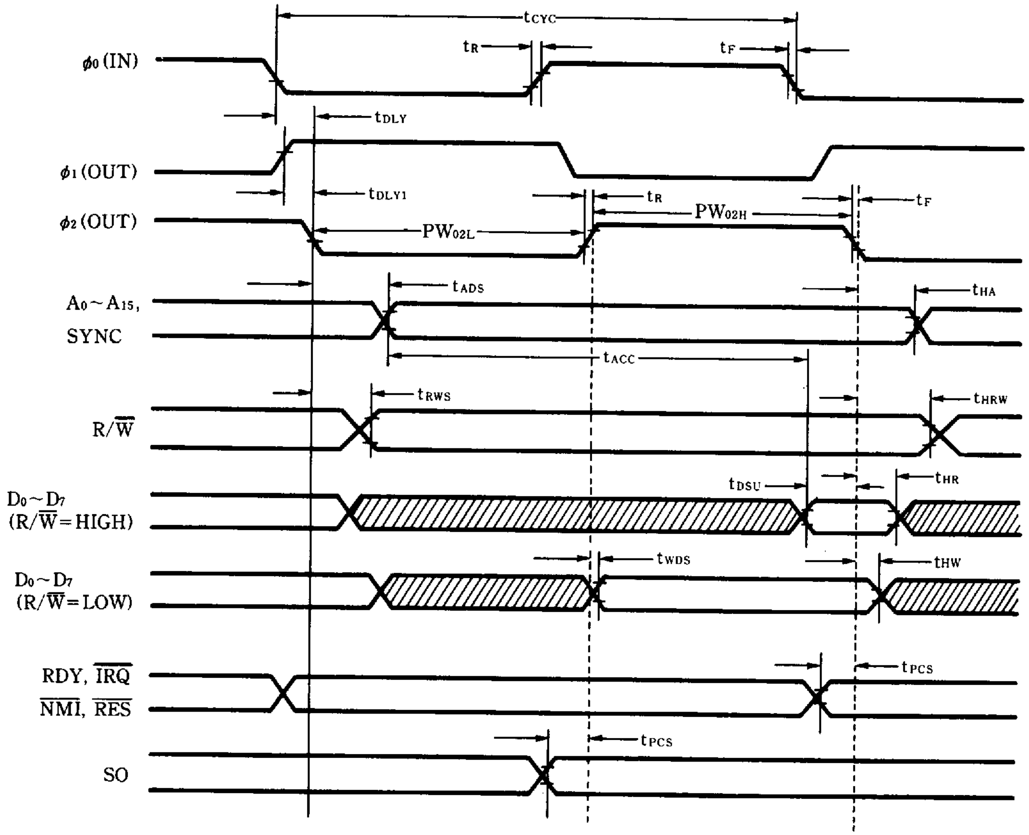
Symbol	Parameters	Measuring Conditions	Specified Value			Unit
			Min	Typ	Max	
V_{IH}	Input High Voltage ϕ_0 (IN), $\overline{NMI}$ $\overline{RES}$, RDY, $\overline{IRQ}$, $\overline{SO}$, $D_0 \sim D_7$		 0.7·Vcc 2.0		 Vcc+0.3 Vcc+0.3	 V V
V_{IL}	Input Low Voltage ϕ_0 (IN), $\overline{NMI}$ $\overline{RES}$, RDY, $\overline{IRQ}$, $\overline{SO}$; $D_0 \sim D_7$		 −0.3 −0.3		 0.2 0.8	 V V
I_{LI}	Input Leak Current $\overline{RES}$, $\overline{NMI}$, RDY, $\overline{IRQ}$, SO (Internal Pull-Up) ϕ_0 (IN)	Vcc = 5.25V V_{IN} = 0 ~ 5.25V	 −100 −10		 10 10	 μA μA
I_{LO}	Output Floating Leakage Current $D_0 \sim D_7$	V_{IN} = 0 ~ 5.25V	 −10		 10	 μA
V_{OH}	Output High Voltage ϕ_1 (OUT), ϕ_2 (OUT), SYNC, $D_0 \sim D_7$, $A_0 \sim A_{15}$, R/$\overline{W}$	I_{LOAD} = −100μA Vcc = 4.75V	2.4			V
V_{OL}	Output Low Voltage ϕ_1 (OUT), ϕ_2 (OUT), SYNC, $D_0 \sim D_7$, $A_0 \sim A_{15}$, R/$\overline{W}$	I_{LOAD} = 1.6 mA Vcc = 4.75V			0.4	V
Ioc	Power Dissipation (No-Load) Stand-By Active Low-Power	 ϕ_0 * = Vcc or 0, V_{IN} = Vcc RDY = 0			 28 5 2	 μA mA/MHz mA/MHz
C	Input Capacitance Logic, ϕ_0 (IN) ϕ_1 (OUT), ϕ_2 (OUT), SYNC, $D_0 \sim D_7$, $A_0 \sim A_{15}$, R/$\overline{W}$	V_{IN} = 0, f = 1 MHz			 10 20	 pF pF

● AC ELECTRICAL CHARACTERISTICS (Vcc = 5V ± 10%, Ta = 0 ~ 70°C)

Symbol	Parameters	65C02G (4 MHz)		65C02G-06 (6 MHz)		Unit
		Min	Max	Min	Max	
t_{CYC}	Cycle Time	250	DC	166	DC	ns
PW_{02L}	ϕ_2 (OUT) "Low" Clock Pulse Width	100	DC	75	DC	ns
PW_{02H}	ϕ_2 (OUT)"High" Clock Pulse Width	110	DC	80	DC	ns
t_R, t_F	Clock Rising Time, Clock Falling Time		10		10	ns
t_{ADS}	Address Delay Time		80		70	ns
t_{HA}	Address Hold Time	15		15		ns
t_{RWS}	R/$\overline{W}$ Delay Time		80		80	ns
t_{HRW}	R/$\overline{W}$ Hold Time	20		15		ns
t_{DSU}	Read Data Set-Up Time	30		20		ns
t_{HR}	Read Data Hold Time	10		10		ns
t_{WDS}	Write Data Delay Time		60		50	ns
t_{HW}	Write Data Hold Time	30		20		ns
t_{ACC}	Read Access Time	140		76		ns
t_{PCS}	Processor Control Set-Up Time (RDY, $\overline{SO}$, $\overline{IRQ}$, $\overline{NMI}$, $\overline{RES}$)	60		40		ns
t_{DLY}	Delay Time ϕ_0 (IN) to ϕ_2 (OUT)		50		40	ns
t_{DLY1}	Delay Time ϕ_1 (OUT) to ϕ_2 (OUT)	-20	20	-20	20	ns

(Output Load Includes Scope and Jig Capacitance is 130pF)

■ TIMING CHART



■ PIN DESCRIPTION

● Clock Input (ϕ_0 (IN))

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 4 and 8 MHz. And when ϕ_0(IN) stops at highlevel or lowlevel, the CPU becomes the stand-by mode.

