MAR. 2009 Rev. 1.5

4-BIT SINGLE CHIP MICROCOMPUTERS

MC40P5X01 SERIES

USER`S MANUAL

- MC40P5001
- MC40P5101
- MC40P5201
- MC40P5301

We hereby introduce the manual for CMOS 4-bit microcomputer MC40P5x01 Series.
This manual is prepared for the users who should understand fully the functions and features of MC40P5x01 Series so that you can utilize this product to its fullest capacity. A detailed explanations of the specifications and applications regarding the hardware is hereby provided.

REVISION HISTORY

Ver. 1.5 (MAR. 2009)
Added details of instruction system.(Page 33~48)
Modified Pin assignment.(Page 7)
Deleted details of SPGM.(Page 49)

Ver. 1.4 (FEB. 2008)
Added the I_{OL} capability of Built in Transistor for I.R LED Drive.(Page 2)
Added REMDRV Port I_{OL4} and V_{OL4} parameter data at Electrical characteristics.(Page 15)
Added REMDRV I_{OL4} vs V_{OL4} characteristic Graph.(Page 17)
Added the pin description of MC40P5101D.(Page 11)
Added REMOUT port structure.(Page 12)
Added Chpater 5 Circuit Diagram.(Page 64,65)
Modified Characteristic Graph location and note(Page 15, 16, 17)
Modified the REMOUT port of MC40P5001D & MC40P5201D to REMDRV.(Page 3, 5, 10)

Ver. 1.3
Modified some errata.

Ver. 1.2
Modified Pin assignment.(Page 7)

Ver. 1.1
Added LVD parameter data at Electrical characteristics.(Page 15)

Ver. 1.0 (AGU. 2007)
First Edition.

The contents of this user`s manual are subject to change for the reasons of later improvement of the features.
The information, diagrams, and other data in this user`s manual are correct and reliable; however, ABOV Semiconductor Inc. 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

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

CHAPTER 1. Introduction

Outline of characteristics

The MC40P5x01D series is 4-bit remote control MCU which uses CMOS technology and the 1K bytes EPROM version.

This enables transmission code outputs of different configurations, multiple custom code output, and double push key output for easy fabrication.

The MC40P5x01D series is suitable for remote control of TV, VCR, FANS, Air-conditioners, Audio Equipments, Toys, Games etc.

Characteristics

- Program memory : 1,024 bytes
- Data memory : 32 × 4 bits
- 43 types of instruction set
- 3 levels of subroutine nesting
- Operating frequency : 2.4MHz ~ 4MHz
- Instruction cycle : $f_{OSC}/48$
- 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 (MC40P5001D, MC40P5201D)
  - 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 : 2.0 ~ 3.6V
- 20/24 pin SOP package

Block Diagram

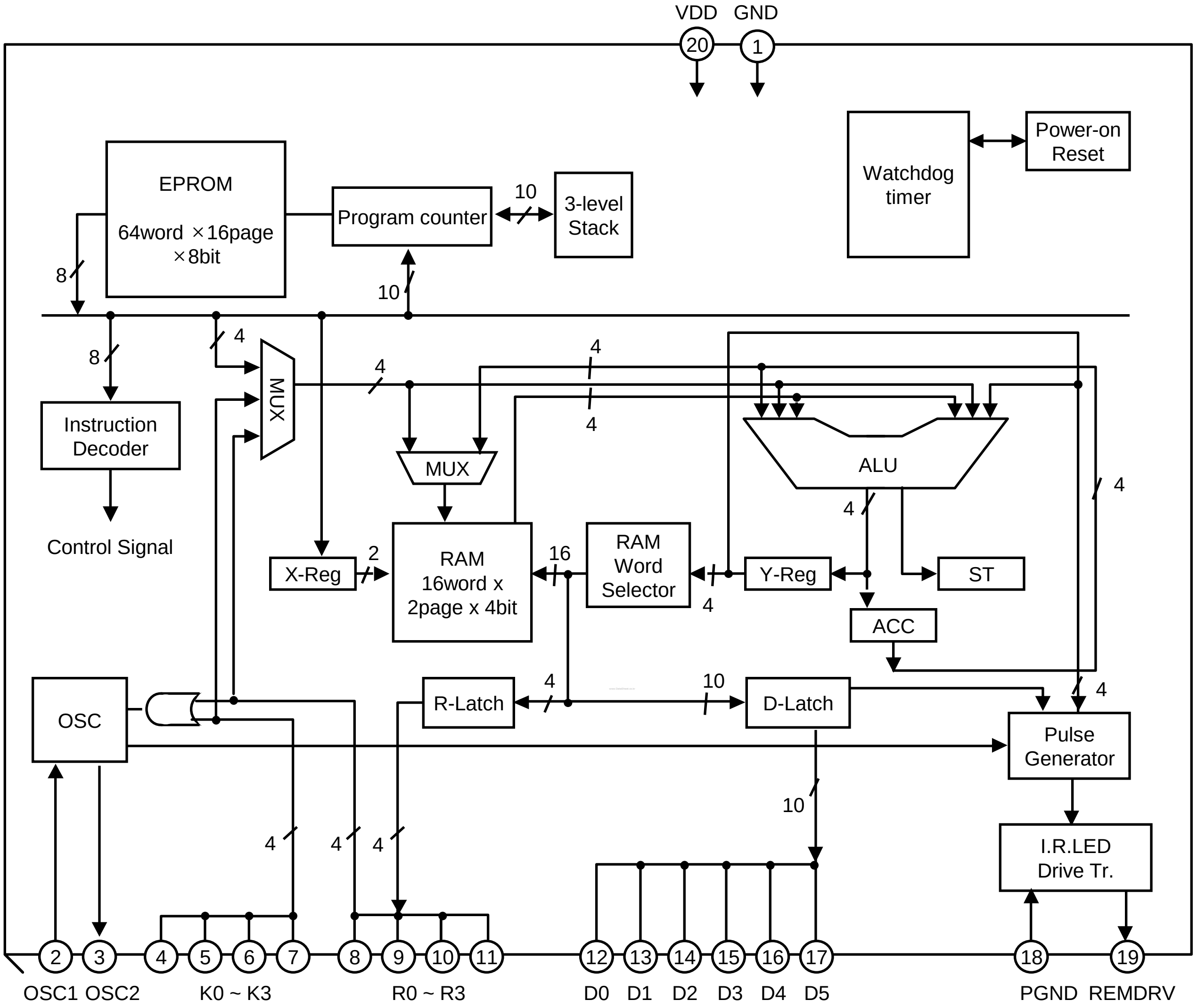


Fig 1-1 Block Diagram (MC40P5001D, 20 pins)

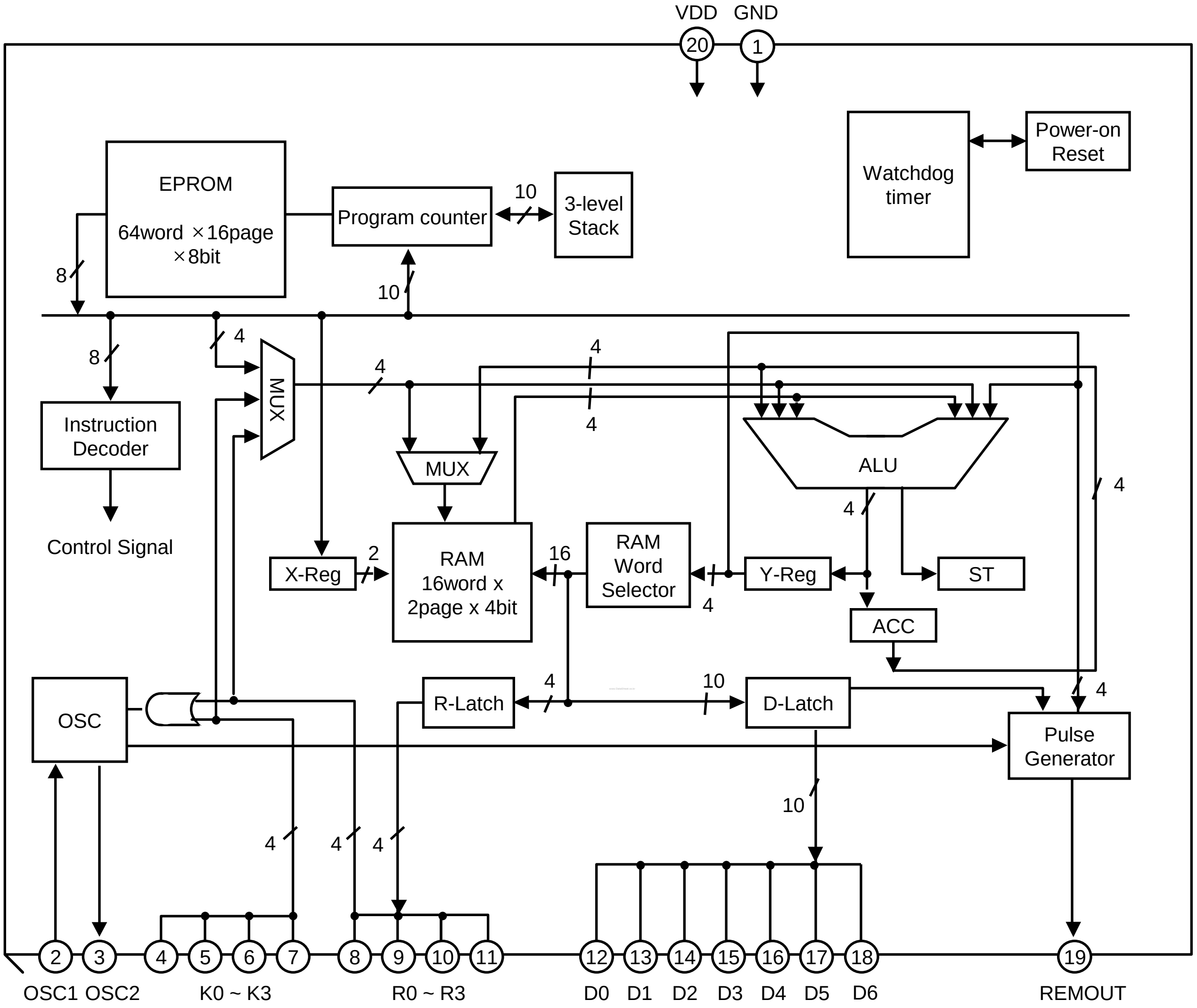


Fig 1-2 Block Diagram (MC40P5101D, 20 pins)

