MSM80C154S
MSM83C154S
MSM85C154HVS

USER'S MANUAL

CONTENTS

1. INTRODUCTION

1.1 MSM80C154S/MSM83C154S/MSM85C154HVS Outline 3
1.2 MSM80C154S/MSM83C154S Features 5
1.3 Additional Features in MSM80C154S/MSM83C154S/MSM85C154HVS 7

2. SYSTEM CONFIGURATION

2.1 MSM80C154S/MSM83C154S/MSM85C154HVS Logic Symbols 11
2.2 MSM80C154S/MSM83C154S Pin Layout 12
2.2.1 MSM80C154S/MSM83C154S external dimensions 15
2.2.2 MSM85C154HVS pin layout and external dimensions 17
2.3 MSM80C154S Block Diagram 18
2.4 MSM83C154S Block Diagram 19
2.5 MSM85C154HVS Block Diagram 20
2.6 Timing and Control 21
2.6.1 Outline of MSM80C154S/MSM83C154S timing 21
2.6.2 Major synchronizing signals 23
(1) ALE 23
(2) $\overline{\text{PSEN}}$ 23
(3) $\overline{\text{WR}}$ 23
(4) $\overline{\text{RD}}$ 23
2.6.3 MSM80C154S fundamental operation time charts 24
(1) External program memory read cycle timing chart 24
(2) MOVX A, @Rr 24
(3) MOVX @Rr, A 25
(4) MOVX A, @DPTR 25
(5) MOVX @DPTR, A 26
(6) MOV direct, PORT[0, 1, 2, 3] execution 26
2.6.4 MSM83C154S fundamental operation time charts 27
(1) MOVX A, @Rr 27
(2) MOVX @Rr, A 27
(3) MOVX A, @DPTR 28
(4) MOVX @DPTR, A 28
(5) MOV direct, PORT[0, 1, 2, 3] execution 29
2.7 Instruction Register (IR) and Instruction Decoder (PLA) 30
2.8 Arithmetic Operation Section 31
(1) Outline 31
(2) Arithmetic operation instruction decoder 31
(3) Arithmetic and logic unit (ALU) 31
2.9 Program Counter 32
2.10 Program Memory and External Data Memory 33
2.10.1 MSM80C154S/MSM83C154S program area and external ROM connections 33
2.10.2 Procedures and circuit connections used when external data memory (RAM) is accessed by data pointer (DPTR) 35
2.10.3 Procedures and circuit connections used when external data memory (RAM) is accessed by registers R0 and R1 38

3. CONTROL

3.1 Oscillators [XTAL1 .2] 43
3.2 CPU Resetting 45
3.2.1 Outline 45
3.2.2 Reset Schmitt trigger circuit 50
3.2.3 CPU internal status by reset 51
3.3 $\overline{EA}$(CPU Memory Separate) 52
3.3.1 Outline 52
(1) Internal ROM mode 52
(2) External ROM mode 52

4. INTERNAL SPECIFICATIONS

4.1 Internal Data Memory (RAM) and Special Function Registers 55
4.1.1 Outline 55
4.2 Internal Data Memory (RAM) 57
4.2.1 Internal data memory (RAM) 57
4.2.2 Internal data memory registers R0 thru R7 59
4.2.3 Stack 60
4.3 Internal Data Memory (RAM) Operating Procedures 61
4.3.1 Internal data memory indirect addressing 61
4.3.2 Internal data memory register R0 thru R7 designation 62
4.3.3 Internal data memory 1-bit data designation 63
4.4 Special Function Registers(TCON, SCON,...ACC, B) 65
4.4.1 Outline 65
4.4.2 Special function registers 67
4.4.2.1 Timer mode register (TMOD) 67
4.4.2.2 Power control register (PCON) 68
4.4.2.3 Timer control register (TCON) 69
4.4.2.4 Serial port control register (SCON) 70
4.4.2.5 Interrupt enable register (IE) 71
4.4.2.6 Interrupt priority register (IP) 72
4.4.2.7 Program status word register (PSW) 73
4.4.2.8 I/O control register (IOCON) 74
4.4.2.9 Timer 2 control register (T2CON) 75
4.5 Timer/Counters 0, 1, and 2 76
4.5.1 Outline 76
4.5.2 Timer/counters 0 and 1 76
4.5.2.1 Outline 76
4.5.2.2 Timer/counter 0 and 1 counting control 76
4.5.2.3 Timer/counter 0 and 1 count clock designation 78
4.5.2.3.1 External clock detector circuit for timer/counters 0 and 1 79
4.5.2.4 Counting control of timer/counters 0 and 1 by $\overline{INT}$ pin 80
4.5.2.5 Timer/counters 0/1 timer modes 82
4.5.2.5.1 Outline 82
4.5.2.5.2 Mode 0 82
4.5.2.5.3 Mode 1 84
4.5.2.5.4 Mode 2 86
4.5.2.5.5 Mode 3 88
4.5.2.5.6 32-bit timer mode 89

