ABOV SEMICONDUCTOR Co., Ltd.
4-BIT SINGLE-CHIP MICROCONTROLLERS

MC40P5004
MC40P5204
MC40P5404

User's Manual

REVISION HISTORY

VERSION 1.01 (Sep 3, 2010) this book

Corrected the instruction explanation of NEGA, ALEI, EORM.

VERSION 1.0 (Aug 12, 2010)

Added circuit diagram of MC40P5404.

VERSION 0.3 (MAY 11, 2010)

The REMDRV port was renamed to REMOUT port.
The MC40P5004R model was changed to 20SSOP package from 20TSSOP.

VERSION 0.2 (Nov 11, 2009)

The package with push-pull REMOUT port (MC40P5102 and MC40P5304) is removed from line-up.
The 20TSSOP package model MC40P5004R is added to line-up.

VERSION 0.1 (Apr 21, 2008)

Version 1.01
Published by FAE Team
©2009 ABOV Semiconductor Co., Ltd. All rights reserved.

Additional information of this manual may be served by ABOV Semiconductor offices in Korea or Distributors.

ABOV Semiconductor reserves the right to make changes to any information here in at any time without notice.

The information, diagrams and other data in this manual are correct and reliable; however, ABOV Semiconductor is in no way responsible for any violations of patents or other rights of the third party generated by the use of this manual.

Table of Contents

1. OVERVIEW 4
1.1 Description 4
1.2 Features 4
1.3 Ordering Information 5
2. BLOCK DIAGRAM 6
2.1 MC40P5004 (20 pin package) 6
2.2 MC40P5204 (24 pin package) 7
2.3 MC40P5404 (20 pin package) 8
3. PIN ASSIGNMENT 9
4. PACKAGE DIAGRAM 10
5. PIN DESCRIPTION 13
5.1 PIN DESCRIPTION 13
6. PORT STRUCTURES 14
7. ELECTRICAL CHARACTERISTICS 16
7.1 Absolute Maximum Ratings (Ta = 25℃) 16
7.2 Recommended Operating Conditions 16
7.3 Electrical characteristics (Ta=25℃, V_{DD}= 3V) 16
8. Architecture 19
8.1 Program Memory (EPROM) 19
8.2 EPROM Address Register 20
8.3 Data memory (RAM) 21
8.4 X-register (X) 22
8.5 Y-register (Y) 22
8.6 Accumulator (Acc) 22
8.7 Arithmetic and Logic Unit (ALU) 22
8.8 State Counter (SC) 22
8.9 Clock Generator 23
8.10 Pulse Generator 24
8.11 Reset Operation 25
8.12 STOP Operation 27
8.13 Port Operation 27
9. Instruction 28
9.1 Instruction Table 29
9.2 DETAILS OF INSTRUCTION SYSTEM 31
9.3 Assembler Macro 44
10. SPGM(Serial Program) 45
Summary of Protocol 45
11. APPLICATION 49
11.1 Circuit Diagram of MC40P5004 and MC40P5404 49

MC40P5004
MC40P5204
MC40P5404

CMOS SINGLE-CHIP 4-BIT MICROCONTROLLER

1. OVERVIEW

1.1 Description

The MC40P5x04 series is 4-bit remote control MCU which uses CMOS technology and the 4K bytes EPROM version. This enables transmission code outputs of different configurations, multiple custom code output, and double push key output for easy fabrication. The MC40P5x04 series is suitable for remote control of TV, VCR, FANS, Air-conditioners, Audio Equipments, Toys, Games etc.

1.2 Features

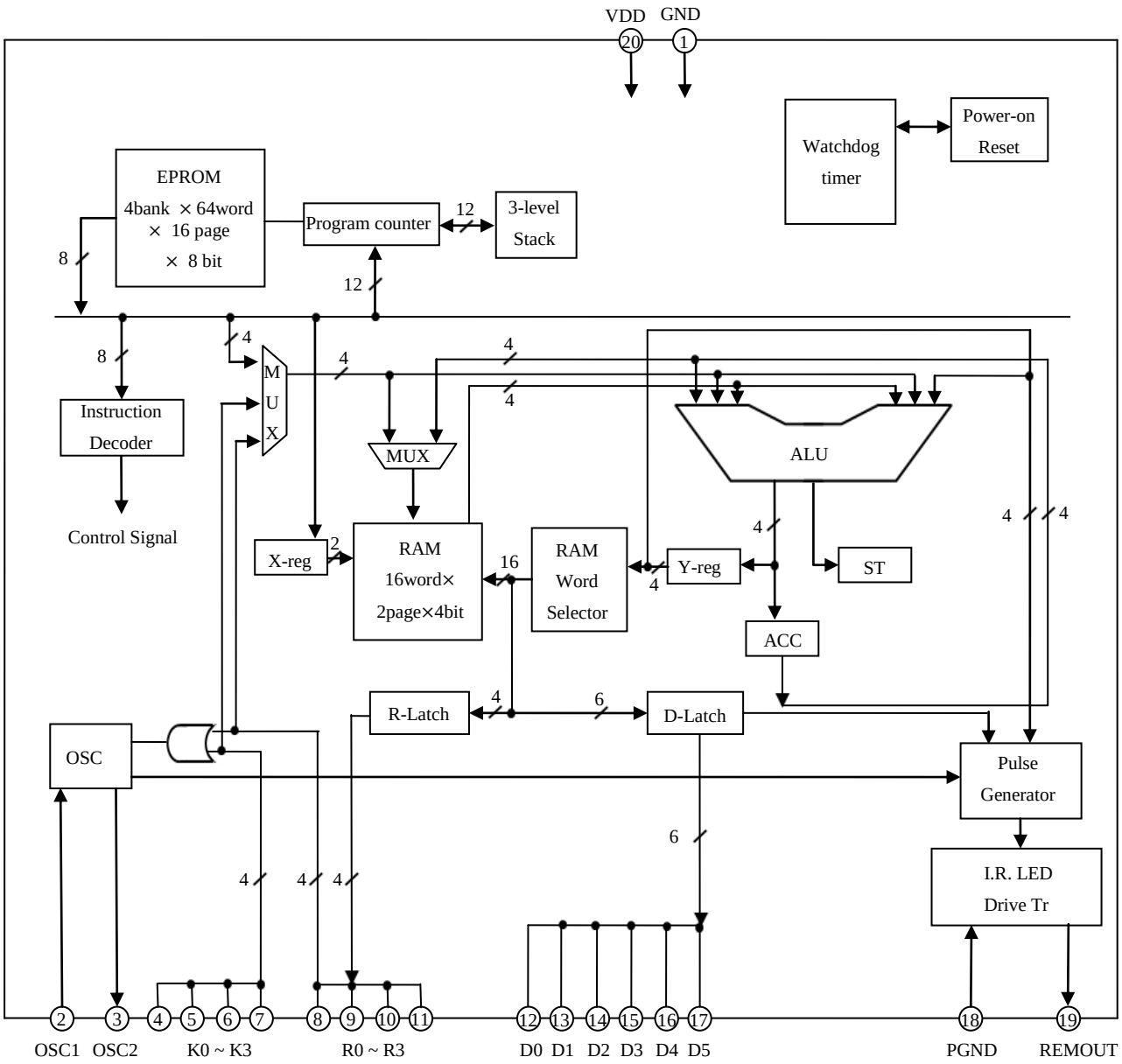
- **Program memory : 4,096 bytes**
  MTP : 1K * 4, 2K * 2, 4K * 1
- **Data memory : 32 × 4 bits**
- **43 types of instruction set**
- **3 levels of subroutine nesting**
- **Operating frequency : 2.4MHz ~ 4MHz**
- **Instruction cycle : f_{osc}/48 or f_{osc}/12**
- **CMOS process (Single 3.0V power supply)**
- **Stop mode (Through internal instruction)**
- **Released stop mode by key input**
- **Built in Power-on Reset circuit**
- **Built in Transistor for I.R LED Drive**
  - I_{OL}=250mA at V_{DD}=3V and V_O=0.3V
  - I_{OL}=500mA at V_{DD}=3V and V_O=0.52V
- **Built in Low Voltage reset circuit**
- **Built in a watch dog timer (WDT)**
- **Low operating voltage : 1.8 ~ 3.6V**
- **20/24 pin SOP and 20 pin SSOP package**

1.3 Ordering Information

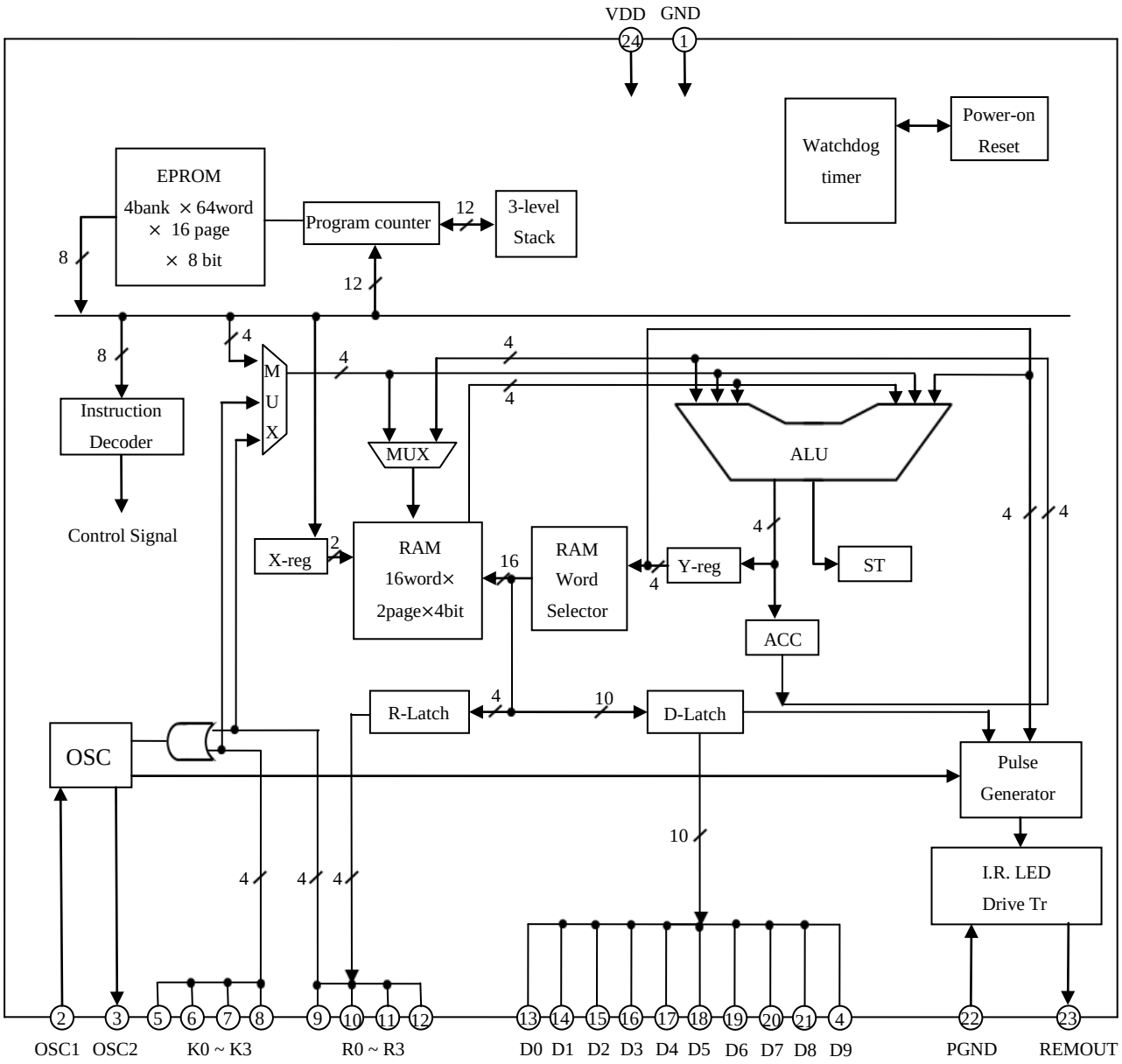
Series	MC40P5004D	MC40P5004R	MC40P5204D	MC40P5404D
Program memory	4,096			
Data memory	32 x 4			
I/O Ports	2	2	2	2
Input Ports	6	6	6	6
Output Ports	6 (D0~D5)	6 (D0~D5)	10 (D0~D9)	7 (D0~D6)
Built-in Drive Tr.	O	O	O	O
Package	20SOP	20SSOP	24SOP	20SOP