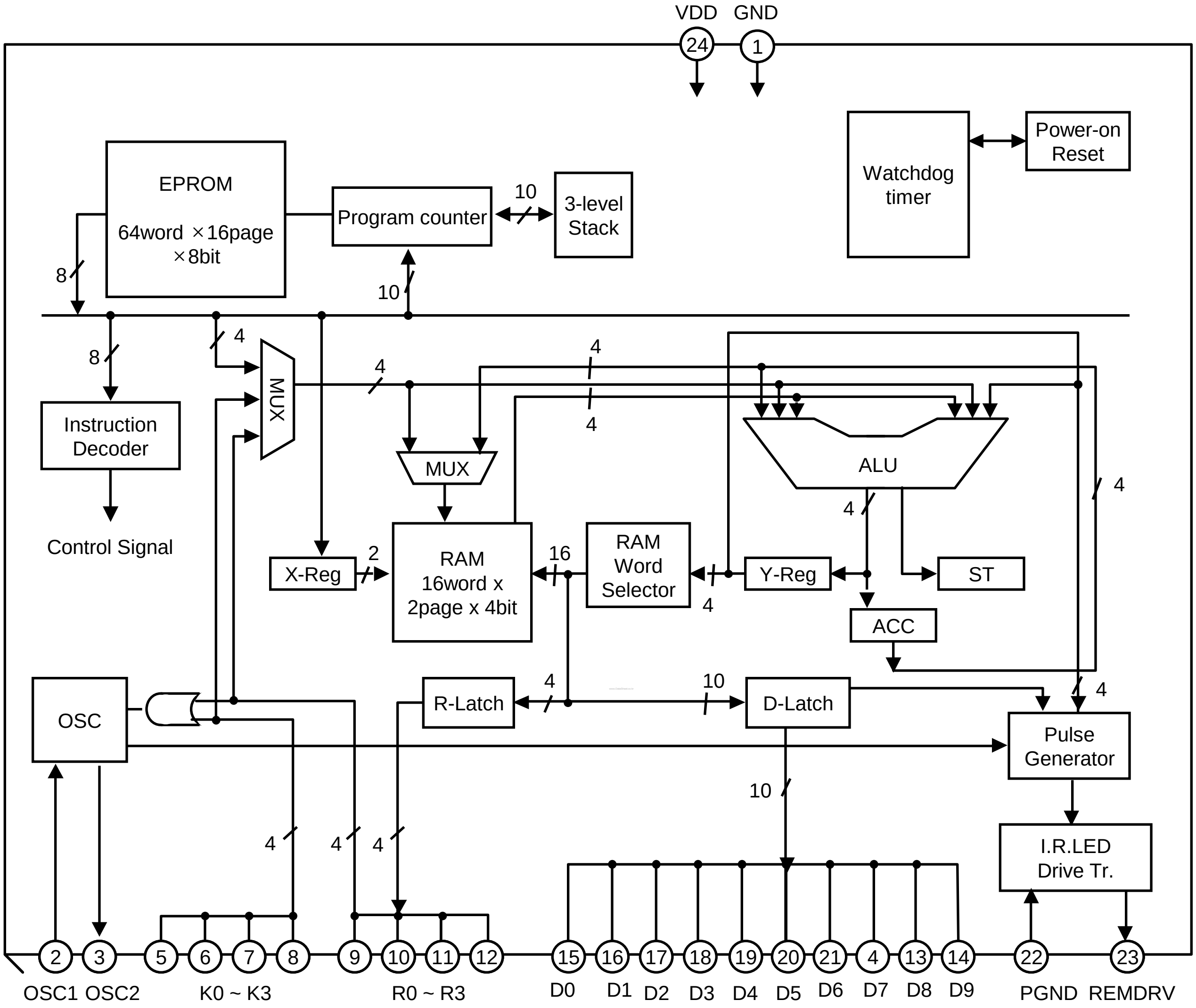


Fig 1-3 Block Diagram (MC40P5201D, 24 pins)

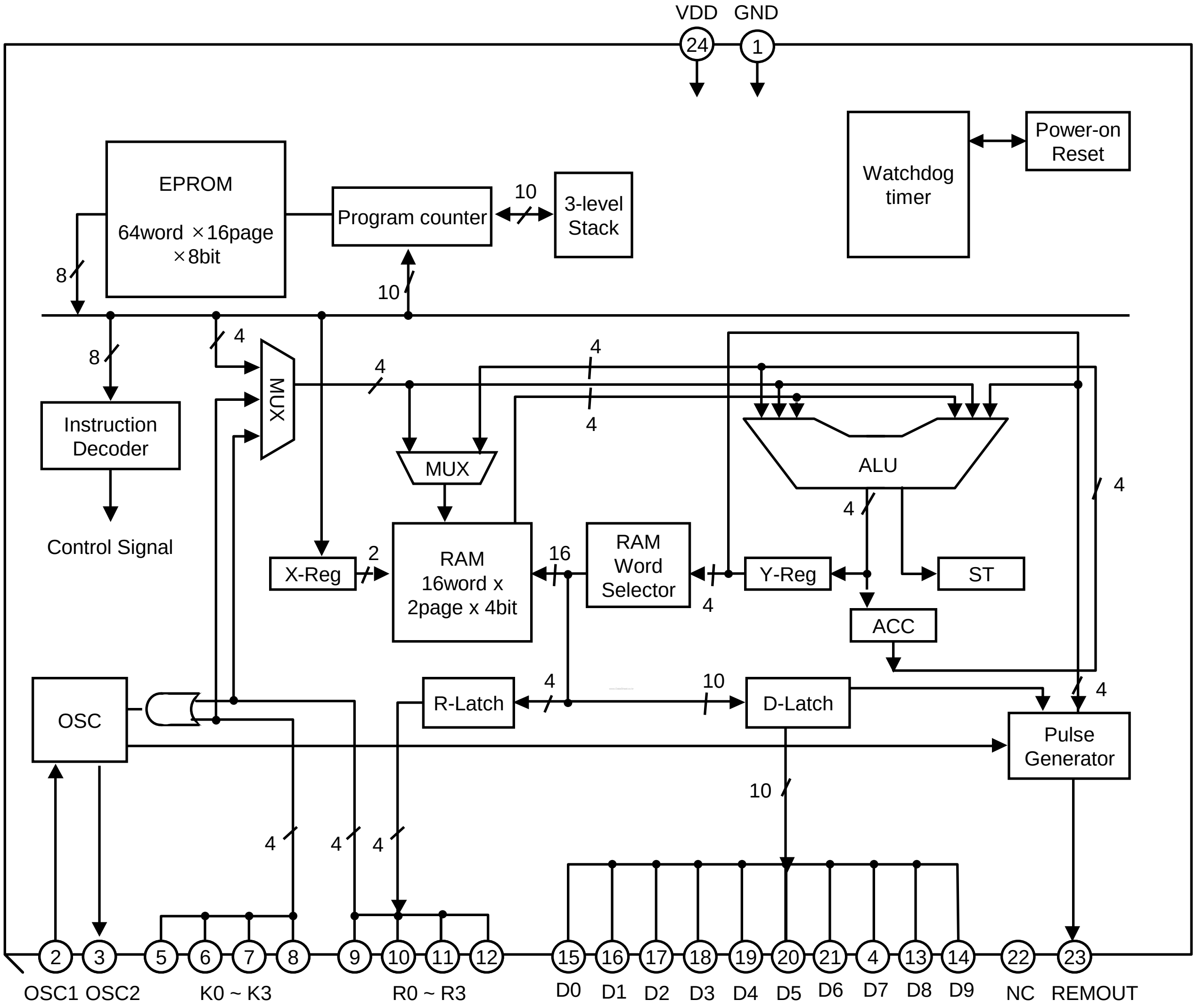


Fig 1-4 Block Diagram (MC40P5301D, 24 pins)

Pin Assignment

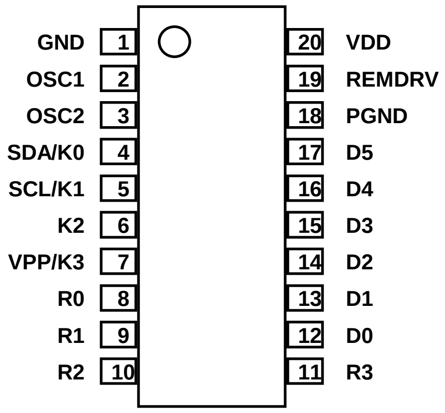


Fig 1-5 MC40P5001D Pin Assignment (20 PIN)

REMDRV : open drain output
VPP : K3 (port No.7)

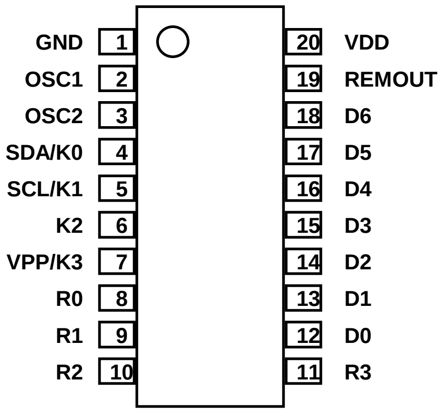


Fig 1-6 MC40P5101D Pin Assignment (20 PIN)

REMOUT : Push Pull output
VPP : K3 (port No.7)

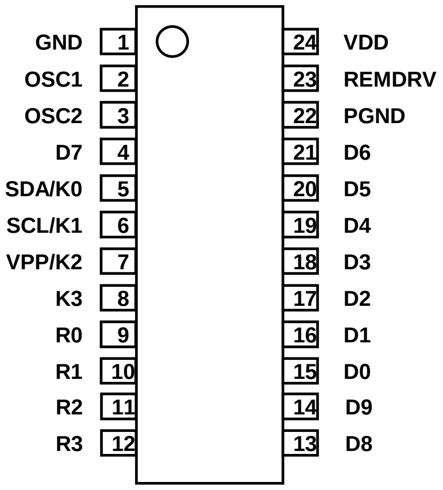


Fig 1-7 MC40P5201D Pin Assignment (24 PIN)