4.5.2.5.7 Caution about use of timer counters 0 and 1 90
4.5.2.5.8 Caution about use of timer counters 0 and 1 when setting software power down mode 91
4.5.3 Timer/counter 2 92
4.5.3.1 Outline 92
4.5.3.2 Timer 2 control register (T2CON) 92
4.5.3.3 Timer/counter 2 operation modes 93
4.5.3.3.1 16-bit auto reload mode 93
4.5.3.3.2 16-bit capture mode 94
4.5.3.3.3 16-bit baud rate generator mode 95
4.5.3.4 Timer/counter 2 detector circuit 97
4.5.3.4.1 T2(timer/counter 2 external clock detector) 97
4.5.3.4.2 T2EX(timer/counter 2 external flag input detector) 97
4.5.3.5 Timer/counter carry signal detector circuit 98
4.6 Serial Port 99
4.6.1 Outline 99
4.6.2 Special function registers for serial port 101
4.6.2.1 SCON 101
4.6.2.2 SBUF 103
4.6.2.3 TCLK 103
4.6.2.4 RCLK 103
4.6.2.5 SMOD 104
4.6.2.6 SERR 105
4.6.3 Operating modes 106
4.6.3.1 Mode 0 106
4.6.3.1.1 Outline 106
4.6.3.1.2 Mode 0 baud rate 106
4.6.3.1.3 Mode 0 transmit operation 106
4.6.3.1.4 Mode 0 receive operation 106
4.6.3.2 Mode 1 110
4.6.3.2.1 Outline 110
4.6.3.2.2 Mode 1 baud rate 110
4.6.3.2.3 Mode 1 transmit operation 111
4.6.3.2.4 Mode 1 receive operation 111
4.6.3.2.5 Mode 1 UART error detection 112
4.6.3.3 Mode 2 115
4.6.3.3.1 Outline 115
4.6.3.3.2 Mode 2 baud rate 115
4.6.3.3.3 Mode 2 transmit operation 115
4.6.3.3.4 Mode 2 receive operation 115
4.6.3.3.5 Mode 2 UART error detection 116
4.6.3.4 Mode 3 119
4.6.3.4.1 Outline 119
4.6.3.4.2 Mode 3 baud rate 119
4.6.3.4.3 Mode 3 transmit operation 120
4.6.3.4.4 Mode 3 receive operation. 120
4.6.3.4.5 Mode 3 UART error detection 121
4.6.4 Serial port application examples 124
4.6.4.1 I/O extension 124