● Clock Output (ϕ_1 (OUT), ϕ_2 (OUT))

The two signal ϕ_1 (OUT) or ϕ_2 (OUT) for the system-clock output. These are provided with each device for control bus synchronous signal. Phase relation ϕ_1 (OUT), ϕ_2 (OUT) are as shown in Fig. 1

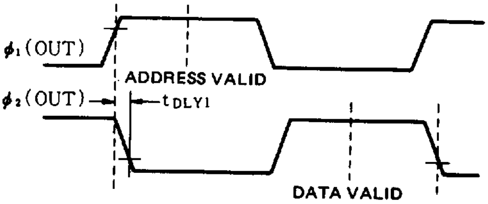


Fig. 1 Phase Relation by System-Clock ϕ_1 (OUT), ϕ_2 (OUT)

- Address Bus (A_0 ~ A_{15})

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

- Data Bus (D_0 ~ D_7)

D_0 ~ D_7 constitute the 8-bit bidirectional data bus, input or output.

- Bus Direction Indicative Signal (R/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.

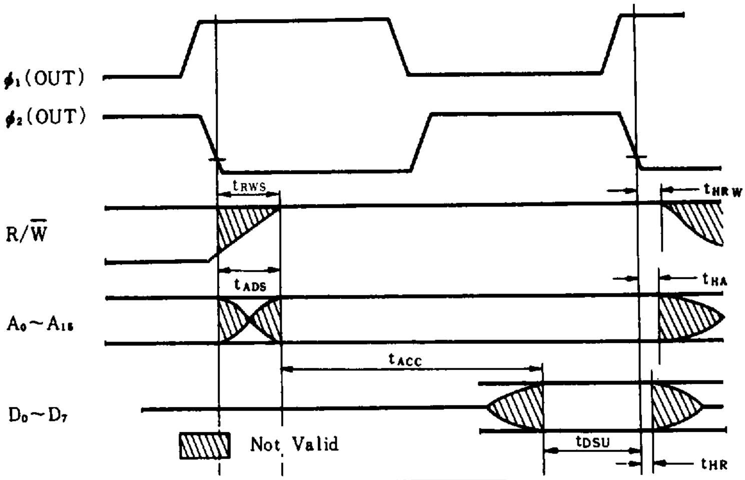


Fig. 2 Read Mode Timing

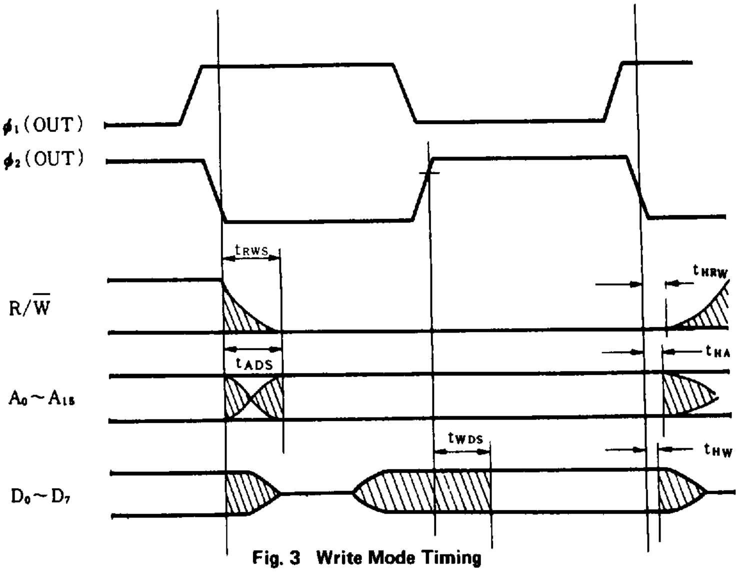


Fig. 3 Write Mode Timing

- Ready Signal (RDY)

This RDY input allows to single-step operation or stop on all cycles. When the falling edge of ϕ_2(OUT) is detected, the CPU stop. When the CPU stop, the address line fetch the current address and when the operation is WRITE mode, the data bus fetch the current data. When the RDY input is low, the CPU becomes low-power mode.

- System Reset $(\overline{\text{RES}})$

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 5 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.

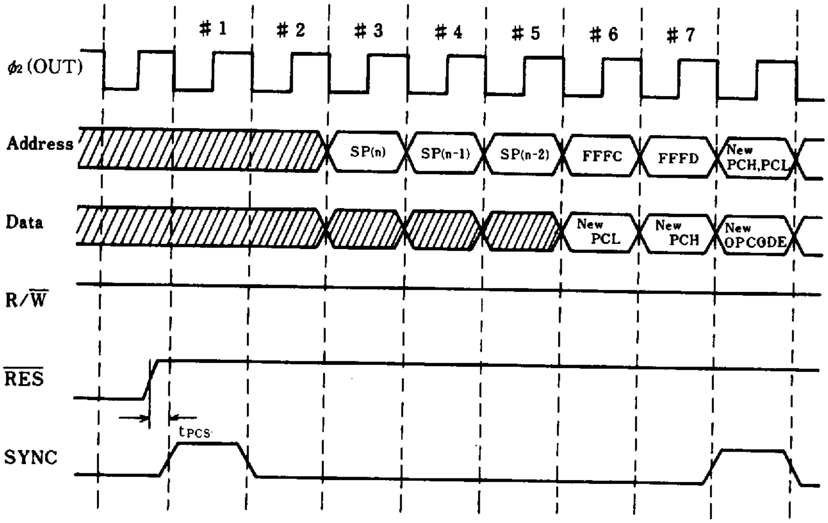


Fig. 4 Reset Mode Timing

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

$\overline{\text{IRQ}}$ (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 program-counter. The program control is changed a memory vector which is stored these location.