REMDRV : open drain output
VPP : K2 (port No. 7)

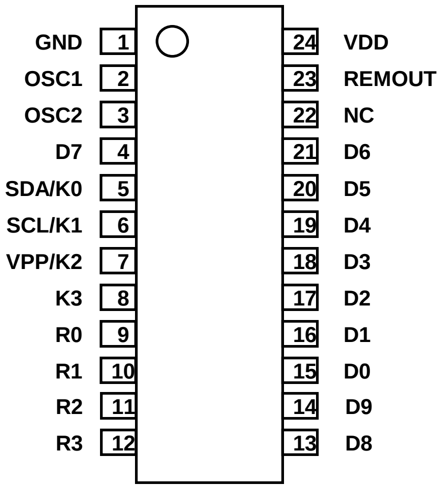


Fig 1-8 MC40P5301D Pin Assignment (24 PIN)

REMOUT : Push Pull output
VPP : K2 (port No. 7)

Pin Dimension

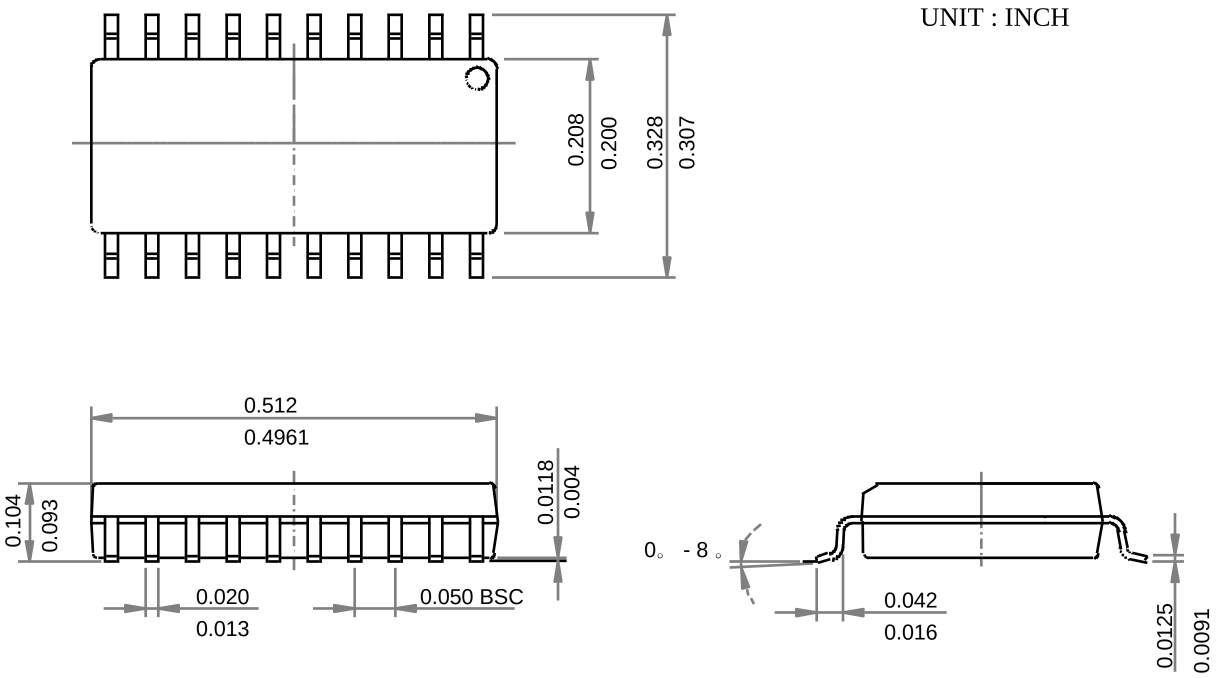


Fig 1-7. 20SOP (209MIL)

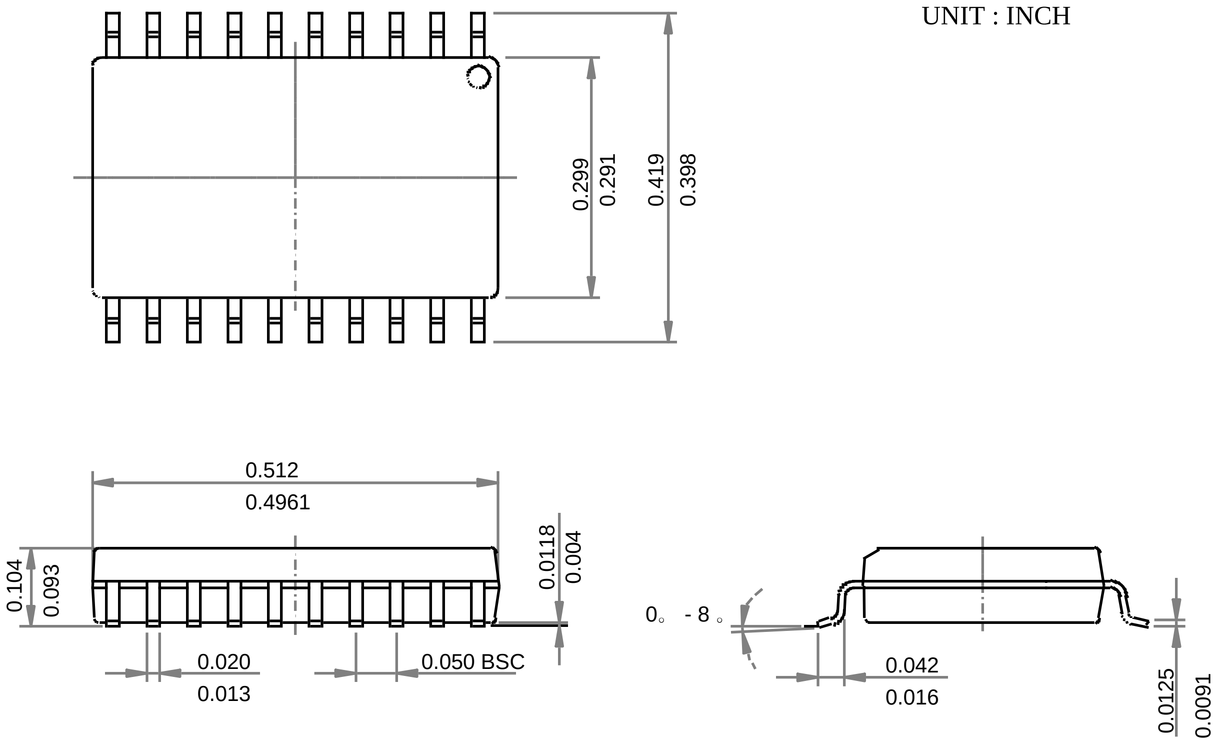


Fig 1-8. 20SOP (300MIL)

Pin Dimension

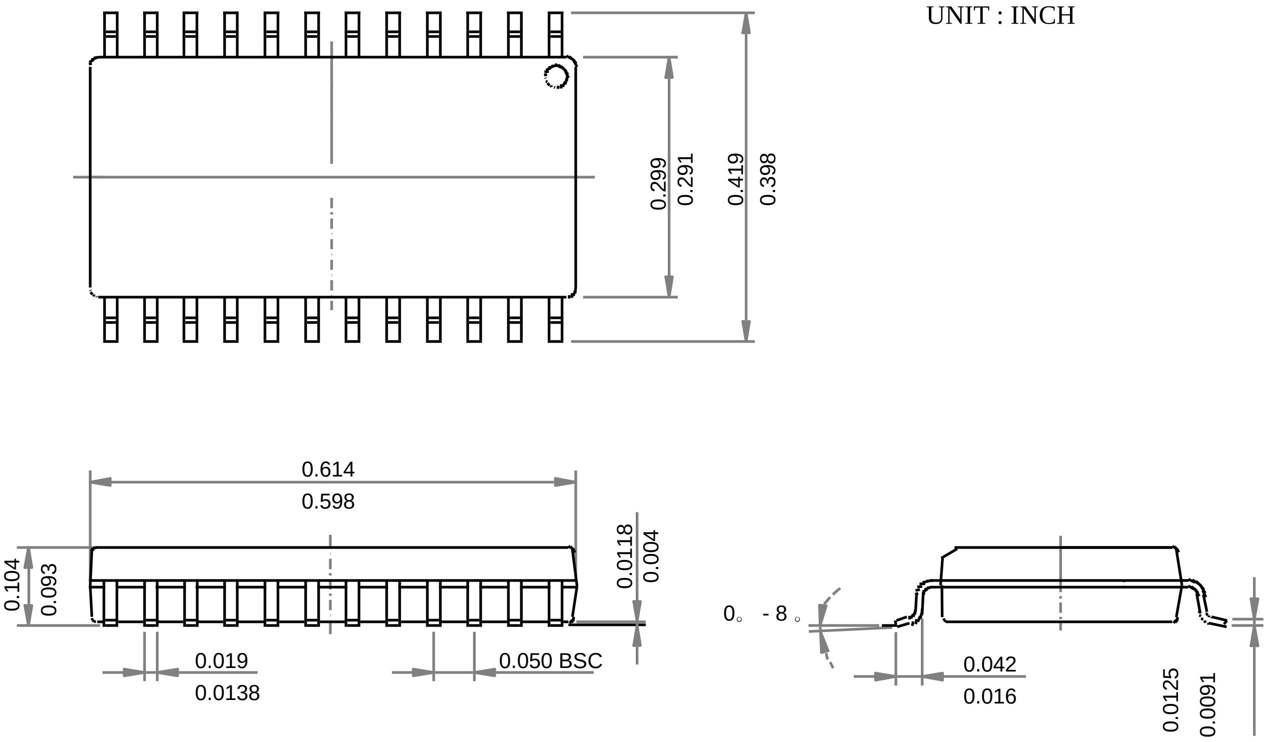


Fig 1-9. 24SOP (300MIL)