4.6.4.2 Multi-processor systems 128
4.7 Interrupt 129
4.7.1 Outline 129
4.7.2 Interrupt enable register (IE) 131
4.7.3 Interrupt priority register (IP) 132
4.7.3.1 Priority interrupt routine flow 133
4.7.3.2 Interrupt routine flow when priority circuit is stopped 134
4.7.3.3 Interrupt priority when priority register (IP) contents are all “0” 135
4.7.4 Detection of external interrupt signals $\overline{INT}$0 and $\overline{INT}$1 136
4.7.4.1 Outline of $\overline{INT}$ signal detection 136
4.7.4.2 External interrupt signal 0 and 1 level detection 136
4.7.4.3 External interrupt signal 0 and 1 trigger detection 137
4.7.5 MSM80C154S/MSM83C154S interrupt response time charts 138
4.7.5.1 Interrupt response time chart when interrupt conditions are satisfied during execution of ordinary instruction in main routine 138
4.7.5.2 Interrupt response time chart when interrupt conditions are satisfied during execution of IE or IP register operation instruction in main routine 140
4.7.5.3 Interrupt response time chart when an ordinary instruction is executed after temporarily returning to the main routine from continuous interrupt processing 142
4.7.5.4 Interrupt response time chart when an IE or IP manipulating instruction is executed after temporarily returning to the main routine from continuous interrupt processing 144
4.8 CPU “Power Down” 146
4.8.1 Outline 146
4.8.2 Idle mode (IDLE) setting 146
4.8.3 Soft power down mode (PD) setting 151
4.8.3.1 Caution about software power down mode setting 151
4.8.4 Hard power down mode (HPD) setting 161
4.9 CPU Power Down Mode (IDLE, PD, and HPD) Cancellation (CPU Activation) 169
4.9.1 Outline 169
4.9.2 Cancellation by CPU resetting (RESET pin) 169
4.9.3 Cancellation of CPU power down mode(IDLE, PD)by interrupt signal 176
4.9.3.1 Cancellation of CPU power down mode (IDLE, PD) from interrupt address 176
4.9.3.2 Cancellation of CPU power down mode (IDLE, PD) by interrupt request signal and restart from next address of stop address 182
4.10 MSM80C154S/83C154S Battery Backup with Hard Power Down Mode 187

5. INPUT/OUTPUT PORTS

5.1 Outline 192
5.2 Port 0 192
5.3 Port 1 195
5.4 Port 2 201
5.5 Port 3 203
5.6 Port 0, 1, 2, and 3 Output and Floating Status Settings in CPU Power Down Mode (PD, HPD) 205

5.7 High Impedance Input Port Setting of Each Quasi-bidirectional Port 1, 2, and 3 207
5.8 100 kΩ Pull-Up Resistance Setting for Quasi-bidirectional Input Ports 1, 2, and 3 207
5.9 Precautions When Driving External Transistors by Quasi-bidirectional Port Output Signals 208
5.10 Port Output Timing 210
1) One machine cycle instruction output timing 210
2) Two machine cycle instruction output timing 211
5.11 Port Data Manipulating Instructions 212

6. ELECTRICAL CHARACTERISTICS

6.1 Absolute Maximum Ratings 216
6.2 Operational Ranges. 216
6.3 DC Characteristics 217
6.4 External Program Memory Access AC Characteristics 221
6.5 External Data Memory Access AC Characteristics 223
6.6 Serial Port (I/O Extension Mode) AC Characteristics 225
6.7 AC Characteristics Measuring Conditions 227
6.8 XTAL1 External Clock Input Waveform Conditions 228

7. DESCRIPTION OF INSTRUCTIONS

7.1 Outline 231
7.2 Description of Instruction Symbols 232
7.3 List of Instructions. 233
7.4 Simplified Description of Instructions 234
7.5 Detailed Description of MSM80C154S/MSM83C154S Instructions 246