2. BLOCK DIAGRAM

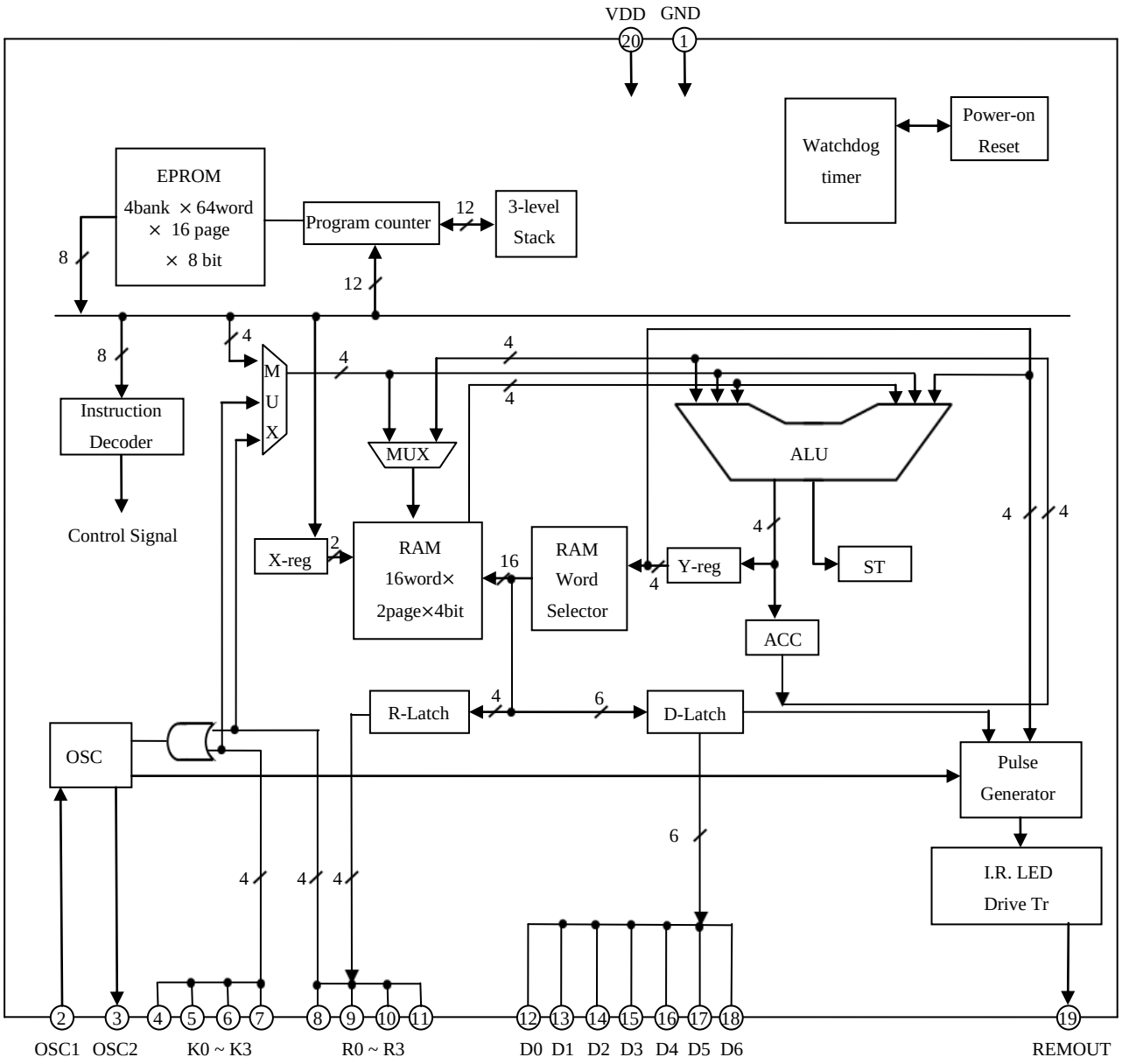
2.1 MC40P5004 (20 pin package)



2.2 MC40P5204 (24 pin package)



2.3 MC40P5404 (20 pin package)



3. PIN ASSIGNMENT

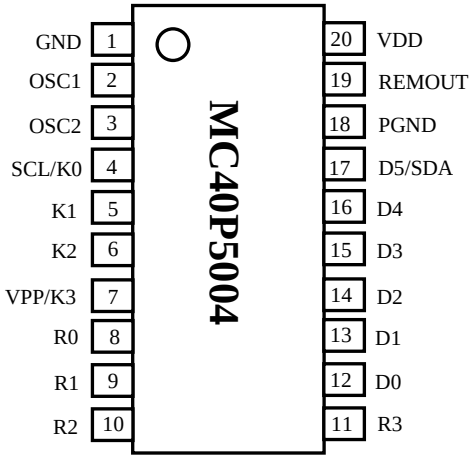


Fig 3-1 MC40P5004 Pin Assignment
(20 PIN)

REMOUT : open drain output

VPP : K3 (PIN No.7)

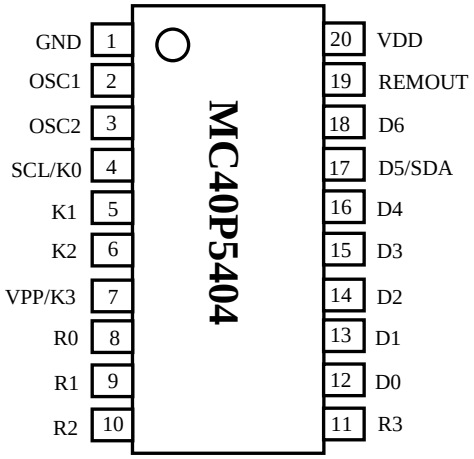


Fig 3-3 MC40P5404 Pin Assignment
(20 PIN)

REMOUT : open drain output

VPP : K3 (PIN No.7)

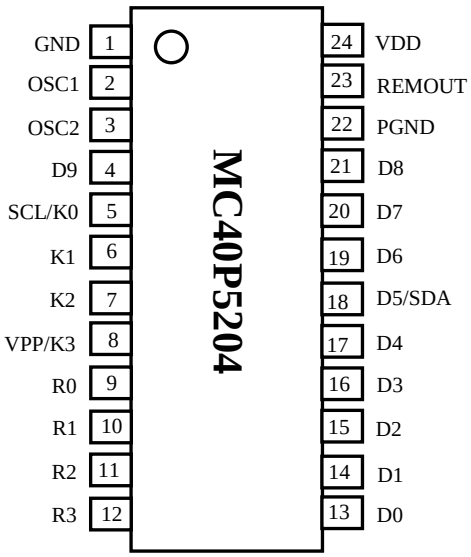


Fig 3-2 MC40P5204 Pin Assignment
(24 PIN)

REMOUT : open drain output

VPP : K3 (PIN No.8)

4. PACKAGE DIAGRAM

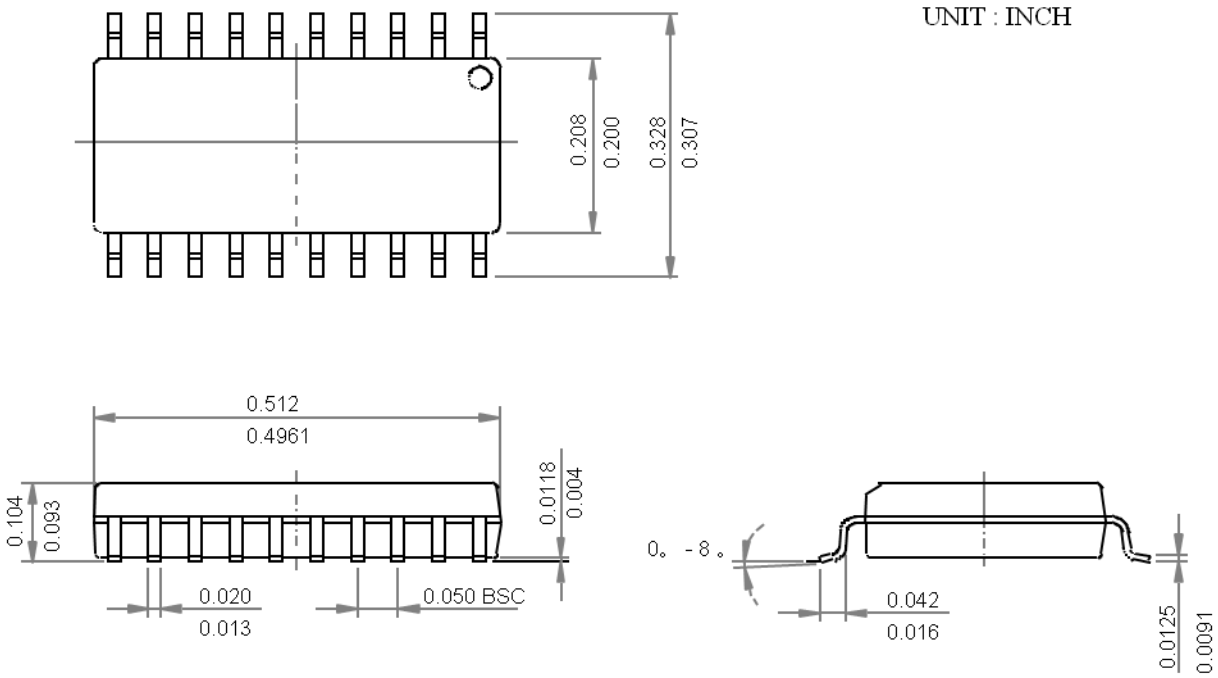


Fig 4-1 20SOP (209MIL)

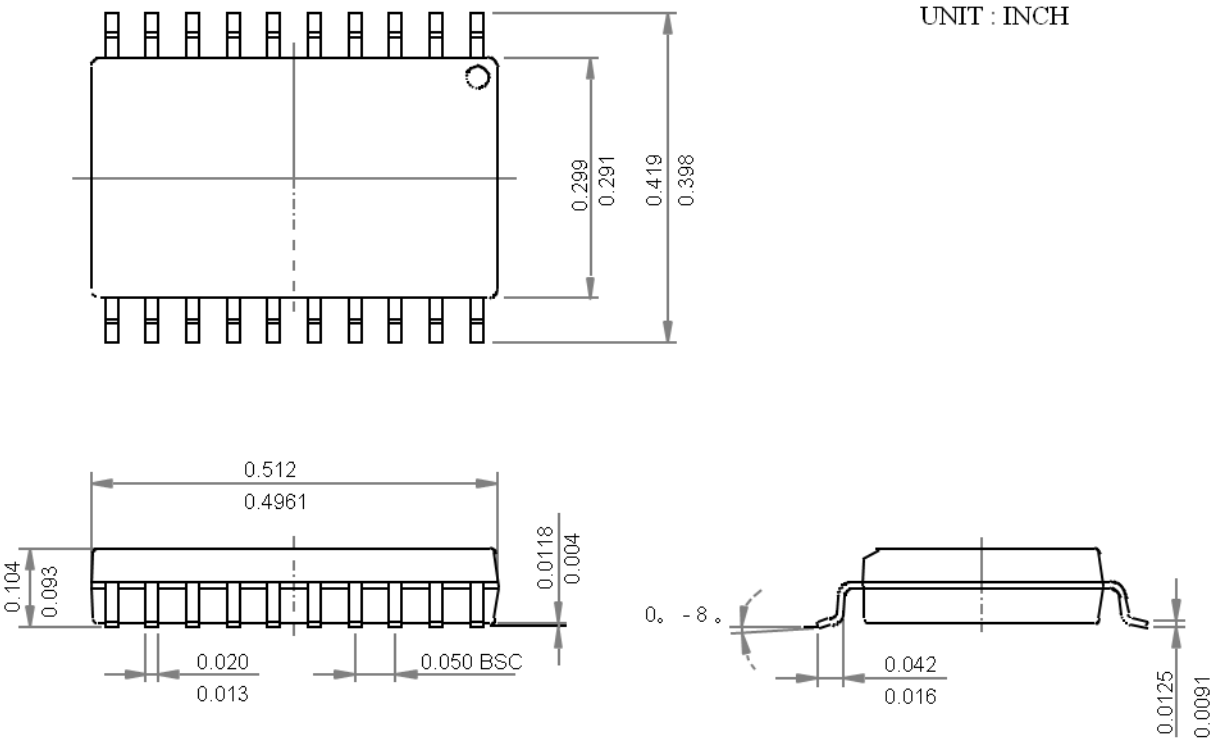


Fig 4-2 20SOP (300MIL)