Pin Description and Circuit

Pin Description (MC40P5001D, 20 pins)

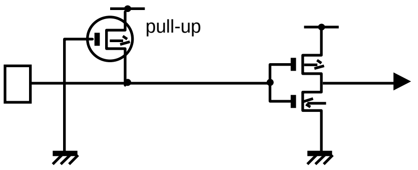
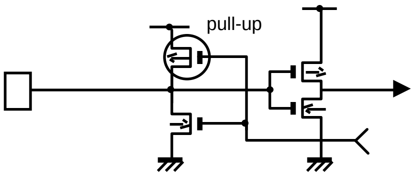
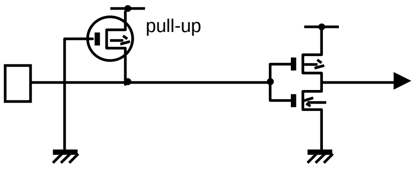
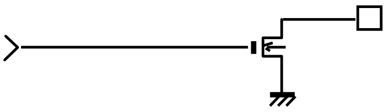
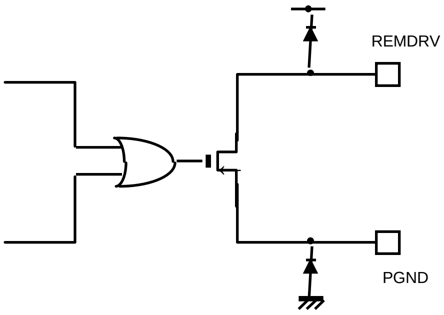
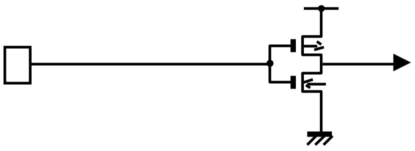
Pin	I/O	Function
VDD	-	Connected to 2.0~ 3.6V power supply
GND	-	Connected to 0V power supply.
K0 ~ K3	Input	4-bit input port with built in pull-up resistor. STOP mode is released by "L" input of each pin.
D0 ~ D5	Output	Each can be set and reset independently. The output is the structure of N-channel-open-drain.
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. Input mode is set only when each of them output "H". In outputting, each can be set and reset independently(or at once.) The output is in the form of C-MOS. STOP mode is released by "L" input of each pin.
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)
REMDRV	Output	High current output port for driving I.R.LED. The output is in the form N-channel open drain.

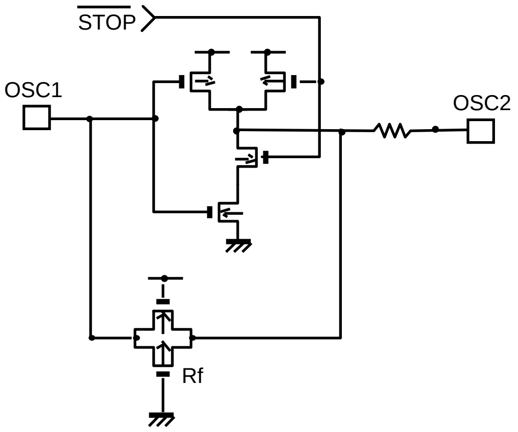
Pin Description and Circuit

Pin Description (MC40P5101D, 20 pins)

Pin	I/O	Function
VDD	-	Connected to 2.0~ 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 ~ D6	Output	Each can be set and reset independently. The output is the structure of N-channel-open-drain.
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. Input mode is set only when each of them output "H". In outputting, each can be set and reset independently(or at once.) The output is in the form of C-MOS. STOP mode is released by "L" input of each pin.
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.
REMOUT	Output	Output port for driving I.R.LED. The output is push-pull.

Pin Circuit (MC40P5001D & MC40P5101D, 20pins)

Pin	I/O	I/O circuit	Note
R0 ~ R1	I		- Built in MOS Tr for pull-up, about 140kΩ.
R2 ~ R3	I/O		- CMOS output. - "H" output at reset. - Built in MOS Tr for pull-up, about 140kΩ.
K0 ~ K3	I		- Built in MOS Tr for pull-up, about 140kΩ.
D0 ~ D5 (D0 ~ D6)	O		- Open drain output. - "L" output at reset. - D0~D3 are “L” output at STOP MODE. - D4 ~D5 keep before stop mode at STOP MODE. - D0~D6 : MC40P5101
REMDRV (MC40P5001D)	O		- Open drain output - Output Tr. Disable at reset.
REMOUT (MC40P5101D)	O		- Push Pull output.

Pin	I/O	I/O circuit	Note
OSC2	O		- Built in feedback-resistor about 1㏁
OSC1	I		

Note : at 24 pins, D0 ~D5 is changed to D0 ~ D9 (MC40P5201D, 24pins)
D8, D9 pin is automatically outputted “L” at STOP mode.
There is REMDRV at MC40P5201D and REMOUT at MC40P5301D.

Optional Features

The MC40P5001D is offered to the following option (OTP).
You can set up on MASK

- I/O terminals having pull-up resistor : R2 ~ R3
- Input terminals having STOP release mode : K0 ~ K3, R0 ~ R3
- Output form at STOP mode : D0 ~D3 pins are changed “Low” by force
- Output form at STOP mode : D4 ~D5 pins keep the status before stop mode

Electrical Characteristics

Absolute maximum ratings (Ta = 25℃)

Parameter	Symbol	Max. rating	Unit
Supply Voltage	V_{DD}	-0.3 ~ 5.0	V
Power dissipation	P_D	700 *	mW
Storage temperature range	Tstg	-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.

Recommended operating condition

Parameter	Symbol	Condition	Rating	Unit
Supply Voltage	V_{DD}	2.4MHz ~ 4MHz	2.0 ~ 3.6	V
Operating temperature	Topr	-	-20 ~ +70	℃

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	㏀	VI=GND
R Pull-up Resistance	R_{PU2}	70	140	300	㏀	VI=GND, Output off
Feedback Resistance	R_{FD}	0.3	1.0	3.0	㏁	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_{OL1}*2	0.5	1.1	3	mA	V_{OL1}=0.4V
REMDRV output L current	I_{OL4}*3	-	250 520	-	mA	V_{OL4}=0.3V V_{OL4}=0.52V
REMOUT output H current	I_{Oh1}*4	-5	-15	-30	mA	V_{oH1}=2V
REMOUT leakage current	I_{OLK1}	-	-	1	uA	V_{0UT}=V_{DD}, Output off
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} *5	-	0.5	1.5	mA	f_{OSC}=4MHz
System clock frequency f_{OSC}/48	f_{OSC}	2.4	-	4	MHz	MHZ version

*1 Refer to Fig.1-11 < I_{OL2} vs. V_{OL2} Graph>

*2 Refer to Fig.1-12 < I_{OL1} vs. V_{OL1} Graph>

*3 Refer to Fig.1-13 < I_{OL4} vs. V_{OL4} Graph>

*4 Refer to Fig.1-10 < I_{OH1} vs. V_{OH1} Graph>

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

Fig 1-10. I_{OH1} vs V_{OH1} Graph (REMOUT Port)

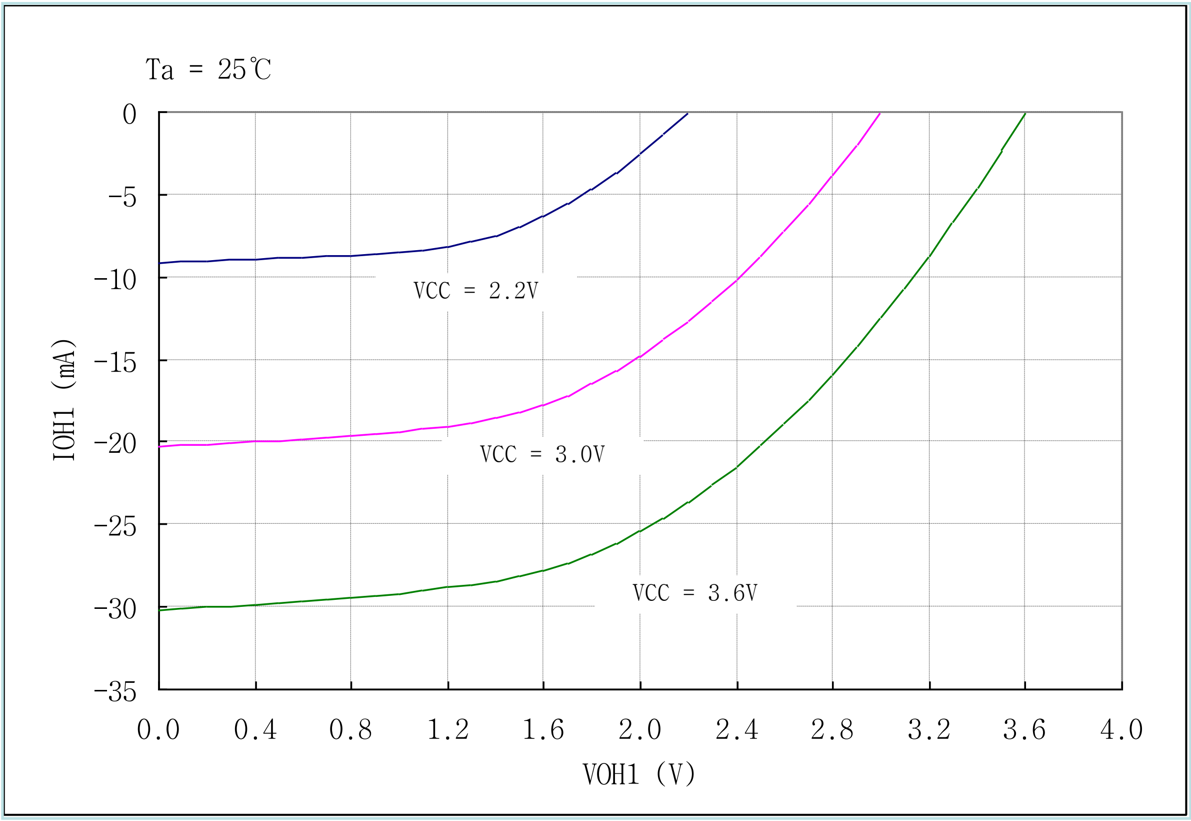


Fig 1-11. I_{OL2} vs. V_{OL2} Graph. (D, R Port)