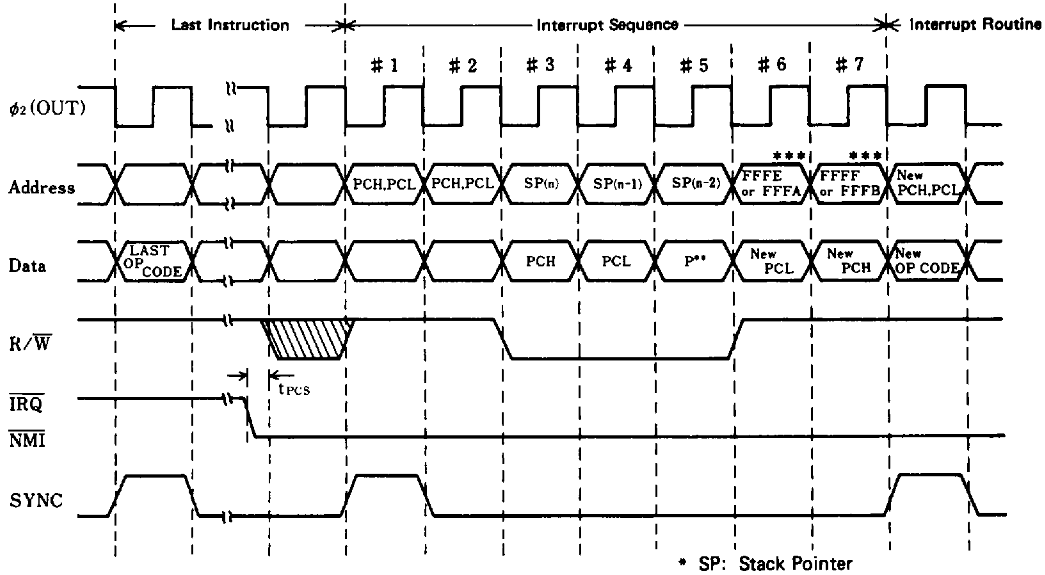
To accept an interrupt, RDY signal should be high level. These are just same with all interruptions. When it is used to the wired OR with this pin, it must use a pullup resistor.

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

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

$\overline{IRQ}$ 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.



* SP: Stack Pointer

** P: Processor Status Register

*** Vector Address of $\overline{IRQ}$ FFFE/FFFF
Vector Address of $\overline{NMI}$ FFFA/FFFB

Fig. 5 Interrupt Operation Timing

- 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 of ϕ_2(OUT), the input must be synchronized outside.

- Instruction Fetch Cycle Synchronous Signal (SYNC)

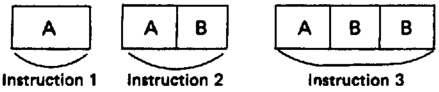
This output signal indicates the cycle that the CPU fetch the instruction code. It becomes "High Level" at the instruction is load and during the cycle time that SYNC is high level. During cycle time that SYNC is high level, if RDY input is set at the low level, the CPU halts with the state until RDY 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. 6 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-byte 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. 6 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.

OP-code

↑

The execution on the accumulator.

Fig. 7 Accumlator

• 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.

OP-code	Data

↑

Address is not needed.

Fig. 8 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.

OP-code	Low Order Address	High Order Address

Describes directly the execution address.

Fig. 9 Absolute

• Zero Page Addressing

This type is 2-byte instruction of OP-code and operand. The high order address is automatically set "00". With addressing a low order 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. 10 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. 11 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 automatically "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. 12 Z Page X; Z Page Y

• Indexed 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 execution 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. 13 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. 14 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 order 8-bit of effective 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. 15 (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 order 8-bit of effective address.

OP-code	Low Order Address

Execution Address Low Order:

00	Low Order Address

Content of Address + Y register

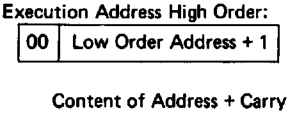


Fig. 16 (IND), Y

• 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 order 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".

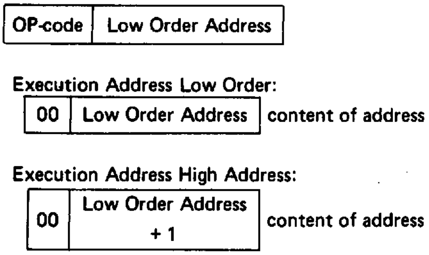


Fig. 17 Indirect

• 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.

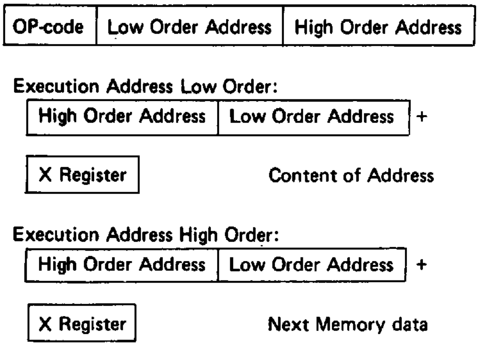


Fig. 18 JMP (IND), X

• 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.

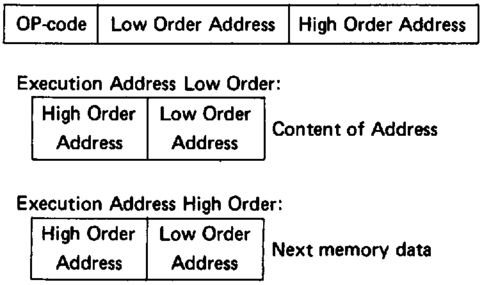


Fig. 19 JMP (IND)

• 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.