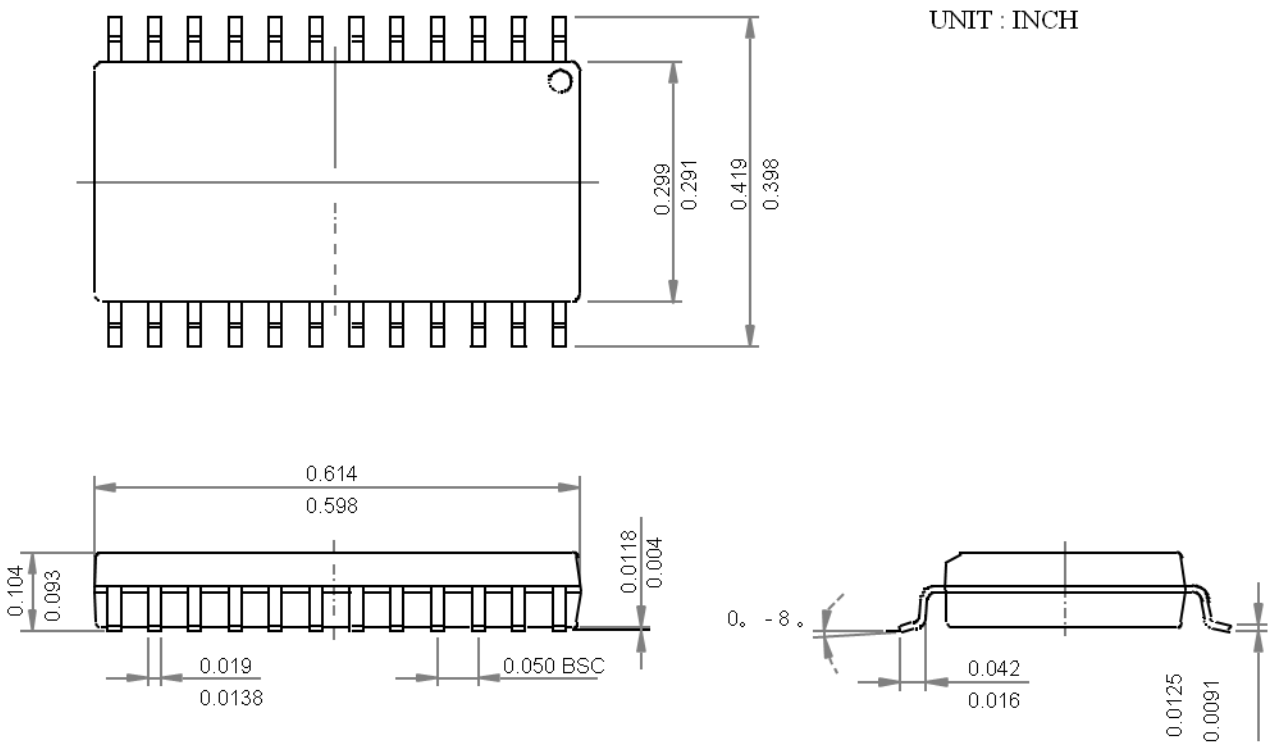
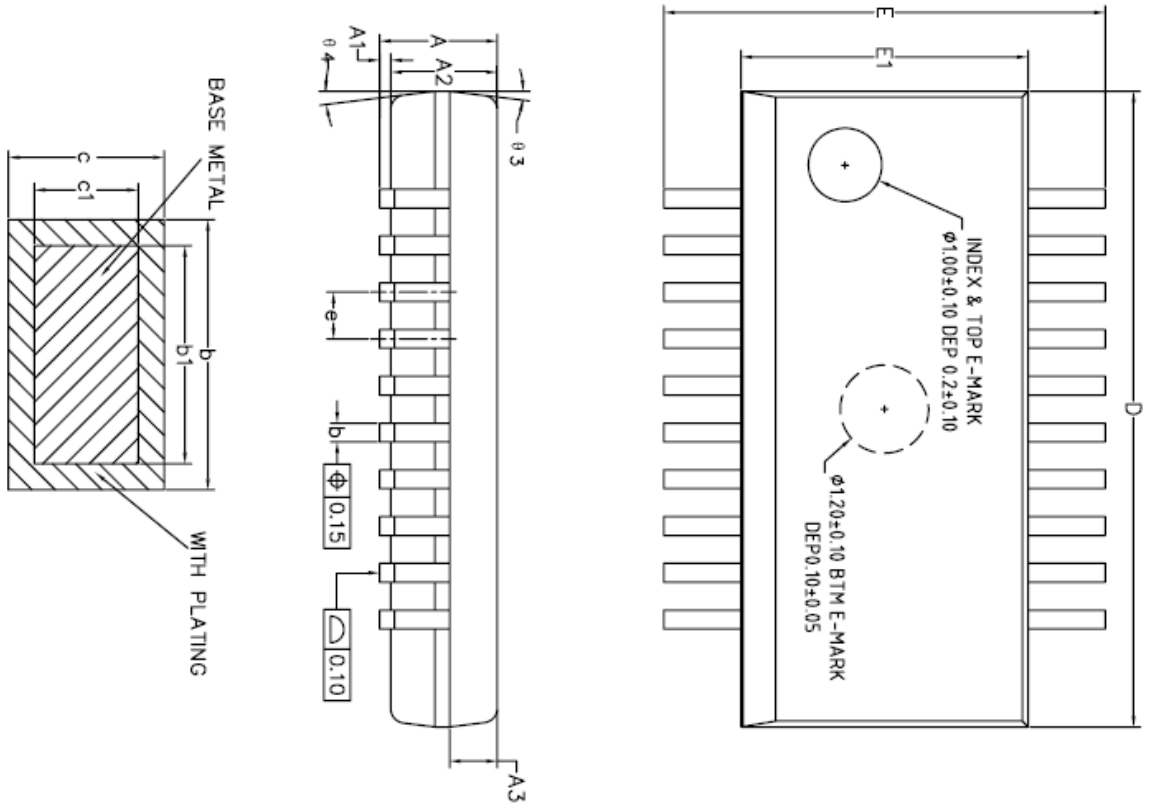


Fig 4-3 24SOP (300MIL)



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	1.35	1.60	1.75
A1	0.10	0.15	0.25
A2	1.25	1.45	1.65
A3	0.55	0.65	0.75
b	0.21	–	0.31
b1	0.20	0.25	0.27
c	0.20	–	0.25
c1	0.19	0.20	0.21
D	8.53	8.63	8.73
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	0.535	0.635	0.735
L	0.45	0.60	0.80
L1	1.05REF		
L2	0.25BSC		
R	0.08	–	–
R1	0.08	–	–
h	0.30	0.40	0.50
θ	0°	–	8°
θ 1	6°	8°	10°
θ 2	6°	8°	10°
θ 3	5°	7°	9°
θ 4	5°	7°	9°

Fig 4-4 20SSOP

5. PIN DESCRIPTION

5.1 PIN DESCRIPTION

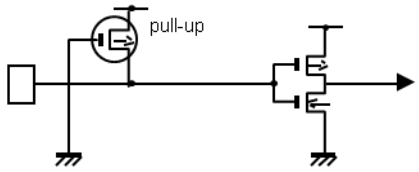
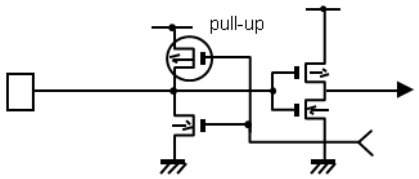
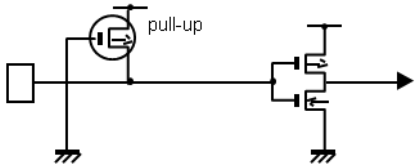
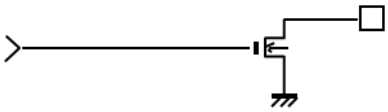
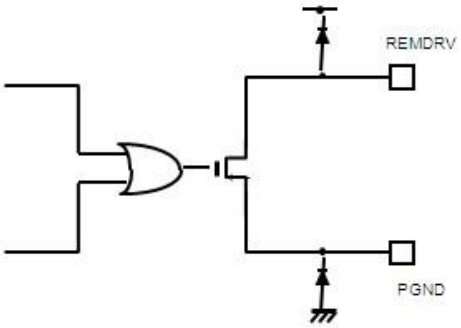
Pin	I/O	Function
VDD	-	Connected to 1.8 ~ 3.6V power supply
GND	-	Connected to 0V power supply.
K0 ~ K3	Input	4-bit input port with built in pull-up resistor. STOP mode is released by "L" input of each pin.
D0 ~ D9	Output	Each can be set and reset independently. The output is the structure of N-channel-open-drain. D0~D3 are “L” output at STOP MODE. D4 ~D5 keep status before stop mode at STOP MODE.
R0 ~ R1	Input	2-bit input port with built in pull-up resistor. STOP mode is released by "L" input of each pin.
R2 ~ R3	I/O	2-bit I/O port with built in pull-up resistor. STOP mode is released by "L" input of each pin. Input mode is set only when each of them output "H". The output is in the form of C-MOS. In outputting, each can be set and reset independently (or at once.)
OSC1	Input	Oscillator input. Input to the oscillator circuit and connection point for ceramic resonator. A feedback resistor is connected between this pin and OSC2.
OSC2	Output	Connect a resonator between this pin and OSC1.
PGND	-	Ground pin for internal high current N-channel transistor. (connected to GND)
REMOUT	Output	High current output port for driving I.R.LED. The output is in the form N-channel open drain.

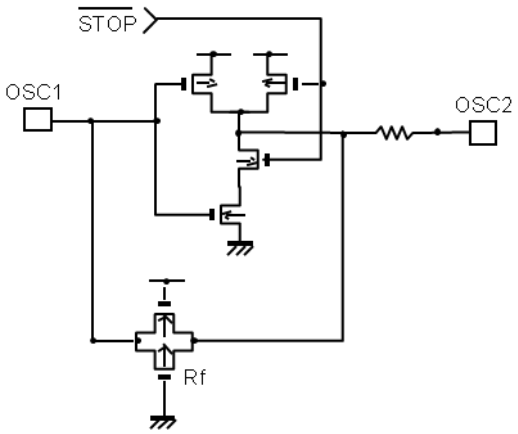
Note: D port pin mapping is

MC40P5004 (D0 ~ D5), MC40P5204(D0~D9), MC40P5404(D0~D6)

There is no PGND in MC40P5404.

6. PORT STRUCTURES

Pin	I/O	I/O circuit	Note
R0 ~ R1	I		- Built in MOS Tr for pull-up, about 140 kΩ.
R2 ~ R3	I/O		- CMOS output. - "H" output at reset. - Built in MOS Tr for pull-up, about 140 kΩ.
K0 ~ K3	I		- Built in MOS Tr for pull-up, about 140 kΩ.
D0 ~ D9	O		- Open drain output. - "L" output at reset.
REMOUT	O		- Open drain output - Output Tr. Disable at reset.

OSC2	O		- Built in feedback-resistor about 1 MΩ
OSC1	I		

Note: The gate voltage of REMOUT port (NMOS Transistor) is always higher than VDD voltage when it drives REMOUT port as Logic “LOW”..

7. ELECTRICAL CHARACTERISTICS

7.1 Absolute Maximum Ratings (Ta = 25℃)

Parameter	Symbol	Max. rating	Unit
Supply Voltage	V_{DD}	-0.3 ~ 5.0	V
Power dissipation	P_D	700 *	mV
Storage temperature range	T_{STQ}	-55 ~ 125	℃
Input voltage	V_{IN}	-0.3 ~ V_{DD}+0.3	V
Output voltage	V_{OUT}	-0.3 ~ V_{DD}+0.3	V

*Thermal derating above 25℃: 6mW per degree ℃ rise in temperature

7.2 Recommended Operating Conditions

Parameter	Symbol	Condition	Rating	Unit
Supply Voltage	V_{DD}	2.4MHz ~ 4MHz	1.8 ~ 3.6	V
Operating temperature	T_{OPR}	-	-20 ~ +70	℃

7.3 Electrical characteristics (Ta=25℃, V_{DD}= 3V)

Parameter	Symbol	Limits			Unit	Condition
		Min.	Typ.	Max.		
Input H current	I_{IH}	-	-	1	uA	VI=V_{DD}
K Pull-up Resistance	R_{PU1}	70	140	300	kΩ	VI=GND
R Pull-up Resistance	R_{PU2}	70	140	300	kΩ	VI=GND, Output off
Feedback Resistance	R_{FD}	0.3	1.0	3.0	MΩ	V_{OSC1}=GND, V_{OSC2}=VDD
K, R input H voltage	V_{IH1}	2.1	-	-	V	-
K, R input L voltage	V_{IL1}	-	-	0.9	V	-
D, R output L voltage	V_{OL2}*1	-	0.15	0.4	V	I_{OL}=3mA
OSC2 output L voltage	V_{OL3}	-	0.4	0.9	V	I_{OL}=150uA
OSC2 output H voltage	V_{OH3}	2.1	2.5	-	V	I_{OH}=-150uA

REMOUT output L current		I_{OL4}*2	-	250 500	-	mA	V_{OL4}=0.3V V_{OL4}=0.52V
D, R output leakage current		I_{OLK2}	-	-	1	uA	V_{0UT}=V_{DD}, Output off
Low Voltage Reset voltage		V_{LVR}	-	1.5	-	V	
Current on STOP mode		I_{STP}	-	-	1	uA	At STOP mode
Operating Supply current		I_{DD2} *3	-	-	5	mA	f_{OSC}=4MHz
System clock frequency	f_{OSC}/48	f_{OSC}	2.4	-	4	MHz	MHZ version

*1 Refer to Fig.7-1 < I_{OL2} vs. V_{OL2} Graph>

*2 Refer to Fig.7-2 < I_{OL4} vs. V_{OL4} Graph>

*3 I_{DD1}, I_{DD2}, is measured at RESET mode.