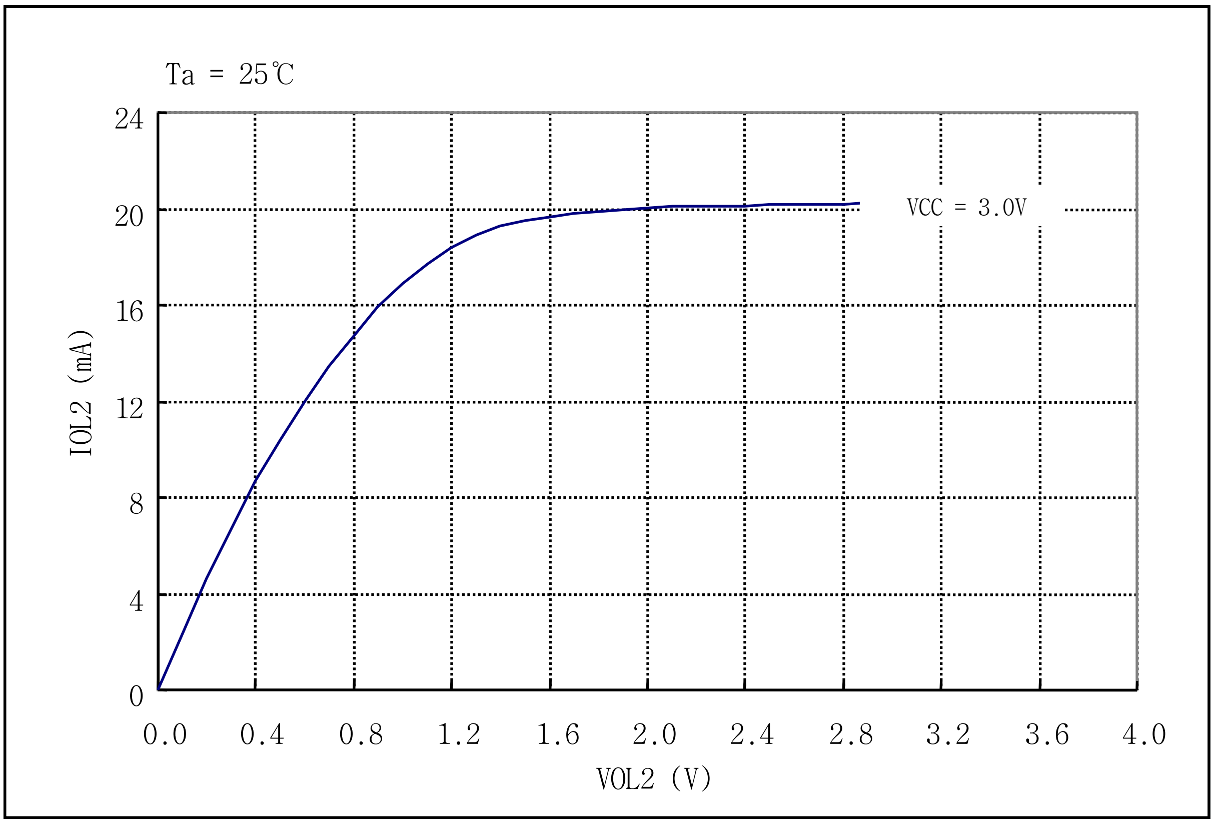


Fig 1-12. I_{OL1} vs V_{OL1} Graph (REMOUT without built-in Transistor of MC40P5101D and MC40P5301D)

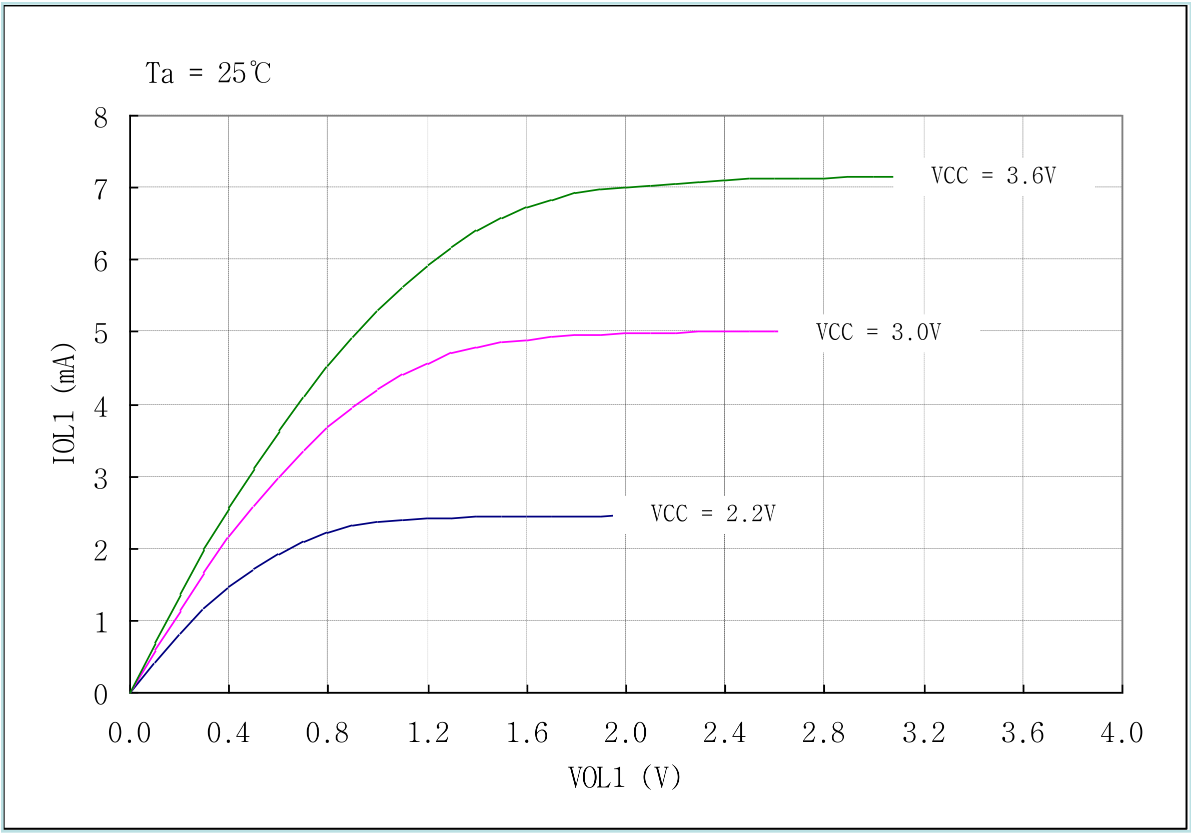
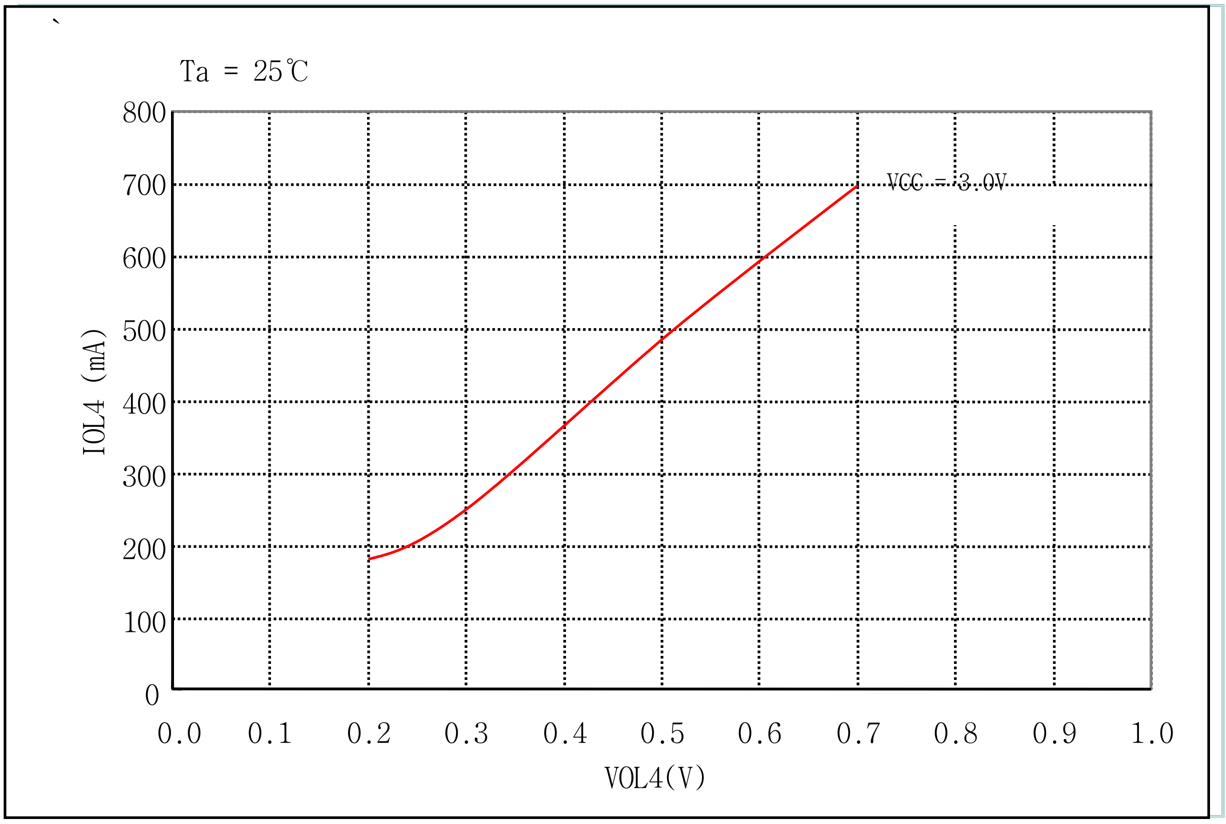