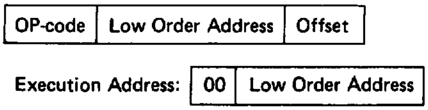


Fig. 20 BBR, BBS

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

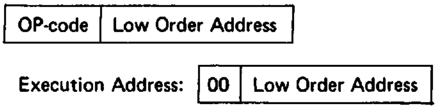
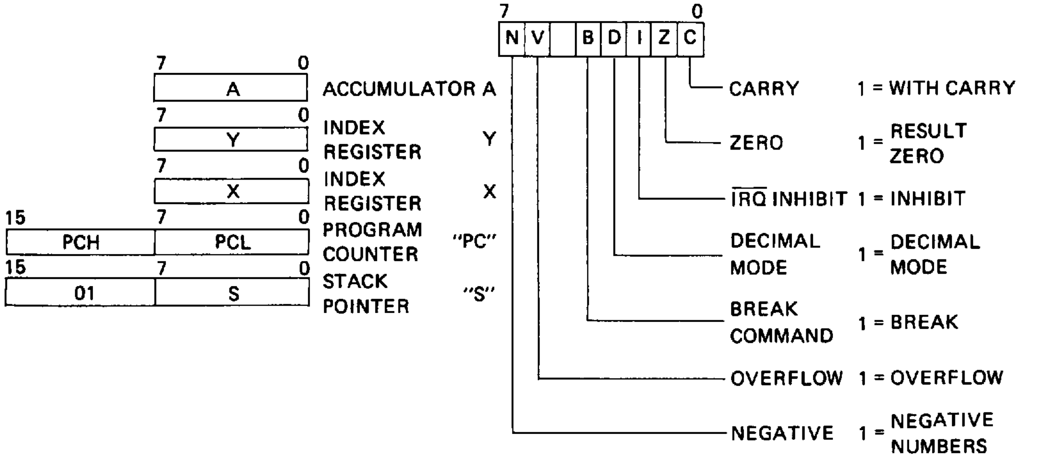


Fig. 21 RMB, SMB

■ INTERNAL REGISTER

PROCESSOR STATUS REGISTER "P"



■ 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 register X
	CPY	Compare memory with index register Y
(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 accumulator)
	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 ar.d borrow from accumulator
	SEC	Set carry flag
	SED	Set decimal
	SEI	Set disable interrupt ststus
(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 register 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	– Operation Code
Impied	– Addressing Mode
1 7	– Bytes: Cycles

MSD \ LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRK IMP 1 7	ORA (IND.X) 2 6			TSB ZP 2 5	ORA ZP 2 3	ASL ZP 2 5	RMB0 ZP 2 5	PHP IMP 1 3	ORA IMM 2 2	ASL ACC 1 2		TSB ABS 3 6	ORA ABS 3 4	ASL ABS 3 6	BBS0 ZP 3 5**	0
1	BPL REL 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 IMP 1 2	ORA ABS,Y 3 4*	INC ACC 1 2		TRB ABS 3 6	ORA ABS,X 3 4*	SAL ABS,X 3 6*	BBR1 ZP 3 5**	1
2	JSR ABS 3 6	AND (IND,X) 2 6			BIT ZP 2 3	AND ZP 2 3	ROL ZP 2 5	RMB2 ZP 2 5	PLP IMP 1 4	AND IMM 2 2	ROL ACC 1 2		BIT ABS 3 4	AND ABS 3 4	ROL ABS 3 6	BBR2 ZP 3 5**	2
3	BMI REL 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 IMP 1 2	AND ABS,Y 3 4*	DEC ACC 1 2		BIT ABS,X 3 4*	AND ABS,X 3 4*	ROL ABS,X 3 6*	BBR3 ZP 3 5**	3
4	RTI IMP 1 6	EOR (IND,X) 2 6				EOR ZP 2 3	LSR ZP 2 5	RMB4 ZP 2 5	PHA IMP 1 3	EOR IMM 2 2	LSR ACC 1 2		JMP ABS 3 3	EOR ABS 3 4	LSR ABS 3 6	BBR4 ZP 3 5**	4
5	BVC REL 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 IMP 1 2	EOR ABS,Y 3 4*	PHY IMP 1 3			EOR ABS,X 3 4*	LSR ABS,X 3 6*	BBR5 ZP 3 5**	5
6	RTS IMP 1 6	ADC (IND,X) 2 6†			STZ ZP 2 3	ADC ZP 2 3†	ROR ZP 2 5	RMB6 ZP 2 5	PLA IMP 1 4	ADC IMM 2 2†	ROR ACC 1 2		JMP IND 3 6	ADC ABS 3 4†	ROR ABS 3 6	BBR6 ZP 3 5**	6
7	BVS REL 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 IMP 1 2	ADC ABS,Y 3 4*†	PLY IMP 1 4		JMP (IND),X 3 6	ADC ABS,X 3 4*†	ROR ABS,X 3 6*	BBR7 ZP 3 5**	7
8	BRA REL 2 3	STA (IND,X) 2 6			STY ZP 2 3	STA ZP 2 3	STX ZP 2 3	SMB0 ZP 2 5	DEY IMP 1 2	BIT IMM 2 2	TXA IMP 1 2		STY ABS 3 4	STA ABS 3 4	STX ABS 3 4	BBR0 ZP 3 5**	8
9	BCC REL 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 IMP 1 2	STA ABS,Y 3 5	TXS IMP 1 2		STZ ABS 3 4	STA ABS,X 3 5	STZ ABS,X 3 5	BBS1 ZP 3 5**	9
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 IMP 1 2	LDA IMM 2 2	TAX IMP 1 2		LDY ABS 3 4	LDA ABS 3 4	LDX ABS 3 4	BBS2 ZP 3 5**	A
B	BCS REL 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 IMP 1 2	LDA ABS,Y 3 4*	TSX IMP 1 2		LDY ABS,X 3 4*	LDA ABS,X 3 4*	LDX ABS,Y 3 4*	BBS3 ZP 3 5**	B
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 IMP 1 2	CMP IMM 2 2	DEX IMP 1 2		CPY ABS 3 4	CMP ABS 3 4	DEC ABS 3 6	BBS4 ZP 3 5**	C
D	BNE REL 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 IMP 1 2	CMP ABS,Y 3 4*	PHX IMP 1 3			CMP ABS,X 3 4*	DEC ABS,X 3 6*	BBS5 ZP 3 5**	D
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 IMP 1 2	SBC IMM 2 2†	NOP IMP 1 2		CPX ABS 3 4	SBC ABS 3 4†	INC ABS 3 6	BBS6 ZP 3 5**	E
F	BEO REL 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 IMP 1 2	SBC ABS,Y 3 4*†	PLX IMP 1 4			SBC ABS,X 3 4*†	INC ABS,X 3 6*	BBS7 ZP 3 5**	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