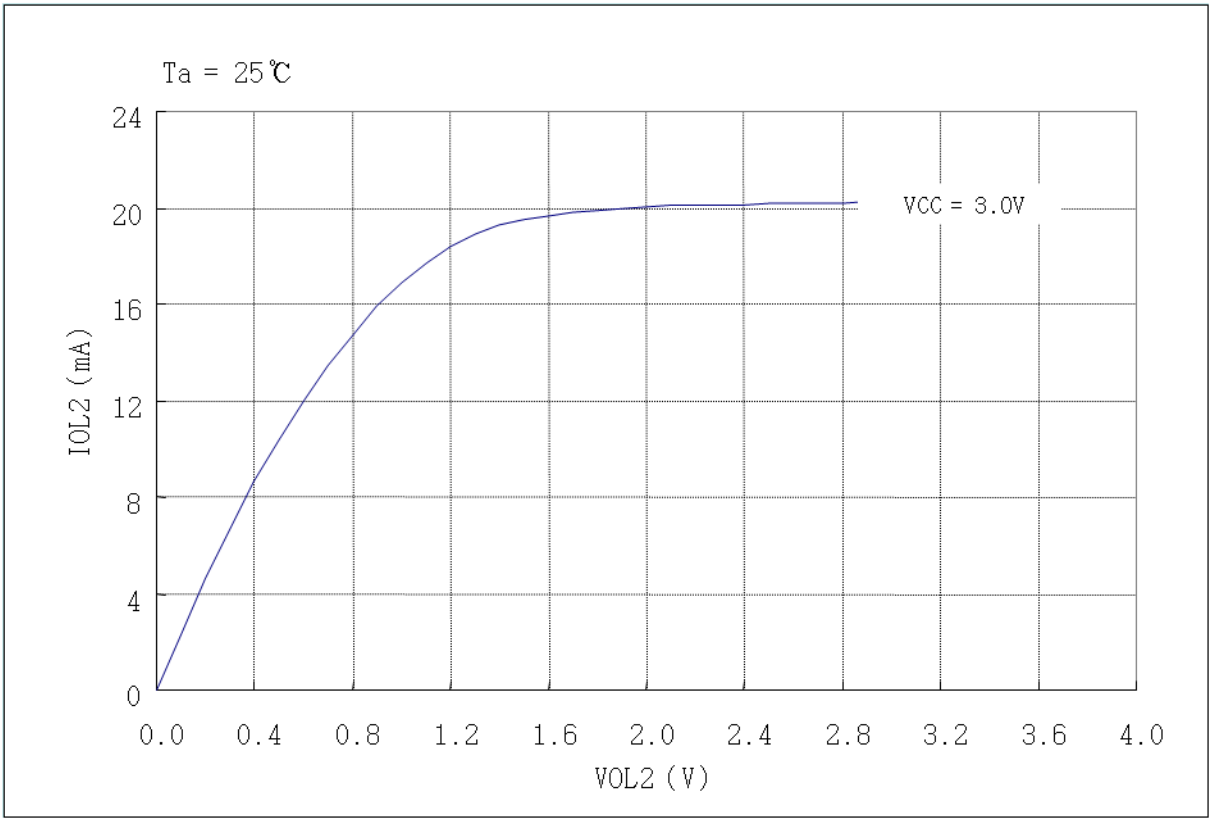


Fig 7-1. I_{OL2} vs. V_{OL2} Graph. (D, R Port)

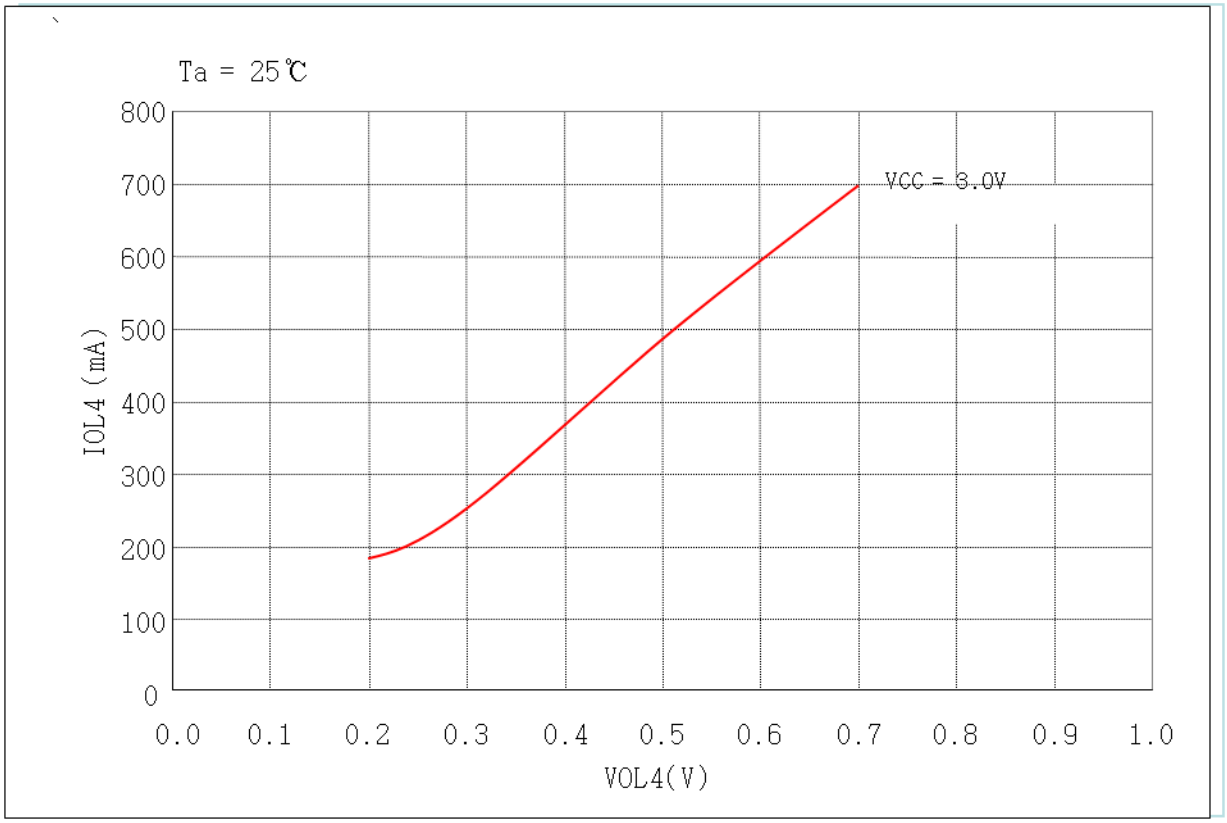


Fig 7-2. I_{OL4} vs. V_{OL4} Graph (REMOUT Port with built-in Transistor of MC40P5004,MC40P5204 and MC40P5404)

8. Architecture

8.1 Program Memory (EPROM)

The MC40P5x04 series can incorporate maximum 4,096 words (4 bank x 64 words x 16 page x 8bits) for program memory. Program counter PC (A0~A5), Page address register (A6~A9) and Bank address register (A10, A11) are used to address the whole area of program memory having an instruction (8bits) to be next executed.

The program memory consists of 64 words on each page, and thus each page can hold up to 64 steps of instructions.

The program memory is composed as shown below

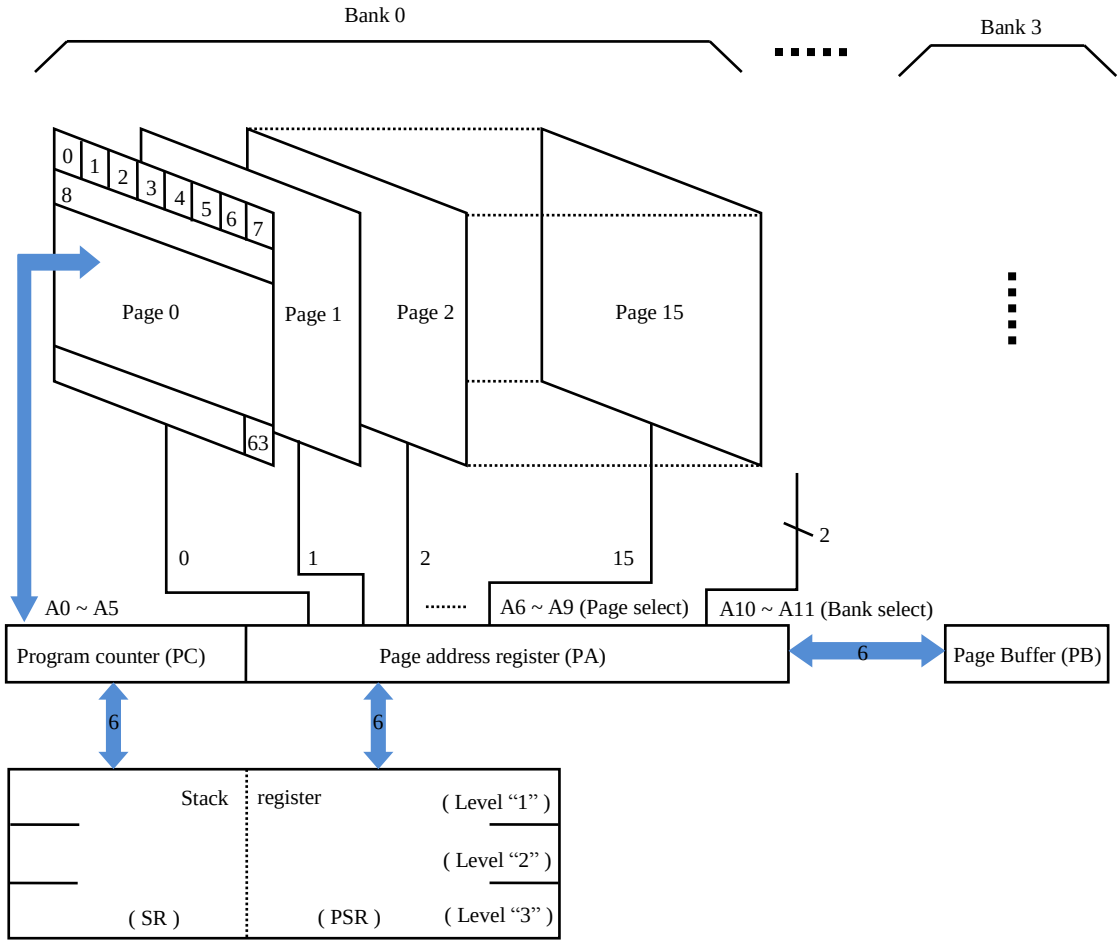


Fig 8-1 Configuration of Program Memory

8.2 EPROM Address Register

The following registers are used to address the EPROM.

- Page address register (PA)

  Holds EPROM's page number (0~Fh) and bank address (0 ~ 3h)to be addressed.

- Page buffer register (PB)

  Value of PB is loaded by an LPBI command when newly addressing a page. Then it is shifted into the PA when rightly executing a branch instruction (BR) and a subroutine call (CAL).

  When addressing more than 1K, LPBI command must be used continuously. First LPBI value will be written at LSB 4bits, second is written at MSB 2bits.

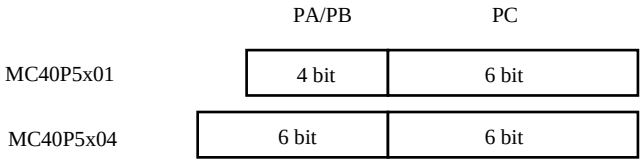


Fig 8-2 Compare MC40P5x01 with MC40P5x04

※ Example of command flow

Command	PA	PB	PC
⋮	XX_XXXX	XX_XXXX	pc
LPBI #7 (high 2bits 00, low 4bits #7)	XX_XXXX	00_0111	Next pc
LPBI #3 (high 2bits #3, low 4bits keep)	XX_XXXX	11_0111	Next pc
BR #2A	11_0111	11_0111	#2A

- Program counter (PC)

  Available for addressing word on each page.

- Stack register (SR)

  Stores returned-word address in the subroutine call mode.

(1) Page address register and page buffer register

Address one of pages #0 to #15 in the EPROM by the 4-bit binary counter. Unlike the program counter, the page address register is usually unchanged so that the program will repeat on the same page unless a page changing command is issued. To change the page address, take two steps such as (1) writing in the page buffer what page to jump (execution of LPBI) and (2) execution of BR or CAL, because instruction code is of eight

bits so that page and word can not be specified at the same time.

In case a return instruction (RTN) is executed within the subroutine that has been called In the other page, the page address will be changed at the same time.

(2) Program counter

This 6-bit binary counter increments for each fetch to address a word in the currently addressed page having an instruction to be next executed. For easier programming, at turning on the power, the program counter is reset to the zero location. The PA is also set to "0". Then the program counter specifies the next EPROM address in random sequence. When BR, CAL or RTN instructions are decoded, the switches on each step are turned off not to update the address. Then, for BR or CAL, address data are taken in from the instruction operands (a0 to a5), or for RTN, and address is fetched from stack register No. 1.

(3) Stack register

This stack register provides two stages each for the program counter (6bits) and the page address register (4bits) so that subroutine nesting can be made on two levels.

8.3 Data memory (RAM)

Up to 32 nibbles (16 words x 2pages x 4bits) is incorporated for storing data. The whole data memory area is indirectly specified by a data pointer (X, Y). Page number is specified by zero bit of X register, and words in the page by 4 bits in Y-register. Data memory is composed in 16 nibbles/page. Figure 8-3 shows the configuration.