Fig 1-13. I_{OL4} vs. V_{OL4} Graph. (REMOUT Port with built-in Transistor of MC40P5001D and MC40P5201D)



CHAPTER 2. Architecture

BLOCK DESCRIPTION

Program Memory (EPROM)

The MC40P5x01D series can incorporate maximum 1,024 words (64 words x 6 page x 8bits) for program memory. Program counter PC (A0~A5) and page address register (A6~A9) 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.

he program memory is composed as shown below

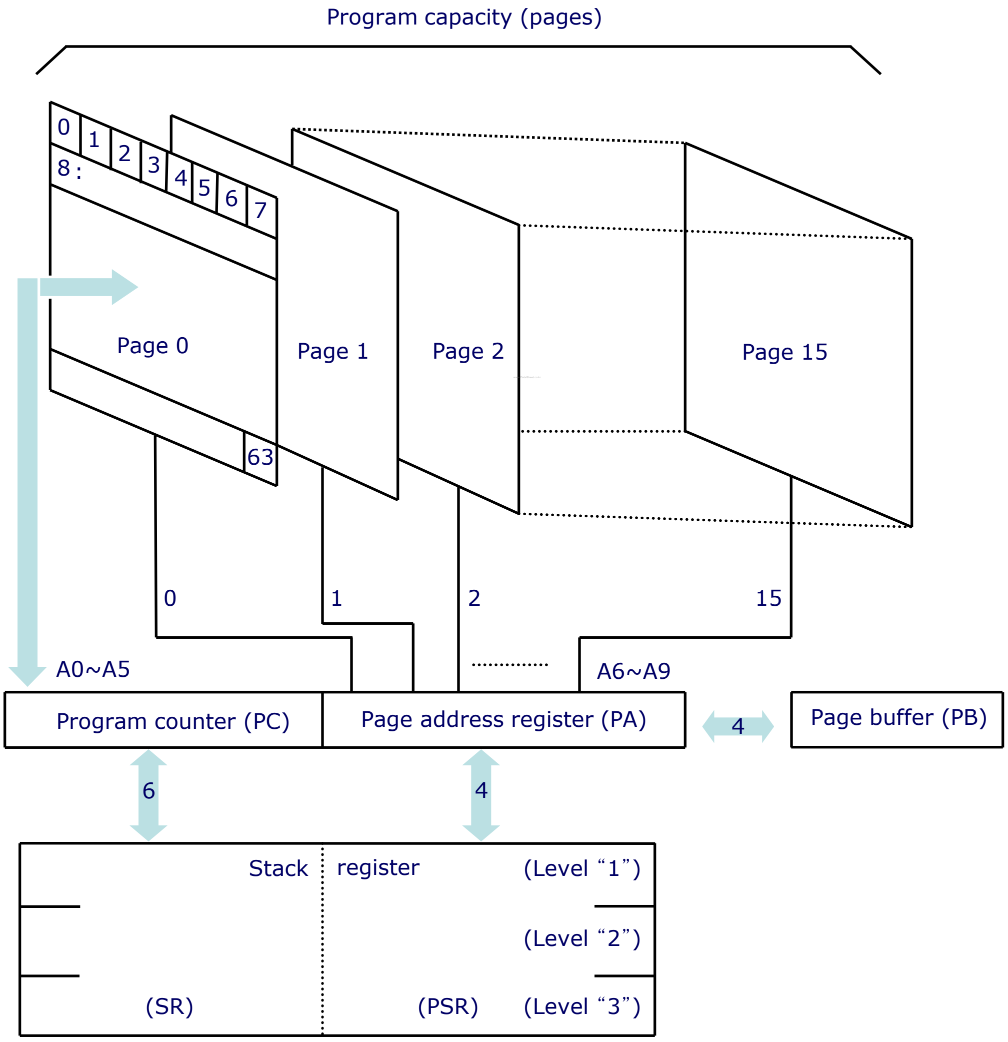


Fig 2-1 Configuration of Program Memory

EPROM Address Register

The following registers are used to address the EPROM.

- Page address register (PA)
  Holds EPROM's page number (0~Fh) 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).
- 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.

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 2-2 shows the configuration.

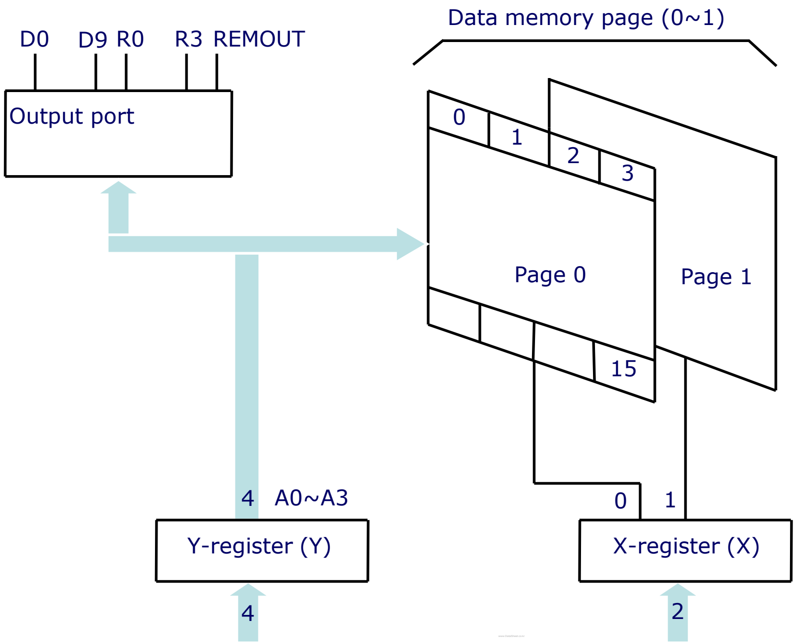


Fig 2-2 Configuration of Data Memory

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	X1=1
Y=0	D0	D8
Y=1	D1	D9

Table 2-1 Mapping table between X and Y register

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.

Accumulator (Acc)

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

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.

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.

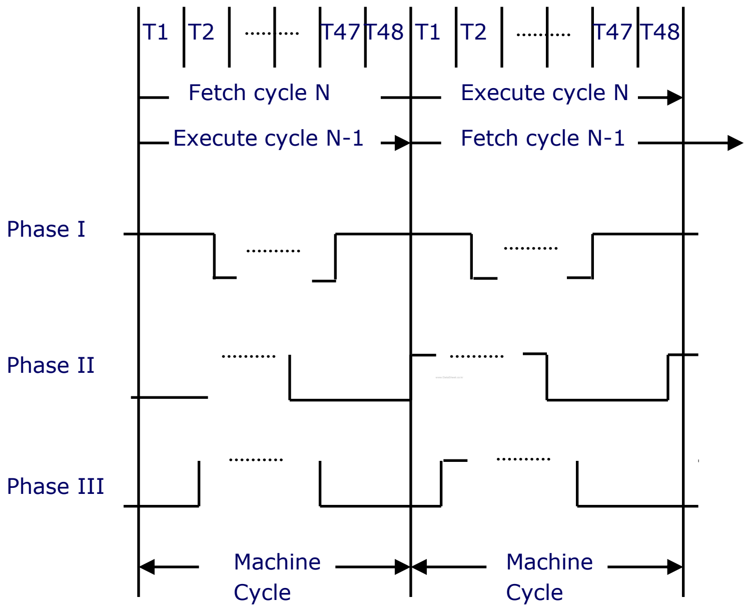


Fig 2-3 Fundamental timing chart

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.

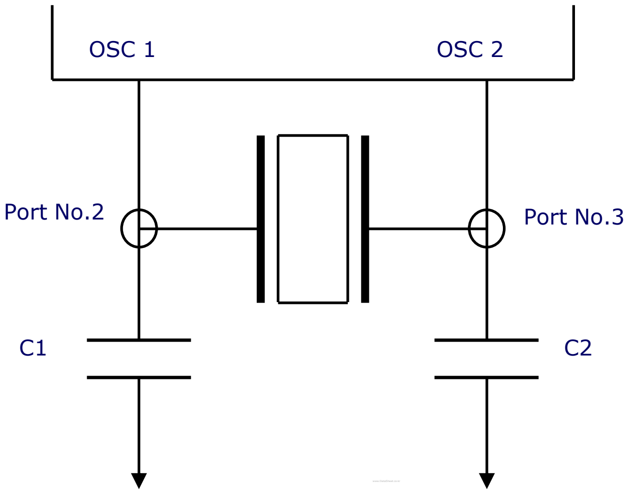


Figure 2-4 Oscillator circuit with external capacitor

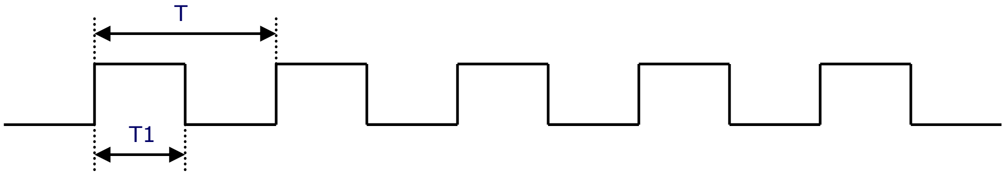
※ Note
Matching Test results are below.

Maker	Device Names	Test Result
BaoTong	RT3.640MG	matched
Chequers	ZTT3.64MGW ZTT3.84MGW ZTT4.0MGW	matched
TDK	FCR3.64MC5 FCR4.0MC5	matched

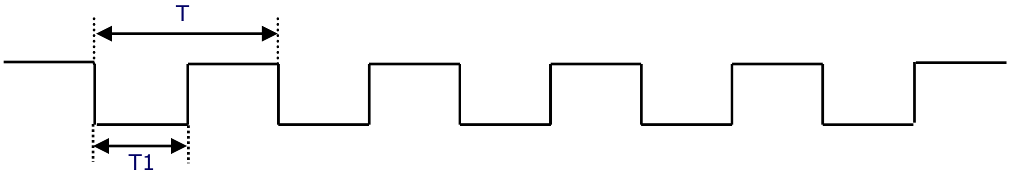
Table 2-2 Matching Test Results

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.