□ Newly Designed Instruction

† Cycles Add 1 when decimal mode

* Cycles Add 1 when page crossing occurs

** Cycles Add 1 when branch in same page
Add 2 when branch in different page

■ 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+M+C→A (1)(5)	69	2	2	6D	4	3	65	3	2							61	6	2	71	5	2
AND	A∧M→A (1)	29	2	2	2D	4	3	25	3	2							21	6	2	31	5	2
ASL	C←7 0←0 (1)				0E	6	3	06	5	2	0A	2	1									
BBR(#0−7)	Branch on Mb=0							−	5	3												
BBS(#0−7)	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	A∧M (6)	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 Allways (2)																					
BRK	Break													00	7	1						
BVC	Branch on V=0 (2)																					
BVS	Branch on V=1 (2)																					
CLC	0→C													18	2	1						
CLD	0→D													D8	2	1						
CLI	0→I													58	2	1						
CLV	0→V													B8	2	1						
CMP	A−M (1)	C9	2	2	CD	4	3	C5	3	2							C1	6	2	D1	5	2
CPX	X−M	E0	2	2	EC	4	3	E4	3	2												
CPY	Y−M	C0	2	2	CC	4	3	C4	3	2												
DEC	M−1→M (1)				CE	6	3	C6	5	2	3A	2	1									
DEX	X−1→X													CA	2	1						
DEY	Y−1→Y													88	2	1						
EOR	V∀M→A (1)	49	2	2	4D	4	3	45	3	2							41	6	2	51	5	2
INC	M+1→M (1)				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→7 0→C (1)				4E	6	3	46	5	2	4A	2	1									
NOP	No Operation													EA	2	1						
ORA	A∨M→A (1)	09	2	2	0D	4	3	05	3	2							01	6	2	11	5	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→S Ms→Y													7A	4	1						
RMB(#0−7)	0→Mb (4)							−	5	2												
ROL	7 0←C (1)				2E	6	3	26	5	2	2A	2	1									
ROR	→C→7 0 (1)				6E	6	3	66	5	2	6A	2	1									
RTI	Rtrn Int													40	6	1						
RTS	Rtrn Sub													60	6	1						
SBC	A−M−C̄→A (1)(5)	E9	2	2	ED	4	3	E5	3	2							E1	6	2	F1	5	2
SEC	1→C													38	2	1						
SED	1→D													F8	2	1						
SEI	1→I													78	2	1						
SMB(#0−7)	1→Mg (4)							−	5	2												
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													AA	2	1						
TAY	A→Y													A8	2	1						
TRB	Ā∧M→M				1C	6	3	14	5	2												
TSB	A∨M→M				0C	6	3	04	5	2												
TSX	S→X													BA	2	1						
TXA	X→A													8A	2	1						
TXS	X→S													9A	2	1						
TYA	Y→A													98	2	1						

(Symbol Description)

X: Index X	Ms: Designated Memory with Stack pointer	+: Add	∀ : Exclusive OR
Y: Index Y	Mb: Zero page Memory Bit	−: Subtract	n: Machine Cycles
A: Accumulator	M_7: Memory Bit 7	∧: Logical AND	#: Bytes
M: Designated Memory with Effective Address	M_6: Memory Bit 6	∨: Logical OR	

MODE

																										PROCESSOR STATUS CODES							
ZPAGE, X			ZPAGE,Y			ABS, X			ABS, Y			RELATIVE			INDIRECT			BIT ADDRESSING(OP BY BIT 0)								7	6	5	4	3	2	1	0
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	N	V	·	B	D	I	Z	C
75	4	2				7D	4	3	79	4	3				72	5	2									N	V	·	·	·	·	Z	C
35	4	2				3D	4	3	39	4	3				32	5	2									N	·	·	·	·	·	Z	·
16	6	2				1E	6	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																		M_7	M_6	·	·	·	·	Z	·
												30	2	2												·	·	·	·	·	·	·	·
												D0	2	2												·	·	·	·	·	·	·	·
												10	2	2												·	·	·	·	·	·	·	·
												80	3	2												·	·	·	·	·	·	·	·
																										·	·	·	1	·	1	·	·
												50	2	2												·	·	·	·	·	·	·	·
												70	2	2												·	·	·	·	·	·	·	·
																										·	·	·	·	·	·	·	0
																										·	·	·	·	0	·	·	·
																										·	·	·	·	·	0	·	·
																										·	0	·	·	·	·	·	·
D5	4	2				DD	4	3	D9	4	3				D2	5	2									N	·	·	·	·	·	Z	C
																										N	·	·	·	·	·	Z	C
																										N	·	·	·	·	·	Z	C
D6	6	2				DE	6	3																		N	·	·	·	·	·	Z	·
																										N	·	·	·	·	·	Z	·
																										N	·	·	·	·	·	Z	·
55	4	2				5D	4	3	59	4	3				52	5	2									N	·	·	·	·	·	Z	·
F6	6	2				FE	6	3																		N	·	·	·	·	·	Z	·
																										N	·	·	·	·	·	Z	·
																										N	·	·	·	·	·	Z	·
															6C	5	3									·	·	·	·	·	·	·	·
																										·	·	·	·	·	·	·	·
B5	4	2				BD	4	3	B9	4	3				B2	5	2									N	·	·	·	·	·	Z	·
			B6	4	2				BE	4	3															N	·	·	·	·	·	Z	·
B4	4	2				BC	4	3																		N	·	·	·	·	·	Z	·
56	6	2				5E	6	3																		0	·	·	·	·	·	Z	C
																										·	·	·	·	·	·	·	·
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	6	3																		N	·	·	·	·	·	Z	C
76	6	2				7E	6	3																		N	·	·	·	·	·	Z	C
																										(Restored)							
																										·	·	·	·	·	·	·	·
F5	4	2				FD	4	3	F9	4	3				F2	5	2									N	V	·	·	·	·	Z	(3)
																										·	·	·	·	·	·	·	1
																										·	·	·	·	1	·	·	·
									•																	·	·	·	·	·	1	·	·
																		87	97	A7	B7	C7	D7	E7	F7	·	·	·	·	·	·	·	·
95	4	2				9D	3	3	99	5	3				92	5	2									·	·	·	·	·	·	·	·
			96	4	2																					·	·	·	·	·	·	·	·
94	4	2																								·	·	·	·	·	·	·	·
74	4	2				9E	3	3																		·	·	·	·	·	·	·	·
																										N	·	·	·	·	·	Z	·
																										N	·	·	·	·	·	Z	·
																										·	·	·	·	·	·	Z	·
																										·	·	·	·	·	·	Z	·
																										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)

Description 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	V	Logical OR
Ms	Stack Memory	PCH	Program Counter High
Mb	Memory Bit	PCL	Program Counter Low
+	Add		
	Logical AND		

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

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

ASL One bit left shift. LSB is placed "0". Contents of MSB is placed C.
Operation: C ← $\boxed{7}\boxed{6}\boxed{5}\boxed{4}\boxed{3}\boxed{2}\boxed{1}\boxed{0}$ ← "0"
"P" Register: N, Z, C