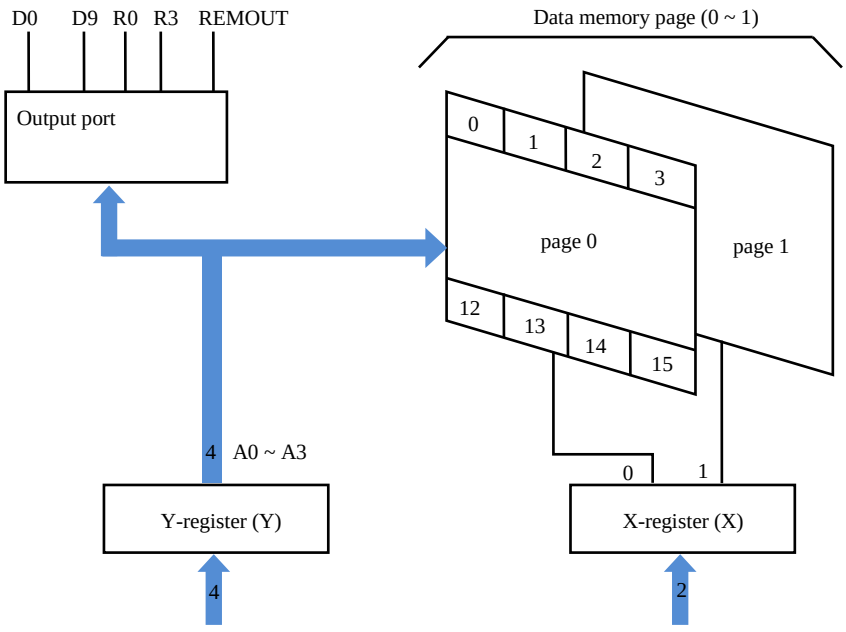


Fig 8-3 Configuration of Data Memory

8.4 X-register (X)

X-register is consist of 2bit, X0 is a data pointer of page in the RAM, X1 is only used for selecting of D8~D9 with value of Y-register

	X1 = 0 (X=0 or 1)	X1 = 1 (X=2 or 3)
Y = 0	D0	D8
Y = 1	D1	D9

Table 8-1 Mapping table between X and Y register for port access

8.5 Y-register (Y)

Y-register has 4 bits. It operates as a data pointer or a general-purpose register.

Y-register specifies and address (a0~a3) in a page of data memory, as well as it is used to specify an output port. Further it is used to specify a mode of carrier signal outputted from the REMOUT port. It can also be treated as a general-purpose register on a program.

8.6 Accumulator (Acc)

The 4-bit register for holding data and calculation results.

8.7 Arithmetic and Logic Unit (ALU)

In this unit, 4bits of adder/comparator are connected in parallel as it's main components and they are combined with status latch and status logic (flag.)

(1) Operation circuit (ALU)

The adder/comparator serves fundamentally for full addition and data comparison. It executes subtraction by making a complement by processing an inversed output of Acc (Acc +1)

(2) Status logic

This is to bring an ST, or flag to control the flow of a program. It occurs when a specified instruction is executed in three cases such as overflow or underflow in operation and two inputs unequal.

8.8 State Counter (SC)

A fundamental machine cycle timing chart is shown below. Every instruction is one byte length. Its execution time is the same. Execution of one instruction takes 48 clocks for fetch cycle and 48clocks for execute cycle (96 clocks in total).

Virtually these two cycles proceed simultaneously, and thus it is apparently completed in 48clocks (one machine cycle). Exceptionally BR, CAL and RTN instructions is normal execution time since they change an addressing sequentially. Therefore, the next instruction is prefetched so that its execution is completed within the fetch cycle.

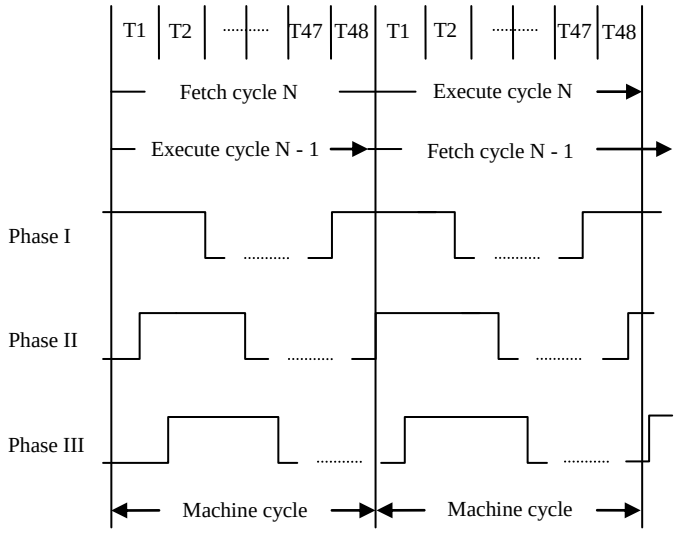


Fig 8-4 Fundamental timing chart

8.9 Clock Generator

The oscillator circuit is designed to operate with an external ceramic resonator. Oscillator circuit is able to organize by connecting ceramic resonator to outside.

* It is necessary to connect capacitor to outside in order to change ceramic resonator, you must refer to a manufacturer`s resonator matching guide.

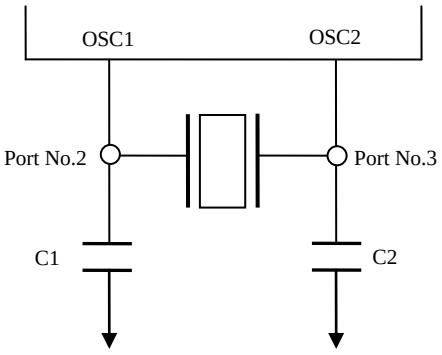
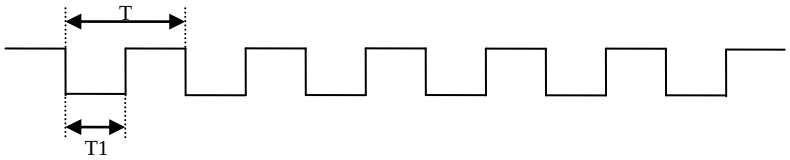


Fig 8-5 Oscillator circuit with external capacitor

8.10 Pulse Generator

The following frequency and duty ratio are selected for carrier signal outputted from the REMOUT port depending on a PMR (Pulse Mode Register) value set in a program.



REMOUT port

PMR	REMOUT signal	Carrier frequency (f_{OSC} = 3.64MHz)
0	$T=1/f_{PUL}$ = 96/ f_{OSC}, T1/T = 1/2	37.92KHz
1	$T=1/f_{PUL}$ = 96/ f_{OSC}, T1/T = 1/3	37.92KHz
2	$T=1/f_{PUL}$ = 64/ f_{OSC}, T1/T = 1/2	56.88KHz
3	$T=1/f_{PUL}$ = 65/ f_{OSC}, T1/T = 22/65	56.00KHz
4	$T=1/f_{PUL}$ = 87/ f_{OSC}, T1/T = 1/3	41.84KHz
5	No Carrier (same to inversion of D0~D9)	-
6	$T=1/f_{PUL}$ = 91/ f_{OSC}, T1/T =31/91	40.00KHz
7	$T= 1/f_{PUL}$ = 101/ f_{OSC}, T1/T = 34/101	36.04KHz

*Default value is “0”

* f_{PUL} = Carrier Pulse frequency, f_{OSC} = Oscillation frequency

Table 8-2 PMR selection table

8.11 Reset Operation

MC40P5x04 series have three reset sources. One is a built-in Power-on reset circuit,
Another is a built-in Low VDD Detection circuit, the other is the overflow of Watch Dog Timer WDT). All reset operations are internal in the MC40P5x04 series.

8.11.1 Built-in Power On Reset Circuit

MC40P5x04 series has a built-in Power-on reset circuit consisting of an about 1 MΩ Resistor and a 3pF Capacitor. When the Power-on reset pulse occurs, system reset signal is latched and WDT is cleared. After the overflow time of WDT(2^{13} x System clock time), system reset signal is released.

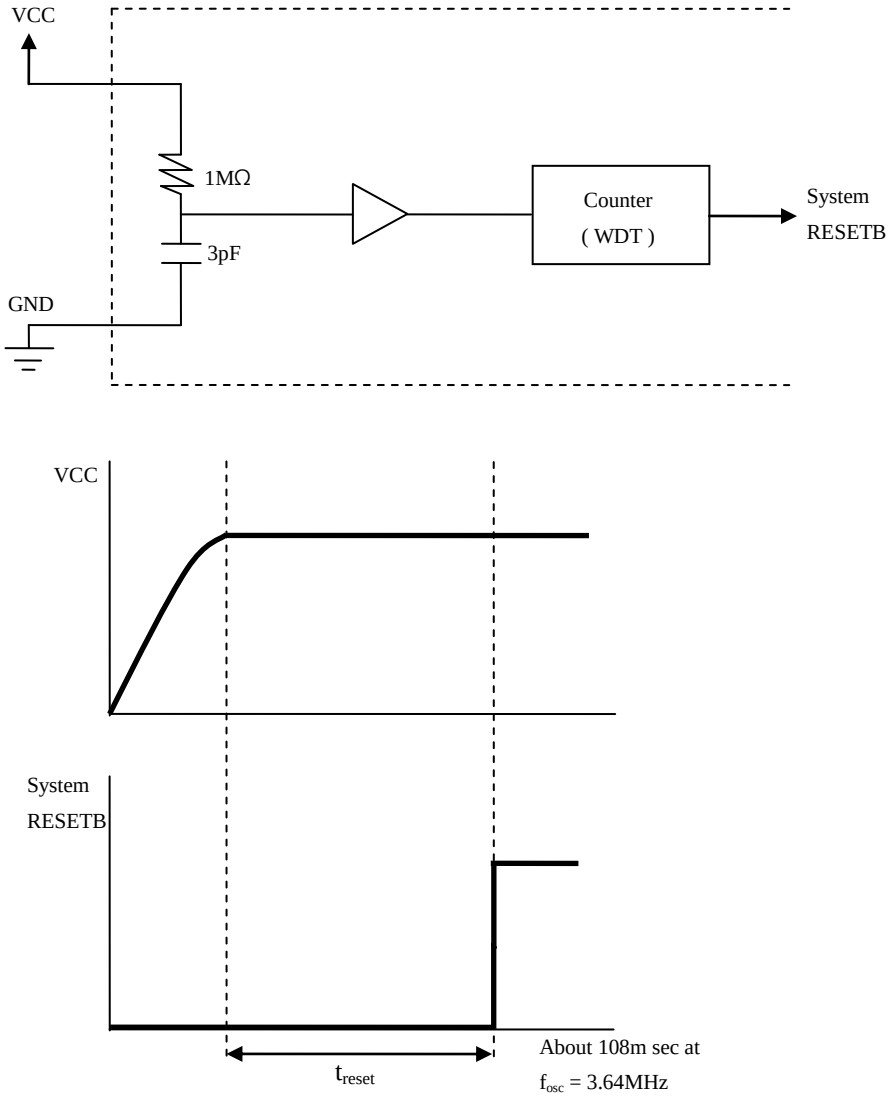


Fig 8-6 Power –On Reset Circuit and Timing Chart

8.11.2 Built-in Low VDD Reset Circuit

MC40P5x04 series have a Low VDD detection circuit.

If VDD become Reset Voltage of Low VDD Detection circuit at a active status, system reset occur and WDT is cleared.

After VDD is increased upper Reset Voltage again, WDT is re-counted and if WDT is overflowed, system reset is released.

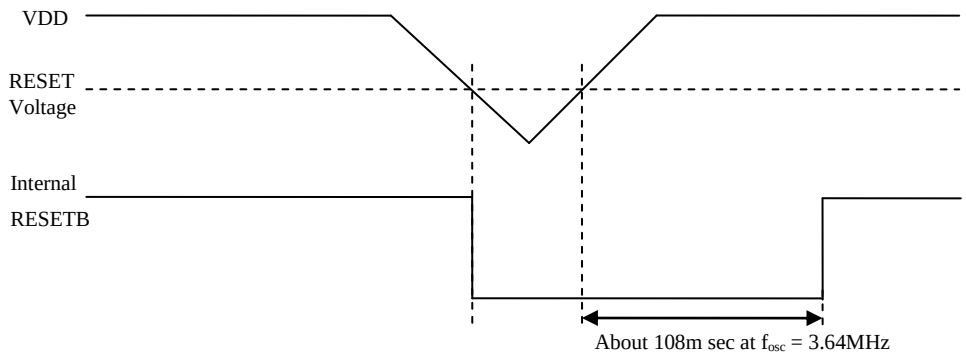


Fig 8-7 Low Voltage Detection Timing Chart

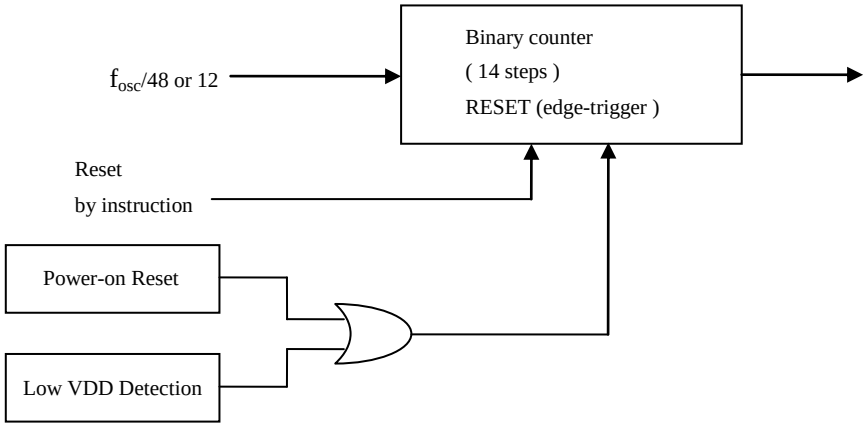
8.11.3 Watch Dog Timer (WDT)

Watch dog timer is organized binary of 14 steps. The signal of fosc /48 or fosc/12 cycle comes in the first step of WDT after WDT reset. If this counter was overflowed, reset signal automatically come out so that internal circuit is initialized.