(MC40P5101D, MC40P5301D)



(MC40P5001D, MC40P5201D)

PMR	REMOUT signal
0	$T=1/f_{PUL} = 96/f_{osc}$, $T1/T = 1/2$
1	$T=1/f_{PUL} = 96/f_{osc}$, $T1/T = 1/3$
2	$T=1/f_{PUL} = 64/f_{osc}$, $T1/T = 1/2$
3	$T=1/f_{PUL} = 64/f_{osc}$, $T1/T = 1/4$
4	$T=1/f_{PUL} = 88/f_{osc}$, $T1/T = 4/11$
5	No Pulse (same to D0~D9)
6	$T=1/f_{PUL} = 96/f_{osc}$, $T1/T =1/4$
7	$T= 1/f_{PUL} = 8/f_{osc}$, $T1/T = 1/2$

*Default value is “0”

* f_{PUL}= Pulse frequency, f_{osc} = Oscillation frequency

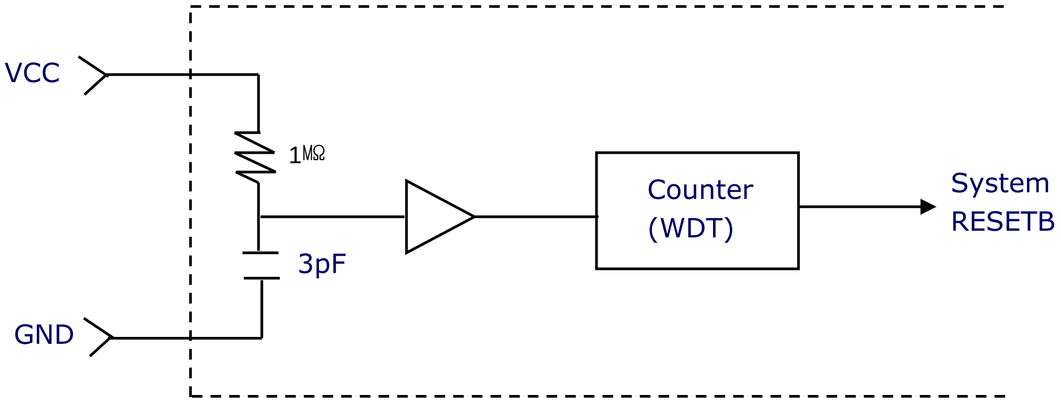
Table 2-3 PMR selection table

Reset Operation

MC40P5x01D 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 Mc40P5001D, MC40P5201D

Built-in Power On Reset Circuit

MC40P5x01D series has a built-in Power-on reset circuit consisting of an about 1㏁ 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 (213 x System clock time), system reset signal is released.



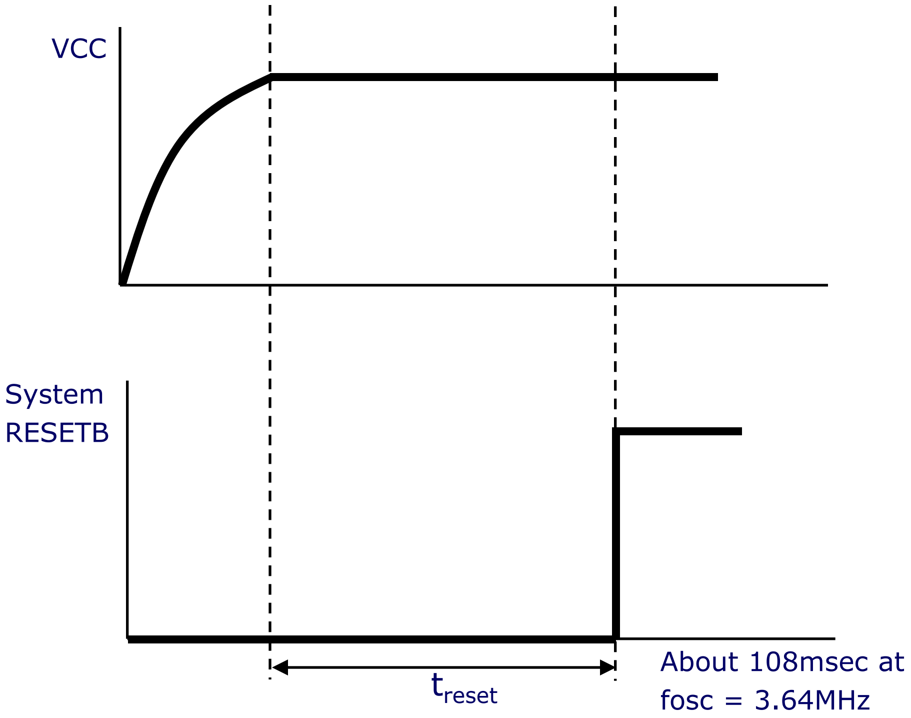


Fig2-5 Power –On Reset Circuit and Timing Chart

Built-in Low VDD Reset Circuit

MC40P5x01D 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.

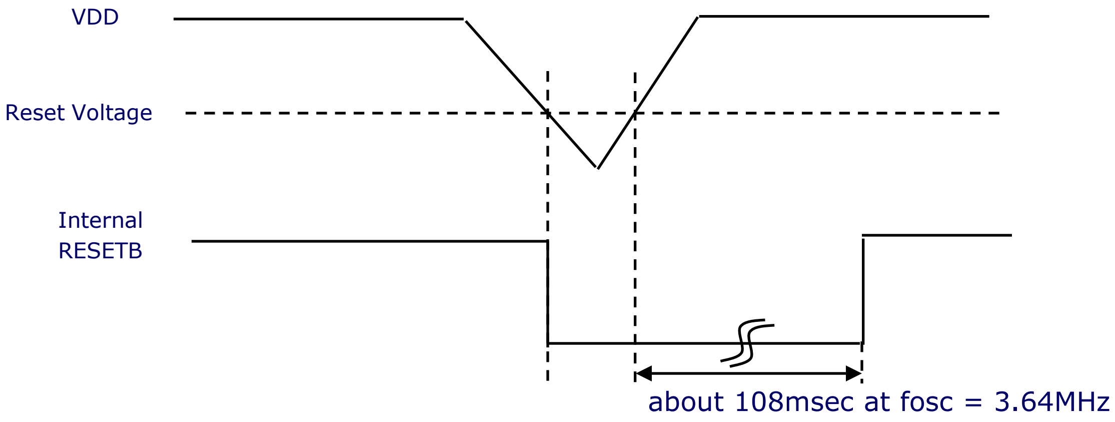
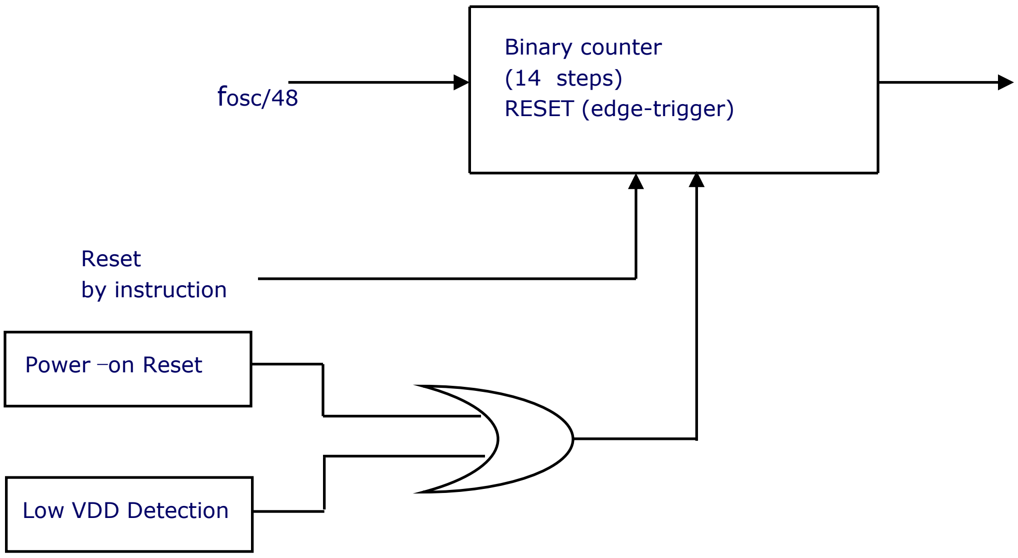


Fig2-6 Low Voltage Detection Timing Chart

Watch Dog Timer (WDT)

Watch dog timer is organized binary of 14 steps. The signal of f_{osc} /48 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} /f_{osc} (108.026ms at f_{osc} = 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 page 29 STOP operation>)



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, D8~D9 output and REMOUT output are "L".
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.

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") SO : REMOUT(PMR) ← 1 RO : REMOUT(PMR) ← 0
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

CHAPTER 3. Instruction

INSTRUCTION FORMAT

All of the 43 instruction in MC40P5X01D 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.

*Format Ⅱ

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″)

*Format Ⅲ

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.

Instruction Table

The MC40P5x01D 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	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		DA	A ← A − 1	B

	Category	Mnemonic	Function	ST[*1]
26	Arithmetic	DY	Y ← Y – 1	B
27		EORM	A ← A ⊕ M(X,Y)	S
28		NEGA	A ← $\bar{A}$ + 1	Z
29	Comparison	ALEM	TEST A ≤ M(X,Y)	E
30		ALEM 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..

DETAILS OF INSTRUCTION SYSTEM

All 43 basic instructions of the MC40P5X01D 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 $0 \le i \le 15$
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 exeute 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.