BBR If specific bit of zero page is a reset state, branch relatively.
$\boxed{\text{OP-Code}}\boxed{\text{Low Order Address}}\boxed{\text{Offset}}$ 3-byte instruction
If the specific bit (a bit is decided on the instruction code) of effective address $\boxed{00}\boxed{\text{Low Order Address}}$ 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

BBS If specific bit of zero page is a set state, branch relatively.
$\boxed{\text{OP-code}}\boxed{\text{Low Order Address}}\boxed{\text{Offset}}$ 3-byte instruction
If the specific bit (a bit is decided on the instruction code) of effective address $\boxed{00}\boxed{\text{Low Order Address}}$ is a set state, relative branch by the $\boxed{\text{Offset}}$ value to base with lead address of next instruction.
Operation: branch when $M_b = 1$
"P" Register: Not affected

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

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

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

BIT Test the memory bit by the accumulator.
Operation: A M, M_7 → N, M_6 → 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)

BMI 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.
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: $\overset{B}{1}$

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: $\overset{C}{0}$

CLD Clear the decimal mode.

Operation: 0 → C

"P" Register: $\overset{D}{0}$

CLI Clear the interrupt disenable flag (I).

Operation: 0 → I

"P" Register: $\overset{V}{0}$

CLV Clear overflow flag.

Operation: 0 → V

"P" Register: $\overset{V}{0}$

CMP Compare memory with accumulator.

Operation: A-M

The result is not stored. If it is negative, 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 registerX.

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   M → A

"P" Register: N, Z

INC Increment the contents of memory.

Operation: M+1 → M

"P" Register: N, Z

INY Increment the contents of index register Y.

Operation: Y+1 → Y

"P" Register: N, Z

JMP Execution of program jumps to designation address.

Operation:

OP-Code	Operand	Operand

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

"P" Register: Not affected

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

The disignation address by operands with 2-bytes is stored into the PCL and PCH.

Lead address of next instruction (2-byte)

→ Ms, S−1 → S

→ Ms, S−1 → S   "P" Register: Not affected

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

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

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

LSR One bit right shift. MSB (7 bit) is placed to 0, LSB (0 bit) is loaded the C.
Operation: 0 → | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | → 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 M → A
"P" Register: N, Z

PHA Store the contents of the accumulator into the memory stack.
Operation: A → Ms, S–1 → S
"P" Register: Not affected

PHP Store the contents of the register P into the stack.
Operation: P → Ms, S–1 → S
"P" Register: Not affected

PHX Store the contents of the index register X into the stack.
Operation: X → Ms, S–1 → S
"P" Register: Not affected

PHY Store the contents of the index register Y into the stack.
Operation: Y → Ms, S–1 → S
"P" Register: Not affected

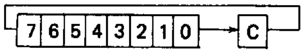
PLA Pull accumulator from stack.
Operaton: Ms → A, S + 1 → S
"P" Register: N, Z

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

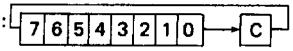
PLX Pull X register from stack.
Operation: Ms → X, S + 1 → S
"P" Register: N, Z

PLY Pull Y register from stack.
Operation: Ms → Y, S + 1 → 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 → 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: Ms → P, S + 1 → S
Ms → PCL, S + 1 → S
Ms → PCH, S + 1 → S
"P" Register: Restore

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: Ms → PCL, S + 1 → S
Ms → PCH, S + 1 → S
"P" Register: Not affected

SBC Subtract memory and borrow from accumulator, and the result is stored into the accumulator.

Operation: $A - M - C \to A$

C = borrow

"P" Register: N, V, Z, C

SEC Set carry flag.

Operation: $1 \to C$

"P" Register: $\begin{matrix}C\\1\end{matrix}$

SED Set decimal flag.

Operation: $1 \to D$

"P" Register: $\begin{matrix}D\\1\end{matrix}$

SEI Set disable interrupt status.

Operation: $1 \to I$

"P" Register: $\begin{matrix}I\\1\end{matrix}$

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: $1 \to M_b$

"P" Register: Not affected

STA Store the contents of the accumulator into the memory.

Operation: $A \to M$

"P" Register: Not affected

STX Store the contents of the index register X into the memory.

Operation: $X \to M$

"P" Register: Not affected

STY Store the contents of the index register Y into the memory.

Operation: $Y \to M$

"P" Register: Not affected

STZ Clear the contents of memory.

Operation: $0 \to M$

"P" Register: Not Affected

TAX Transfer the contents of the accumulator to the index register X.

Operation: $A \to M$

"P" Register: N, Z

TAY Transfer the contents of the accumulator to the index register Y.

Operation: $A \to Y$

"P" Register: N, Z

TRB Reset the contents of memory by accumulator, and test at the same time.

Operation: $\overline{A} \wedge M \to M$

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: $A \vee M \to M$

If the result is zero, Z flag = 1

"P" Register: Z

TSX Transfer stack pointer to the index register X.

Operation: $S \to X$

"P" Register: N, Z

TXA Transfer the contents of the index register X to the accumulator.

Operation: $X \to A$

"P" Register: N, Z

TXS Transfer the contents of the index register X to stack pointer.

Operation: $X \to S$

"P" Register: Not affected

TYA Transfer the contents of the index register Y to the accumulator.