The overflow time is 8 x 6 x 2^{13} /fosc (108.026ms at fosc = 3.64MHz)

Normally, the binary counter must be reset before the overflow by using reset instruction (WDTR), Power-on reset pulse or Low VDD detection pulse.

* It is constantly reset in STOP mode. When STOP is released, counting is restarted. (Refer to STOP operation)



8.12 STOP Operation

Stop mode can be achieved by STOP instructions.

In stop mode

1. Oscillator is stopped, the operating current is low.
2. Watch dog timer is reset, D0~D3 and D8~D9 output are “L”, and REMOUT output are High-Z.
3. Part of output pin other than WDT, D0~D3, D8~D9 output and REMOUT output have a value before come into stop mode.

Stop mode is released when one of K or R input is going to "L".

1. State of D0~D7 output and REMOUT output is return to state of before stop mode is achieved.
2. After 1,024 x 8 enable clocks for stable oscillating, First instruction start to operate.
3. In return to normal operation, WDT is counted from zero again.

But, at executing stop instruction, if one of K or R input is chosen to "L", stop instruction is same to NOP (No Operation) instruction.

8.13 Port Operation

Port operation is defined by value of X,Y register

Value of X-reg	Value of Y-reg	Operation
0 or 1	0 ~ 7	SO : D(Y) ← 1, RO : D(Y) ← 0
0 or 1	8	REMOUT port repeats "H" and "L" in pulse frequency. (When PMR = 5, it is fixed at "H" or “L”) SO : REMOUT ← 0 RO : REMOUT ← 1 (High-Z)
0 or 1	9	SO : D0 ~ D9 ← 1 (High-Z) RO : D0 ~ D9 ← 0
0 or 1	A ~ D	SO : R(Y-Ah) ← 1 RO : R(Y-Ah) ← 0
0 or 1	E	SO : R0 ~ R3 ← 1 RO : R0 ~ R3 ← 0
0 or 1	F	SO : D0 ~ D9 ← 1, R0 ~ R3 ← 1 RO : D0 ~ D9 ← 0, R0 ~ R3 ← 0
2 or 3	0	SO : D(8) ← 1 RO : D(8) ← 0
2 or 3	1	SO : D(9) ← 1 RO : D(9) ← 0

9. Instruction

INSTRUCTION FORMAT

All of the 43 instruction in MC40P5x04 series is format in two fields of OP code and operand which consist of eight bits. The following formats are available with different types of operands.

*Format Ⅰ

All eight bits are for OP code without operand.

*FormatⅡ

Two bits are for operand and six bits for OP code. Two bits of operand are used for specifying bits of RAM and X-register (bit 1 and bit 7 are fixed at ″0″)

*FormatⅢ

Four bits are for operand and the others are OP code. Four bits of operand are used for specifying a constant loaded in RAM or Y-register, a comparison value of compare command, or page addressing in ROM.

*Format Ⅳ

Six bits are for operand and the others are OP code. Six bits of operand are used for word addressing in the ROM.

9.1 Instruction Table

The MC40P5x04 series provides the following 43 basic instructions.

	Category	Mnemonic	Function	ST[*1]
1	Register to Register	LAY	A ← Y	S
2		LYA	Y ← A	S
3		LAZ	A ← 0	S
4	RAM to Register	LMA	M(X, Y) ← A	S
5		LMAIY	M(X, Y) ← A, Y ← Y + 1	S
6		LYM	Y ← M(X, Y)	S
7		LAM	A ← M(X, Y)	S
8		XMA	A ↔ M(X, Y)	S
9	Immediate	LYI i	Y ← i	S
10		LMIIY i	M(X, Y) ← i, Y ← Y + 1,	S
11		LXI n	X ← n	S
12	RAM Bit Manipulation	SEM n	M(n) ← 1	S
13		REM n	M(n) ← 0	S
14		TM n	TEST M(n) ← 1	E
15	ROM Address	BR a	if ST = 1 then Branch	S
16		CAL a	if ST = 1 then Subroutine call	S
17		RTN	Return from Subroutine	S
18		LPBI i	PB ← i (PB3~0 or PB5~4 loading) [*3]	S
19	Arithmetic	AM	A ← A + M(X, Y)	C
20		SM	A ← M(X, Y) – A	B
21		IM	A ← M(X, Y) + 1	C
22		DM	A ← M(X, Y) - 1	B
23		IA	A ← A + 1	S
24		IY	Y ← Y + 1	C

25	Arithmetic	DA	A ← A - 1	B
26		DY	Y ← Y - 1	B
27		EORM	A ← A ⊕ M(X, Y)	S
28		NEGA	A ← $\overline{A}$ + 1	Z
29	Comparison	ALEM	TEST A ≤ M(X, Y)	E
30		ALEI i	TEST A ≤ i	E
31		MNEZ	TEST M(X, Y) ≠ 0	N
32		YNEA	TEST Y ≠ A	N
33		YNEI i	TEST Y ≠ i	N
34		KNEZ	TEST K ≠ 0	N
35		RNEZ	TEST R ≠ 0	N
36	Input/ Output	LAK	A ← K	S
37		LAR	A ← R	S
38		SO	Output(Y) ← 1^{*2}	S
39		RO	Output(Y) ← 0^{*2}	S
40	Control	WDTR	Watch Dog Timer Reset	S
41		STOP	Stop operation	S
42		LPY	PMR ← Y	S
43		NOP	No operation	S

Note) i = 0~f, n = 0~3, a = 6bit PC Address

*1 Column ST indicates conditions for changing status. Symbols have the following meanings

S: On executing an instruction, status is unconditionally set.

C: Status is only set when carry or borrow has occurred in operation.

B: Status is only set when borrow has not occurred in operation.

E: Status is only set when equality is found in comparison.

N: Status is only set when equality is not found in comparison.

Z: Status is only set when the result is zero.

*2 Operation is settled by a value of Y-register..

*3 LPBI instruction loads data to PB3~0 or PB5~4 according to position.

Refer to ABL and ACALL macro.

9.2 DETAILS OF INSTRUCTION SYSTEM

All 43 basic instructions of the MC40P5X04 Series are one by one described in detail below.

Description Form.

Each instruction is headlined with its mnemonic symbol according to the instructions table given earlier. Then, for quick reference, it is described with basic items as shown below. After that, detailed comment follows.

*Items :

- Naming : Full spelling of mnemonic symbol
- Status : Check of status function
- Format : Categorized into Ⅰ to Ⅳ
- Operand : Omitted for Format Ⅰ
- Function

(1) LAY

Naming : Load Accumulator from Y-Register

Status : Set

Format : I

Function : A ← Y

<Comment> Data of four bits in the Y-register is unconditionally transferred to the accumulator. Data in the Y-register is left unchanged.

(2) LYA

Naming : Load Y-register from Accumulator

Status : Set

Format : I

Function : Y ← A

<Comment> Load Y-register from Accumulator

(3) LAZ

Naming : Clear Accumulator

Status : Set

Format : I

Function : A ← 0

<Comment> Data in the accumulator is unconditionally reset to zero.

(4) LMA

Naming : Load Memory from Accumulator

Status : Set

Format : I

Function : M(X,Y) ← A

<Comment> Data of four bits from the accumulator is stored in the RAM location addressed by the X-register and Y-register. Such data is left unchanged.

(5) LMAIY

Naming : Load Memory from Accumulator and Increment Y-Register

Status : Set

Format : I

Function : M(X,Y) ← A, Y ← Y+1

<Comment> Data of four bits from the accumulator is stored in the RAM location addressed by the X register and Y-register. Such data is left unchanged.

(6) LYM

Naming : Load Y-Register form Memory

Status : Set

Format : I

Function : Y ← M(X,Y)

<Comment> Data from the RAM location addressed by the X-register and Y-register is loaded into the Y-register. Data in the memory is left unchanged.

(7) LAM

Naming : Load Accumulator from Memory

Status : Set

Format : I

Function : A ← M(X,Y)

<Comment> Data from the RAM location addressed by the X-register and Y-register is loaded into the Y-register. Data in the memory is left unchanged.

(8) XMA

Naming : Exchanged Memory and Accumulator

Status : Set

Format : I

Function : M(X,Y) ↔ A

<Comment> Data from the memory addressed by X-register and Y-register is exchanged with data from the accumulator. For example, this instruction is useful to fetch a memory word into the accumulator for operation and store current data from the accumulator into the RAM. The accumulator can be restored by another XMA instruction.

(9) LYI i

Naming : Load Y-Register from Immediate

Status : Set

Format : Ⅲ

Operand : Constant 0 ≤ i ≤ 15

Function : Y ← i

<Purpose> To load a constant in Y-register. It is typically used to specify Y-register in a particular RAM word address, to specify the address of a selected output line, to set Y-register for specifying a carrier signal outputted from OUT port, and to initialize Y-register for loop control. The accumulator can be restored by another XMA instruction.

<Comment> Data of four bits from operand of instruction is transferred to the Y-register.

(10) LMIIY i

Naming : Load Memory from Immediate and Increment Y-Register

Status : Set

Format : Ⅲ

Operand : Constant 0 ≤ i ≤ 15

Function : M(X,Y) ← i, Y ← Y + 1

<Comment> Data of four bits from operand of instruction is stored into the RAM location addressed by the X-register and Y-register. Then data in the Y-register is incremented by one.

(11) LXI n

Naming : Load X-Register from Immediate

Status : Set

Format : Ⅱ

Operand : X file address 0 ≤ n ≤ 3

Function : X ← n

<Comment> A constant is loaded in X-register. It is used to set X-register in an index of desired RAM page. Operand of 1 bit of command is loaded in X-register.

(12) SEM n

Naming : Set Memory Bit

Status : Set

Format : Ⅱ

Operand : Bit address 0 ≤ n ≤ 3

Function : M(X,Y,n) ← 1

<Comment> Depending on the selection in operand of operand, one of four bits is set as logic 1 in the RAM memory addressed in accordance with the data of the X-register and Y-register.

(13) REM n

Naming : Reset Memory Bit

Status : Set

Format : Ⅱ

Operand : Bit address 0 ≤ n ≤ 3

Function : M(X,Y,n) ← 0

<Comment> Depending on the selection in operand of operand, one of four bits is set as logic 0 in the RAM memory addressed in accordance with the data of the X-register and Y-register.

(14) TM n

Naming : Test Memory Bit

Status : Comparison results to status

Format : Ⅱ

Operand : Bit address 0 ≤ n ≤ 3

Function : M(X,Y,n) = 1 ?

ST ← 1 when M(X,Y,n)=1, ST ← 0 when M(X,Y,n)=0

<Purpose> A test is made to find if the selected memory bit is logic. 1 Status is set depending on the result.

(15) BR a

Naming : Branch on status 1

Status : Conditional depending on the status

Format : Ⅳ

Operand : Branch address a (Addr)

Function : When ST =1 , PA ← PB, PC ← a(Addr)

When ST = 0, PC ← PC + 1, ST ← 1

Note : PC indicates the next address in a fixed sequence that is actually pseudo-random count.

<Purpose> For some programs, normal sequential program execution can be change. A branch is conditionally implemented depending on the status of results obtained by executing the previous instruction.

<Comment>

- Branch instruction is always conditional depending on the status.
  a. If the status is reset (logic 0), a branch instruction is not rightly executed but the next instruction of the sequence is executed.
  b. If the status is set (logic 1), a branch instruction is executed as follows.
- Branch is available in two types - short and long. The former is for addressing in the current page and the latter for addressing in the other page. Which type of branch to execute is decided according to the PB register. To execute a long branch, data of the PB register should in advance be modified to a desired page address through the LPBI instruction.

(16) CAL a

Naming : Subroutine Call on status 1

Status : Conditional depending on the status

Format : Ⅳ

Operand : Subroutine code address a(Addr)

Function :

When ST =1 ,
PC ← a(Addr)
PA ← PB
SR1 ← PC + 1,
PSR1 ← PA
SR2 ← SR1
PSR2 ← PSR1
SR3 ← SR2
PSR3 ← PSR2

When ST = 0,
PC ← PC + 1
PB ← PS
ST ← 1

Note : PC actually has pseudo-random count against the next instruction.

<Comment>

• In a program, control is allowed to be transferred to a mutual subroutine. Since a call instruction preserves the return address, it is possible to call the subroutine from different locations in a program, and the subroutine can return control accurately to the address that is preserved by the use of the call return instruction (RTN).
Such calling is always conditional depending on the status.
a. If the status is reset, call is not executed.
b. If the status is set, call is rightly executed.
The subroutine stack (SR) of three levels enables a subroutine to be manipulated on three levels. Besides, a long call (to call another page) can be executed on any level.