<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 is 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 borrow 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 ← 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 : $A \leq M(X,Y)$ $ST \leftarrow 1$ (when $A \leq M(X,Y)$)
$ST \leftarrow 0$ (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

Function : $A \leq i$ $ST \leftarrow 1$ (when $A \leq i$)
$ST \leftarrow 0$ (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 : $M(X,Y) \neq 0$ $ST \leftarrow 1$(when $M(X,Y) \neq 0$)
$ST \leftarrow 0$ (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 : $Y \neq A$ $ST \leftarrow 1$ (when $Y \neq A$)
$ST \leftarrow 0$ (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 $0 \leq i \leq 15$

Function : $Y \neq i$ $ST \leftarrow 1$ (when $Y \neq i$)
$ST \leftarrow 0$ (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 $K \neq 0$, $ST \leftarrow 1$

<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 $R \neq 0$, $ST \leftarrow 1$

<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 ← 1(PMR=5)	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 come 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 ← 0	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 1-12 page, 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

※Assembler Macro

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

Page call (2byte) :

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

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

Page branch (2byte) :

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

SPGM(Serial Program)

※ The I^{2}C Bus Protocol

The I^{2}C 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^{2}C bus. and SDA is the Serial Data line. The SCL & SDA lines are connected to all devices on the I^{2}C bus.

Summary of Protocol

・necessary pins (5pins)

- Serial Data (SDA)	: K0
- Serial Clock (SCL)	: K1
- VPP	: K3 **(20pin)**
	: K2 (24pin)
- VDD	
- VSS	

· LOCK PROGRAM / READ DATA Format

ID6	ID5	ID4	ID3	ID2	ID1	ID0	Lock	Tail

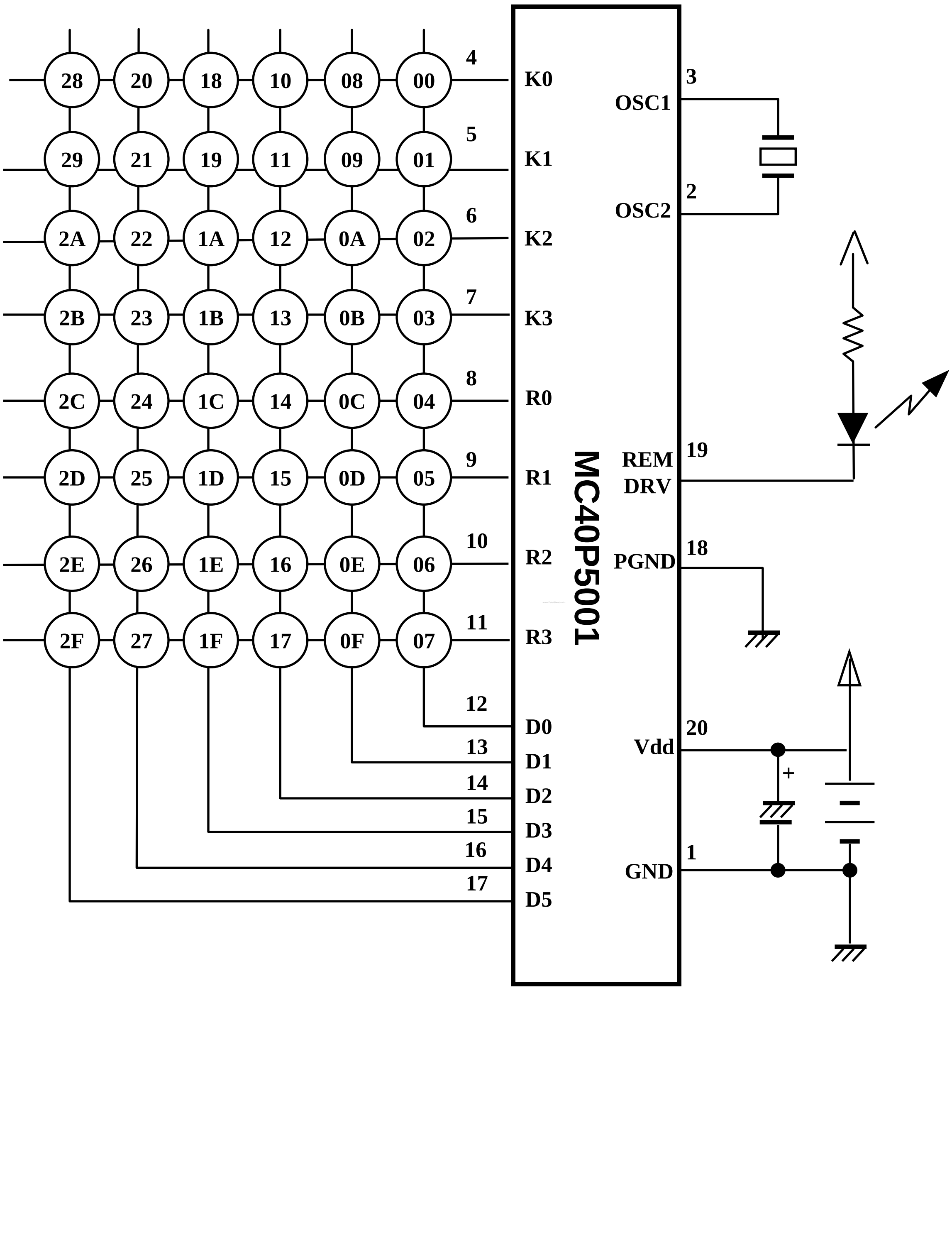
- ID6 – ID0 : it can be treated as User ID.

MC40P5001D ID: 1000 000Xb
MC40P5101D ID: 1001 000Xb
MC40P5201D ID: 1010 000Xb
MC40P5301D ID: 1011 000Xb

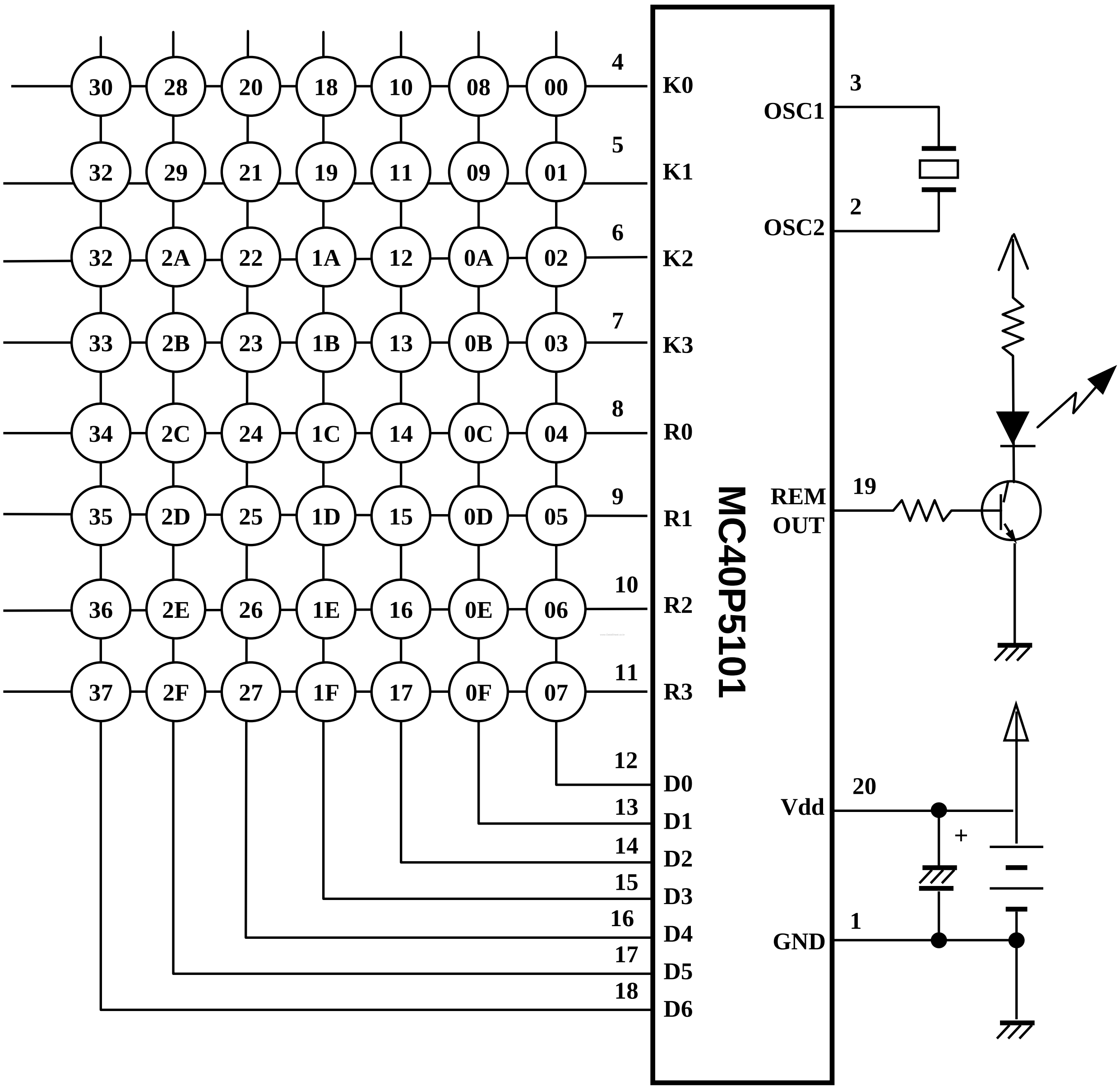
※ For protection the written program code, in other words it can not be read, you have to clear the Lock bit to “0”, and for this, you have to write the Lock Register to 1111_1110b. In this time, ID6 – ID0 keep the existing value without any effect

INTRODUCTION	1
ARCHITECTURE	2
INSTRUCTION	3
SPGM	4
CIRCUIT DIAGRM	5

MC40P5001D with Built-in TR Circuit Diagram



MC40P5101D without Built-in TR Circuit Diagram


30 28 20 18 10 08 00
32 29 21 19 11 09 01
32 2A 22 1A 12 0A 02
33 2B 23 1B 13 0B 03
34 2C 24 1C 14 0C 04
35 2D 25 1D 15 0D 05
36 2E 26 1E 16 0E 06
37 2F 27 1F 17 0F 07
4 K0
5 K1
6 K2
7 K3
8 R0
9 R1
10 R2
11 R3
12 D0
13 D1
14 D2
15 D3
16 D4
17 D5
18 D6
MC40P5101
3 OSC1
2 OSC2
19 REM OUT
20 Vdd
1 GND
+