Operation: $Y \to A$

"P" Register: N, Z

■ DETAILED INSTRUCTION OPERATION

ADDRESS MODE	CYCLE	ADDRESS BUS	DATA BUS	R/$\overline{W}$
1 Immediate IMM	1	PC	OP CODE	1
LDA(A9), AND(29), ORA(09), EOR(49), CMP(C9),	2	PC+1	ID	1
BIT(89), ADC(69), SBC(E9), LDX(A2), LDY(A0),	(1)2a	PC+2	IO	1
CPX(E0), CPY(C0)				
2a Absolute ABS	1	PC	OP CODE	1
	2	PC+1	AAL	1
LDA(AD), STA(8D), ADC(6D), SBC(ED), AND(2D),	3	PC+2	AAH	1
ORA(0D), EOR(4D), CMP(CD), BIT(2C), LDX(AE),	4	AA	DATA	1/0
LDY(AC), STX(8E), STY(8C), CPX(EC), CPY(CC),	(1)4a	PC+3	IO	1
STZ(9C)				
2b Absolute (R·M·W) ABS	1	PC	OP CODE	1
TRB(1C), TSB(0C), LSR(4E), ASL(0E), ROL(2E),	2	PC+1	AAL	1
ROR(6E), INC(EE), DEC(CE)	3	PC+2	AAH	1
	4	AA	DATA	1
	5	AA	IO	1
	6	AA	DATA	0
2c Absolute (JUMP) ABS	1	PC	OP CODE	1
JMP(4C)	2	PC+1	PCL	1
	3	PC+2	PCH	1
	1	NEW PC	OP CODE	1
2d Absolute (Jump to subroutine) ABS	1	PC	OP CODE	1
JSR(20)	2	PC+1	NEW PCL	1
	3	01,S	IO	1
	4	01,S	PCH	0
	5	01,S−1	PCL+2	0
	6	PC+2	NEW PCH	1
	1	NEW PC	OP CODE	1
3a Zero Page ZP	1	PC	OP CODE	1
LDA(A5), STA(85), ADC(65), SBC(E5), AND(25),	2	PC+1	BAL	1
ORA(05), EOR(45), CMP(C5), BIT(24), LDX(A6),	3	00,BAL	DATA	1/0
LDY(A4), STX(86), STY(84), CPX(E4), CPY(C4),	(1)3a	PC+2	IO	1
STZ(64)				
3b Zero Page (R·M·W) ZP	1	PC	OP CODE	1
TRB(14), TSB(04), LSR(46), ASL(06), ROL(26),	2	PC+1	BAL	1
ROR(66), INC(E6), DEC(C6)	3	00,BAL	DATA	1
	4	00,BAL	IO	1
	5	00,BAL	DATA	0

ADDRESS MODE	CYCLE	ADDRESS BUS	DATA BUS	$R/\overline{W}$
4 Accumulator ACC	1	PC	OP CODE	1
LSR(4A), ASL(0A), ROL(2A), ROR(6A), INC(1A), DEC(3A)	2	PC+1	IO	1
5a Implied IMP	1	PC	OP CODE	1
SEC(38), CLC(18), SEI(78), CLI(58), SED(F8), CLD(D8), CLV(B8), NOP(EA), INX(E8), INY(C8), DEX(CA), DEY(88), TAX(AA), TXA(8A), TAY(A8), TYA(98), TSX(BA), TXS(9A)	2	PC+1	IO	1
5b Implied (Stack Push) IMP	1	PC	OP CODE	1
PHA(48), PHP(08), PHX(DA), PHY(5A)	2	PC+1	IO	1
	3	01,S	REG	0
5c Implied (Stack Pull) IMP	1	PC	OP CODE	1
PLA(68), PLP(28), PLX(FA), PLY(7A)	2	PC+1	IO	1
	3	01,S	IO	1
	4	01,S+1	REG	1
5d Implied (Return from subroutine) IMP	1	PC	OP CODE	1
RTS(60)	2	PC+1	IO	1
	3	01,S	IO	1
	4	01,S+1	NEW PCL	1
	5	01,S+2	NEW PCH	1
	6	NEW PC	IO	1
	1	PC+1	OP CODE	1
5e Implied (Return from interrupt) IMP	1	PC	OP CODE	1
RTI(40)	2	PC+1	IO	1
	3	01,S	IO	1
	4	01,S+1	P	1
	5	01,S+2	NEW PCL	1
	6	01,S+3	NEW PCH	1
	1	NEW PC	OP CODE	1
5f Implied (Interrupt) IMP	1	PC	OP CODE	1
BRK(00), $\overline{RES}$, $\overline{IRQ}$, $\overline{NMI}$	2	PC**	IO	1
	3	01,S	PCH	1/0*
	4	01,S−1	PCL***	1/0*
	5	01,S−2	P	1/0*
	6	VA	AAVL	1
	7	VA+1	AAVH	1
	1	AAV	OP CODE	1

* RESET: "1"
** BRK: PC+1
*** BRK: PCL+2

ADDRESS MODE	CYCLE	ADDRESS BUS	DATA BUS	R/$\overline{W}$
6 Indexed Indirect (IND, X)	1	PC	OP CODE	1
LDA(A1), STA(81), ADC(61), SBC(E1), AND(21),	2	PC+1	BAL	1
ORA(01), EOR(41), CMP(C1)	3	PC+1	IO	1
	4	00,BAL+X	AAL	1
	5	00,BAL+X+1	AAH	1
	6	AA	DATA	1/0
	(1)6a	PC+2	IO	1
7 Indirect Index (IND), Y	1	PC	OP CODE	1
LDA(B1), STA(91), ADC(71), SBC(F1), AND(31),	2	PC+1	IAL	1
ORA(11), EOR(51), CMP(D1)	3	00,IAL	AAL	1
	4	00,IAL+1	AAH	1
	(4)(2)4a	00,IAL+1	IO	1
	5	AA+Y	DATA	1/0
	(1)5a	PC+2	IO	1
8a Zero Page Index X ZP, X	1	PC	OP CODE	1
LDA(B5), STA(95), ADC(75), SBC(F5), AND(35),	2	PC+1	BAL	1
ORA(15), EOR(55), CMP(D5), BIT(34), LDY(B4),	3	PC+1	IO	1
STY(94), STZ(74)	4	00,BAL+X	DATA	1/0
	(1)4a	PC+2	IO	1
8b Zero Page Index X (R-M-W) ZP, X	1	PC	OP CODE	1
LSR(56), ASL(16), ROL(36), ROR(76), INC(F6),	2	PC+1	BAL	1
DEC(D6)	3	PC+1	IO	1
	4	00,BAL+X	DATA	1
	5	00,BAL+X	IO	1
	6	00,BAL+X	DATA	0
9 Zero Page Index Y ZP, Y	1	PC	OP CODE	1
LDX(B6), STX(96)	2	PC+1	BAL	1
	3	PC+1	IO	1
	4	00,BAL+Y	DATA	1/0
10a Absolute Index X ABS, X	1	PC	OP CODE	1
LDA(BD), STA(9D), ADC(7D), SBC(FD), AND(3D),	2	PC+1	AAL	1
ORA(1D), EOR(5D), CMP(DD), BIT(3C), LDY(BC),	3	PC+2	AAH	1
STZ(9E)	(4)(2)3a	PC+2	IO	1
	4	AA+X	DATA	1/0
	(1)4a	PC+3	IO	1