• For a long call, an LPBI instruction should be executed before the CAL. When LPBI is omitted (and when PA=PB), a short call (calling in the same page) is executed.

(17) RTN

Naming : Return from Subroutine

Status : Set

Format : Ⅰ

Function :

PC ← SR1
PA, PB ← PSR1
SR1 ← SR2
PSR1 ← PSR2
SR2 ← SR3
PSR2 ← PSR3
SR3 ← SR3
PSR3 ← PSR2
ST ← 1

<Purpose> Control is returned from the called subroutine to the calling program.

<Comment> Control is returned to its home routine by transferring to the PC the data of the return address that has been saved in the stack register (SR1). At the same time, data of the page stack register (PSR1) is transferred to the PA and PB.

(18) LPBI i

Naming : Load Page Buffer Register from Immediate

Status : Set

Format : Ⅲ

Operand : ROM page address 0 ≤ i ≤ 15

Function : PB ← i

<Purpose> A new ROM page address is loaded into the page buffer register (PB). This loading is necessary for a long branch or call instruction. Refer to ABL and ACALL macro.

<Comment> The PB register is loaded together with three bits from 4 bit operand.

(19) AM

Naming : Add Accumulator to Memory and Status 1 on Carry

Status : Carry to status

Format : Ⅰ

Function : A ← M(X,Y)+A, ST ← 1(when total>15),

ST ← 0 (when total ≤15)

<Comment> Data in the memory location addressed by the X and Y-register is added to data of the accumulator. Results are stored in the accumulator. Carry data as results is transferred to status. When the total is more than 15, a carry is caused to put ″1″ in the status. Data in the memory is not changed.

(20) SM

Naming : Subtract Accumulator to Memory and Status 1 Not Borrow

Status : Carry to status

Format : Ⅰ

Function : A ← M(X,Y) - A ST ← 1(when A ≤ M(X,Y))

ST ← 0(when A > M(X,Y))

<Comment> Data of the accumulator is, through a 2`s complemental addition, subtracted from the memory word addressed by the Y-register. Results are stored in the accumulator. If data of the accumulator is less than or equal to the memory word, the status is set to indicate that a borrow is not caused. If more than the memory word, a borrow occurs to reset the status to ″0″.

(21) IM

Naming : Increment Memory and Status 1 on Carry

Status : Carry to status

Format : Ⅰ

Function : A ← M(X,Y) + 1 ST ← 1(when M(X,Y) ≥ 15)

ST ← 0(when M(X,Y) < 15)

<Comment> Data of the memory addressed by the X and Y-register fetched. Adding 1 to this word, results are stored in the accumulator. Carry data as results is transferred to the status. When the total is more than 15, the status is set. The memory is left unchanged.

(22) DM

Naming : Decrement Memory and Status 1 on Not Borrow

Status : Carry to status

Format : Ⅰ

Function : A ← M(X,Y) - 1 ST ← 1(when M(X,Y) ≥1)

ST ← 0 (when M(X,Y) = 0)

<Comment> Data of the memory addressed by the X and Y-register is fetched, and one is subtracted from this word (addition of Fh)> Results are stored in the accumulator. Carry data as results is transferred to the status. If the data is more than or equal to one, the status is set to indicate that no borrow is caused. The memory is left unchanged.

(23) IA

Naming : Increment Accumulator

Status : Set

Format : Ⅰ

Function : A ← A+1

<Comment> Data of the accumulator is incremented by one. Results are returned to the accumulator. A carry is not allowed to have effect upon the status.

(24) IY

Naming : Increment Y-Register and Status 1 on Carry

Status : Carry to status

Format : Ⅰ

Function : Y ← Y + 1 ST ← 1 (when Y = 15)

ST ← 0 (when Y < 15)

<Comment> Data of the Y-register is incremented by one and results are returned to the Y-register. Carry data as results is transferred to the status. When the total is more than 15, the status is set.

(25) DA

Naming : Decrement Accumulator and Status 1 on Borrow

Status : Carry to status

Format : Ⅰ

Function : A ← A - 1 ST ← 1(when A ≥1)

ST ← 0 (when A = 0)

<Comment> Data of the accumulator is decremented by one. As a result (by addition of Fh), if a orrow is caused, the status is reset to ″0″ by logic. If the data is more than one, no borrow occurs and thus the status is set to ″1″.

(26) DY

Naming : Decrement Y-Register and Status 1 on Not Borrow

Status : Carry to status

Format : Ⅰ

Function : Y ← Y -1 ST ← 1 (when Y ≥ 1)

ST ← 0 (when Y = 0)

<Purpose> Data of the Y-register is decremented by one.

<Comment> Data of the Y-register is decremented by one by addition of minus 1 (Fh). Carry data as results is transferred to the status. When the results is equal to 15, the status is set to indicate that no borrow has not occurred.

(27) EORM

Naming : Exclusive or Memory and Accumulator

Status : Set

Format : Ⅰ

Function : A ← M(X,Y) ⊕ A

<Comment> Data of the accumulator is, through a Exclusive OR, subtracted from the memory word addressed by X and Y-register. Results are stored into the accumulator.

(28) NEGA

Naming : Negate Accumulator and Status 1 on Zero

Status : Carry to status

Format : Ⅰ

Function : A ← $\overline{A}$ + 1 ST ← 1(when A = 0)

ST ← 0 (when A != 0)

<Purpose> The 2`s complement of a word in the accumulator is obtained.

<Comment> The 2`s complement in the accumulator is calculated by adding one to the 1`s complement in the accumulator. Results are stored into the accumulator. Carry data is transferred to the status. When data of the accumulator is zero, a carry is caused to set the status to ″1″.

(29) ALEM

Naming : Accumulator Less Equal Memory

Status : Carry to status

Format : Ⅰ

Function : A ≤ M(X,Y) ST ← 1 (when A ≤ M(X,Y))

ST ← 0 (when A > M(X,Y))

<Comment> Data of the accumulator is, through a complemental addition, subtracted from data in the memory location addressed by the X and Y-register. Carry data obtained is transferred to the status. When the status is ″1″, it indicates that the data of the accumulator is less than or equal to the data of the memory word. Neither of those data is not changed.

(30) ALEI

Naming : Accumulator Less Equal Immediate

Status : Carry to status

Format : Ⅲ

Operand : Constant 0 ≤ i ≤ 15

Function : A ≤ i ST ← 1 (when A ≤ i)

ST ← 0 (when A > i)

<Purpose> Data of the accumulator and the constant are arithmetically compared.

<Comment> Data of the accumulator is, through a complemental addition, subtracted from the constant that exists in 4bit operand. Carry data obtained is transferred to the status. The status is set when the accumulator value is less than or equal to the constant. Data of the accumulator is left unchanged.

(31) MNEZ

Naming : Memory Not Equal Zero

Status : Comparison results to status

Format : Ⅰ

Function : M(X,Y) ≠ 0 ST ← 1(when M(X,Y) ≠ 0)

ST ← 0 (when M(X,Y) = 0)

<Purpose> A memory word is compared with zero.

<Comment> Data in the memory addressed by the X and Y-register is logically compared with zero. Comparison data is transferred to the status. Unless it is zero, the status is set.

(32) YNEA

Naming : Y-Register Not Equal Accumulator

Status : Comparison results to status

Format : Ⅰ

Function : Y ≠ A ST ← 1 (when Y ≠ A)

ST ← 0 (when Y = A)

<Purpose> Data of Y-register and accumulator are compared to check if they are not equal.

<Comment> Data of the Y-register and accumulator are logically compared. Results are transferred to the status. Unless they are equal, the status is set.

(33) YNEI

Naming : Y-Register Not Equal Immediate

Status : Comparison results to status

Format : Ⅲ

Operand : Constant 0 ≤ i ≤ 15

Function : Y ≠ i ST ← 1 (when Y ≠ i)

ST ← 0 (when Y = i)

<Comment> The constant of the Y-register is logically compared with 4bit operand. Results are transferred to the status. Unless the operand is equal to the constant, the status is set.

(34) KNEZ

Naming : K Not Equal Zero

Status : The status is set only when not equal

Format : Ⅰ

Function : When K ≠ 0, ST ← 1

<Purpose> A test is made to check if K is not zero.

<Comment> Data on K are compared with zero. Results are transferred to the status. For input data not equal to zero, the status is set.

(35) RNEZ

Naming : R Not Equal Zero

Status : The status is set only when not equal

Format : Ⅰ

Function : When R ≠ 0, ST ← 1

<Purpose> A test is made to check if R is not zero.

<Comment> Data on R are compared with zero. Results are transferred to the status. For input data not equal to zero, the status is set.

(36) LAK

Naming : Load Accumulator from K

Status : Set

Format : Ⅰ

Function : A ← K

<Comment> Data on K are transferred to the accumulator

(37) LAR

Naming : Load Accumulator from R

Status : Set

Format : Ⅰ

Function : A ← R

<Comment> Data on R are transferred to the accumulator

(38) SO

Naming : Set Output Register Latch

Status : Set

Format : Ⅰ

Function :

D(Y) ← 1	0 ≤ Y ≤ 7
REMOUT(PMR=5) ← 0	Y = 8
D0~D9 ← 1 (High-Z)	Y = 9
R(Y) ← 1	Ah ≤ Y ≤ Dh
R ← 1	Y = Eh
D0~D9, R ← 1	Y = Fh

<Purpose> A single D output line is set to logic 1, if data of Y-register is between 0 to 7. Carrier frequency comes out from REMOUT port, if data of Y-register is 8.

All D output line is set to logic 1, if data of Y-register is 9. It is no operation, if data of Y-register between 10 to 15.

When Y is between Ah and Dh, one of R output lines is set at logic 1.

When Y is Eh, the output of R is set at logic 1.

When Y is Fh, the output D0~D9 and R are set at logic 1.

<Comment> Data of Y-register is between 0 to 7, selects appropriate D output.

Data of Y-register is 8, selects REMOUT port.

Data of Y-register is 9, selects all D port.

Data in Y-register, when between Ah and Dh, selects an appropriate R output (R0~R3).

Data in Y-register, when it is Eh, selects all of R0~R3.

Data in Y-register, when it is Fh, selects all of D0~D9 and R0~R3.

(39) RO

Naming : Reset Output Register Latch

Status : Set

Format : Ⅰ

Function :

D(Y) ← 0	0 ≤ Y ≤ 7
REMOUT ← 1(High-Z)	Y = 8
D0~D9 ← 0	Y = 9
R(Y) ← 0	Ah ≤ Y ≤ Dh
R ← 0	Y = Eh
D0~D9, R ← 0	Y = Fh

<Purpose> A single D output line is set to logic 0, if data of Y-register is between 0 to 9.
REMOUT port is set to logic 0, if data of Y-register is 9.
All D output line is set to logic 0, if data of Y-register is 9.
When Y is between Ah and Dh, one of R output lines is set at logic 0.
When Y is Eh, the output of R is set at logic 0
When Y is Fh, the output D0~D9 and R are set at logic 1.

<Comment> Data of Y-register is between 0 to 7, selects appropriate D output.
Data of Y-register is 8, selects REMOUT port.
Data of Y-register is 9, selects D port.
Data in Y-register, when between Ah and Dh, selects an appropriate R output (R0~R3).
Data in Y-register, when it is Eh, selects all of R0~R3.
Data in Y-register, when it is Fh, selects all of D0~D9 and R0~R3.

(40) WDTR

Naming : Watch Dog Timer Reset

Status : Set

Format : Ⅰ

Function : Reset Watch Dog Timer (WDT)

<Purpose> Normally, you should reset this counter before overflowed counter for dc watch dog timer. this instruction controls this reset signal.

(41) STOP

Naming : STOP

Status : Set

Format : Ⅰ

Function : Operate the stop function

<Purpose> Stopped oscillator, and little current.
(See STOP function.)

(42) LPY

Naming : Pulse Mode Set

Status : Set

Format : Ⅰ

Function : PMR ← Y

<Comment> Selects a pulse signal outputted from REMOUT port.

(43) NOP

Naming : No Operation

Status : Set

Format : Ⅰ

Function : No operation

9.3 Assembler Macro

(44) CALL a (2, 3byte) : Long_call Macro

Page call within Bank (2byte) :

```
LPBI i     ; i = low_page address(4bits), PB3~0(low_page address) <-- i
CAL a      ; see you "CAL" instruction.
```

Page call out of Bank (3byte) :

```
LPBI i     ; i = low_page address(4bits), PB3~0(low_page address) <-- i
LPBI i     ; i = high_page address(2bits), PB5~4(high_page address) <-- i
CAL a      ; see you "CAL" instruction.
```

(45) BL a (2, 3byte) : Long_branch Macro

Page branch within Bank (2byte) :

```
LPBI i     ; i = low_ page address(4bits), PB0~3(low_page address) <-- i
BR a       ; see you "BR" instruction.
```

Page branch out of Bank (3byte) :

```
LPBI i     ; i = low_page address(4bits), PB3~0(low_page address) <-- i
LPBI i     ; i = high_page address(2bits), PB5~4(high_page address) <-- i
BR a       ; see you "BR" instruction.
```

(46) ACALL a (3byte) : Absolute call Macro

Full-range ROM address call (3byte) :

```
LPBI i     ; i = low_page address(4bits), PB3~0(low_page address) <-- i
LPBI i     ; i = high_page address(2bits), PB5~4(high_page address) <-- i
CAL a      ; see you "CAL" instruction.
```

(47) ABL a (3byte) : Absolute branch Macro

Full-range ROM address branch (3byte) :

```
LPBI i     ; i = low_page address(4bits), PB3~0(low_page address) <-- i
LPBI i     ; i = high_page address(2bits), PB5~4(high_page address) <-- i
BR a       ; see you "BR" instruction.
```

* a(address) : 6bits, i (low_page address) : 4bits, i (high_page address) : 2bits
  Bank(16pages unit)

10. SPGM(Serial Program)

Summary of Protocol

※ The I^2C Bus Protocol

The I^2C bus protocol is a method of communication. It physically consists of 2 active wires. The active wires, called SCL and SDA, are both bi-directional. SCL is the Serial Clock line. It is used to synchronize all data transfers over the I^2C bus. and SDA is the Serial Data line. The SCL & SDA lines are connected to all devices on the I^2C bus.

- **Necessary pins (5pins)**
  - Serial Data (SDA) : D5
  - Serial Clock (SCL) : K0
  - Programming Power(VPP) : K3
  - VDD
  - VSS

• **OPTION PROGRAM / READ DATA Format**

-	-	-	CLKSEL	OFFSET3	OFFSET2	OFFSET1	OFFSET0

CLKSEL : change the Main Clock fosc/48, fosc/12 (default fosc/48). If you select fosc/12, instruction cycle is 4 times faster than fosc/48, but Carrier Frequency isn't affected.

"0" : fosc/12

"1" : fosc/48

OFFSET : at normal mode, change the ROM address

-	-	-	-	LOCK 3	LOCK 2	LOCK 1	LOCK 0

· LOCK : when program is read by PGM-PLUS, Data Protection

ID 7	ID 6	ID 5	ID 4	ID 3	ID 2	ID 1	ID0

· ID7 – ID0 : it can be treated as User ID.

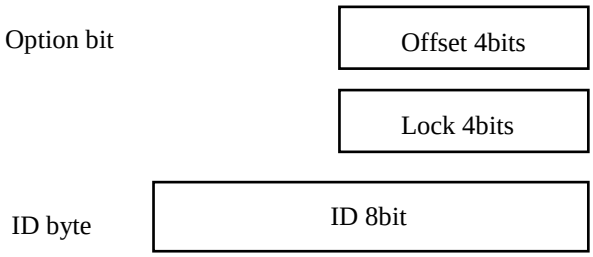
MC40P5004 ID: 1000 1111b

MC40P5204 ID: 1010 1111b

MC40P5404 ID: 1100 1111b

※ For protection the written program code, in other words it can not be read, you have to clear the Option bit to "0", and for this, you have to write the Option Register to 1111_1110b. In this time, ID7 – ID0 keep the existing value without any effect

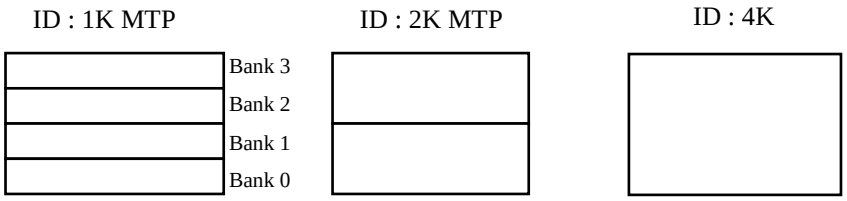
- **4K MTP (Multi Time Programming)**



Lock 4bits : Bank address (Bank 0 ~ Bank 3)
when Lock is set, 00H is read
(0 = Lock , 1 = Unlock)

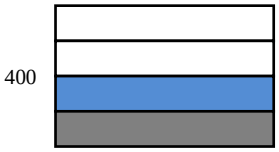
Offset bit's flow
(after write, it will be updated)

1111: initial value

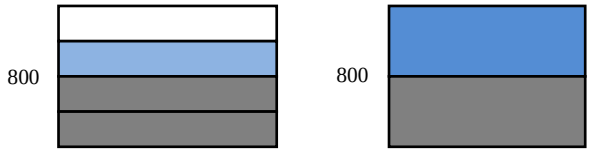


111**0** :
CPU start address is 0x000
1st write 1K
1st write 2K
1st write 4K

11**0**0 :
CPU start address is 0x400
2nd write 1K



1**0**X0 :
CPU start address is 0x800
1000: 3rd write 1K
1010: 2nd write 2K



0XX0 :
CPU start address is 0xC00
0000: 4th write 1K

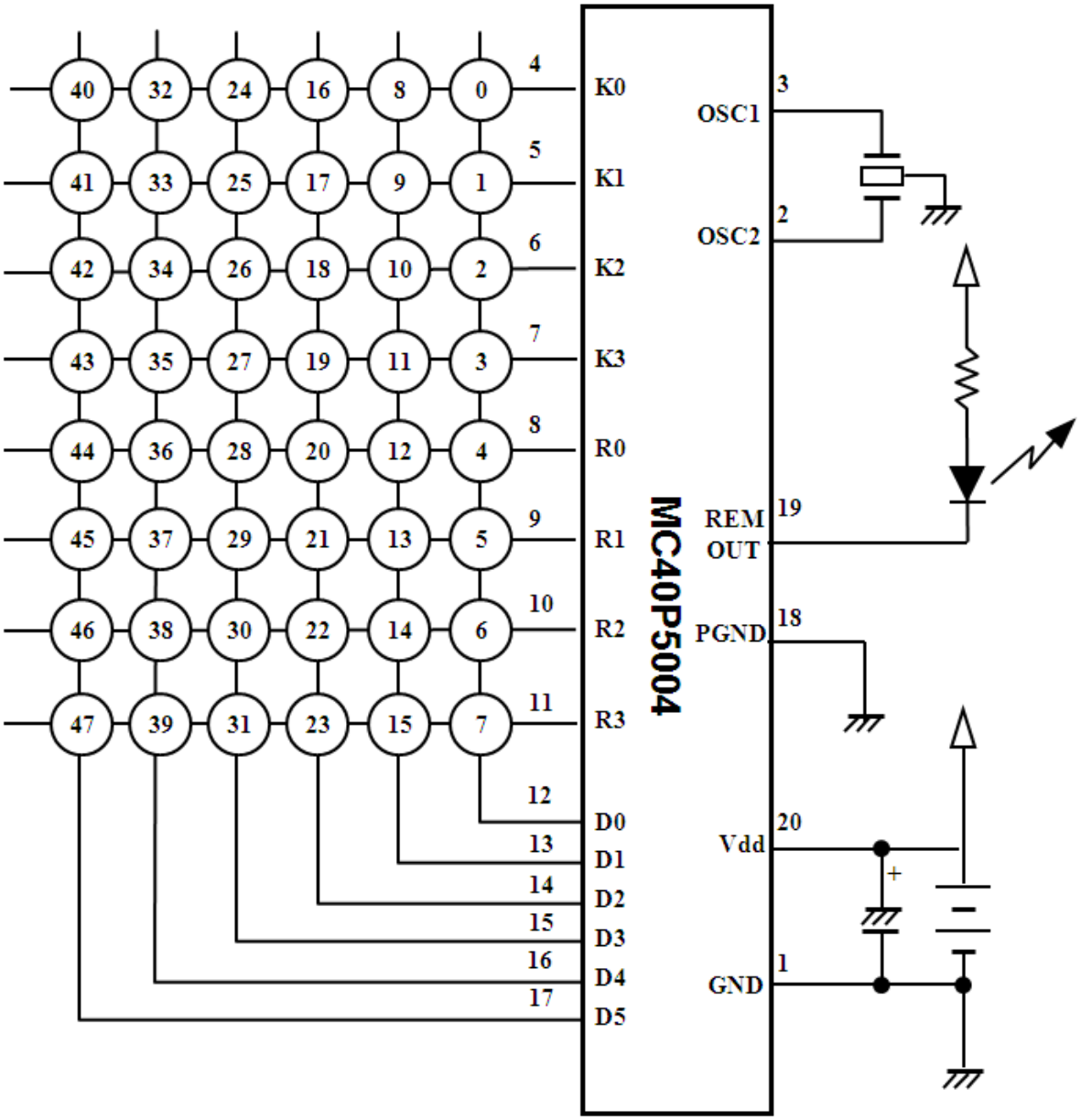
- **Case of Available MTP**

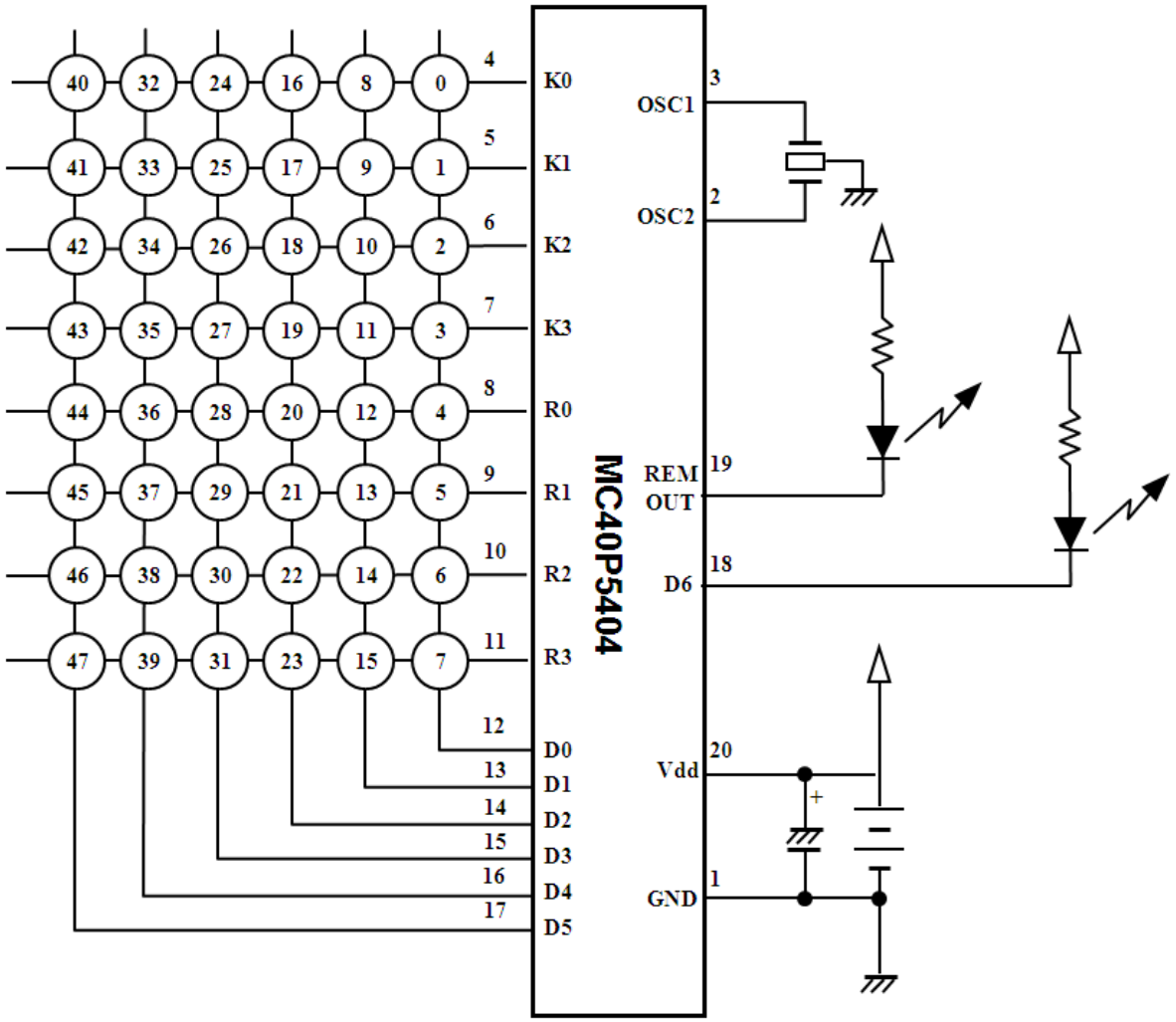
Case	1K	1K	1K	1K	Available
1K*4					O
2K*2					O
4K*1					O
1K, 1K, 2K					X
1K, 2K					O
1K, 2K, 1K					X
2K, 1K, 1K					O

	1st PGM
	2nd PGM
	3rd PGM
	4th PGM
	Expired

11. APPLICATION

11.1 Circuit Diagram of MC40P5004 and MC40P5404




40
32
24
16
8
0
41
33
25
17
9
1
42
34
26
18
10
2
43
35
27
19
11
3
44
36
28
20
12
4
45
37
29
21
13
5
46
38
30
22
14
6
47
39
31
23
15
7
4
5
6
7
8
9
10
11
12
13
14
15
16
17
K0
K1
K2
K3
R0
R1
R2
R3
D0
D1
D2
D3
D4
D5
MC40P5404
OSC1
OSC2
REM
OUT
D6
Vdd
GND
3
2
19
18
20
1
+