ADDRESS MODE	CYCLE	ADDRESS BUS	DATA BUS	R/W
10b Absolute Index X (R·M·W) ABS, X	1	PC	OP CODE	1
LSR(5E), ASL(1E), ROL(3E), ROR(7E), INC(FE),	2	PC+1	AAL	1
DEC(DE)	3	PC+2	AAH	1
	(2)3a	PC+2	IO	1
	4	AA+X	DATA	1
	5	AA+X	IO	1
	5	AA+X	DATA	0
11 Absolute Index Y ABS, Y	1	PC	OP CODE	1
LDA(B9), STA(99), ADC(79), SBC(F9), AND(39),	2	PC+1	AAL	1
ORA(19), EOR(59), CMP(D9), LDX(BE)	3	PC+2	AAH	1
	(4)(2)3a	PC+2	IO	1
	4	AA+Y	DATA	1/0
	(1)4a	PC+3	IO	1
12 Relative REL	1	PC	OP CODE	1
BMI(30), BPL(10), BCC(90), BCS(B0), BEQ(F0),	2	PC+1	off	1
BNE(D0), BVS(70), BVC(50), BRA(80)	(3)2a	PC+2	IO	1
	(2)2b	PC+2	IO	1
	(5)1	PC+2+(off)	OP CODE	1
13 Indirect IND	1	PC	OP CODE	1
LDA(B2), STA(92), ADC(72), SBC(F2), AND(32),	2	PC+1	IAL	1
ORA(12), EOR(52), CMP(D2)	3	00,IAL	AAL	1
	4	00,IAL+1	AAH	1
	5	AA	DATA	1/0
	(1)5a	PC+2	IO	1
14 Absolute Indirect (IND)	1	PC	OP CODE	1
JMP(6C)	2	PC+1	AAL	1
	3	PC+2	AAH	1
	4	PC+2	IO	1
	5	AA	PCL	1
	6	AA+1	PCH	1
	1	NEW PC	OP CODE	1
15 Absolute Indexed Indirect (IND), X	1	PC	OP CODE	1
JMP(7C)	2	PC+1	AAL	1
	3	PC+2	AAH	1
	4	PC+2	IO	1
	5	AA+X	PCL	1
	6	AA+X+1	PCH	1
	1	NEW PC	OP CODE	1

ADDRESS MODE	CYCLE	ADDRESS BUS	DATA BUS	R/W
16a Bit Addressing (R·M·W)	1	PC	OP CODE	1
RMB(07~77), SMB(87~F7)	2	PC+1	BAL	1
	3	00,BAL	DATA	1
	4	00,BAL	IO	1
	5	00,BAL	DATA	0
16b Bit Addressing (Branch)	1	PC	OP CODE	1
BBR(0F~7F), BBS(8F~FF)	2	PC+1	BAL	1
	3	00,BAL	DATA	1
	4	00,BAL	IO	1
	5	PC+2	off	1
	5a	PC+3	IO	1
	5b	PC+3	IO	1
	1	PC+3+off	OP CODE	1

ABBREVIATIONS

AA	Absolute Address	–L	Lower 8 bit Address
IO	Internal Operation	ID	Immediate Data
REG	Data of Each Register (ACC, X, Y, P)	DATA	Memory Data
PC	Program Counter	BAL	Base Address (Low)
S	Stack Address	off	Off-Set Data
VA	Vector Address	IA	Indirect Address
AAV	Absolute Address Vector	X, Y	Index Register
P	Processor Status Register	R-M-W	Read-Modify-Write
–H	Higher 8 bit Address		

NOTE: (1) Add 1 cycle if decimal mode of ADC, SBC

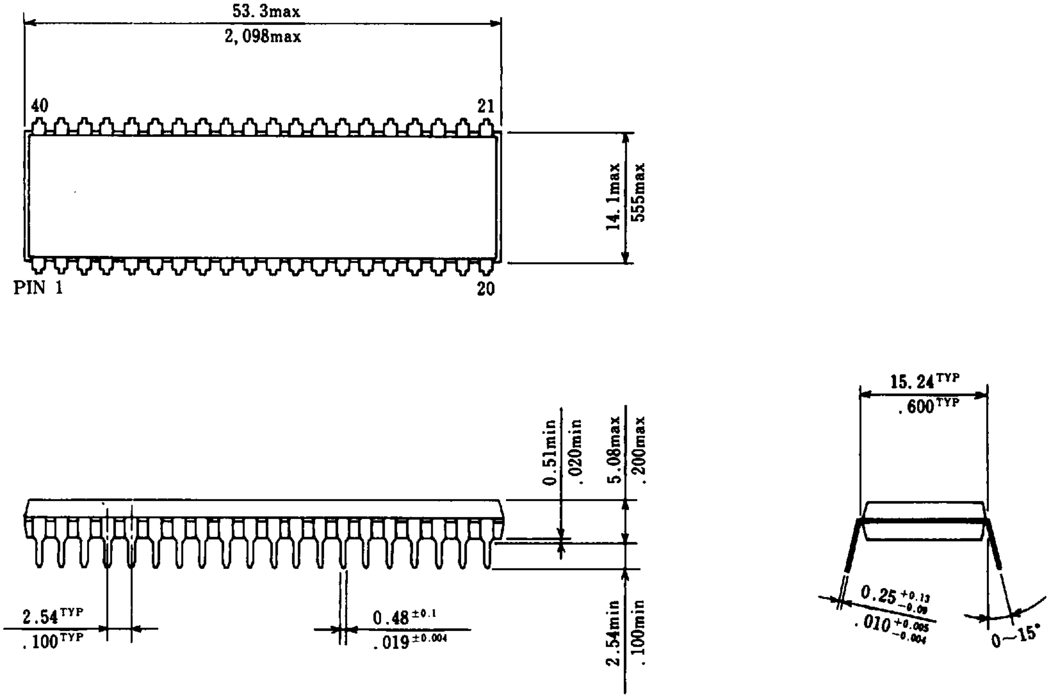
(2) Add 1 cycle if page boundary is crossed

(3) Add 1 cycle if branch is taken

(4) Add 1 cycle for write

(5) PC+2 if branch is not taken

- **40-PIN DUAL-IN-LINE PACKAGE (UNIT: $\frac{\text{mm}}{\text{inch}}$)**


53.3max
2,098max
40
21
PIN 1
20
14.1max
555max
0.51min
.020min
5.08max
.200max
2.54TYP
.100TYP
0.48±0.1
.019±0.004
2.54min
.100min
15.24TYP
.600TYP
0.25 +0.13 −0.09
.010 +0.005 −0.004
0~15°