1. INTRODUCTION

1. INTRODUCTION

1.1 MSM80C154S/MSM83C154S/MSM85C154HVS Outline

MSM80C154S/MSM83C154S/MSM85C154HVS are single-chip 8-bit fully static microcontrollers featuring high performance and low power consumption. All MSM80C31F /MSM80C51F instructions and functions have been retained.
Apart from being without the internal program memory (ROM), MSM80C154S is identical to MSM83C154S. And the difference between MSM85C154HVS and MSM83C154S is that the internal program memory (ROM) in MSM83C154S is replaced by an external ROM connected to MSM85C154HVS by using a piggy-back package.
While the MSM83C154S microcontroller integrates a 16384-word × 8-bit program memory (ROM) in a single chip, MSM80C154S/MSM83C154S/MSM85C154HVS all feature computer functions including a 256-word × 8-bit data memory (RAM), 32 input/ output ports, three 16-bit timer/counters, six interrupts, serial I/O, an 8-bit parallel processing circuit, and a clock generator.
The internal operation in these CPUs is based on an instruction code address method for greater efficiency. In this method, operations are specified in the instruction code (OP) section, and the objective registers are specified by part of that instruction code and the second or third byte following the code. A feature of this method is the ability to achieve several operations by simply changing the manipulation register designation in a single instruction code.
Inclusion of 8-bit multiplication and division instructions further increases the processing capacity of these CPUs.
In addition to expansion of the bit processing area, a comprehensive range of bit processing instructions has also been included. Processing operations include logical processing of the carry flag and specified bit within each register, transfer between the carry flag and specified bit in certain registers, transfer of specified bits between different registers, setting, resetting, and complement of the specified bit in each register, and execution of various bit tests within a wide area.
To make a relative jump after the execution of a bit test instruction, jumps can be made within a wide address range between –128 and +127 relative to the address of the instruction and there is no page field restriction.
The contents of specified registers can be saved in stack by using the PUSH instruction, and the saved contents can be returned from stack to a specified register by the POP instruction.
Absolute interrupt priority can be allocated to any interrupt when in priority circuit operation mode. And by controlling only the interrupt enable register (IE) when in priority circuit stop mode, multi-level interrupt processing can be executed to make interrupt processing much easier than in conventional CPUs.
Employing the low-power consumption feature of C-MOS devices, these CPUs are designed to operate in a number of “CPU power down” modes. In idle mode the IDL bit in the power control register (PCON) is set to “1” to halt CPU operations while the oscillator continues to run. In soft power down mode the PD bit in the power control register is set to “1” to halt CPU operations as well as the oscillator. And in hard power down mode where the HPD bit in the power control register is set in advance to “1”, CPU operations and the oscillator are stopped if the HPDI pin (P3.5) power failure detect signal level is changed from “1” to “0”. CPU power down modes can be cancelled by resetting the CPU via reset pin and restarting execution from address 0, by restarting execution from the relevant interrupt address, or by resuming

execution from the next address after the stop address where CPU power down mode was activated.

Each of the quasi-bidirectional ports 1, 2, and 3 can be set independently as high impedance input ports. And the 10 kΩ pull-up resistance for these input ports can be isolated from the power supply (V_{CC}), leaving only the 100 kΩ pull-up resistance and thereby enabling the quasi-bidirectional ports to be driven by devices with low drive capacity. Furthermore, the outputs of ports, 0, 1, 2, and 3 can be switched to floating status during CPU power down modes (PD, HPD).

Three built-in 16-bit timer/counters capable of operating in a wide range of modes enable the CPUs to be used in many different ways. And since timer/counters 0 and 1 can be operated by external clock during CPU power down modes (PD, HPD) where the oscillator is stopped, these two counters can also be used in cancelling CPU power down modes.

UART based serial communication can be executed at any baud rate by carry signal from timer/counter 1 or timer/counter 2.

If an overrun or framing error is generated during data reception, the SERR bit in the I/O control register is set. And by testing this SERR bit, the accuracy of the data can be checked quite easily to ensure correct serial communication.

As can be seen, these CPUs are equipped with a very comprehensive range of functions. Also note that EASE80C51mkII is available for use as the program development support system for these CPUs.

Equipped with the MSM85C154E dedicated evachip, EASE80C51mkII is capable of program area mapping, realtime tracing, generating breaks according to accumulator contents, and various other functions designed for accurate and efficient support of program development of these CPUs.

With this great line-up of functions and with EASE80C51mkII capable of developing programs in a very short time, MSM80C154S/MSM83C154S/MSM85C154HVS give a highly integrated high performance solution.

1.2 MSM80C154S/MSM83C154S Features

- Full static circuitry
- Internal program memory (ROM)
  16384 words × 8 bits (MSM83C154S)
- External program memory (ROM)
  Connectable up to 64K bytes
- Internal data memory (RAM)
  256 words × 8 bits
- External data memory (RAM)
  Connectable up to 64K bytes
- Four sets of working registers (R0 thru R7 × 4)
- Stack
  Free use of 256-word × 8-bit internal data memory area
- Four input/output ports (8-bit × 4)
- Serial ports (UART operation)
- Six types of interrupts
  (1) Two external interrupts
  (2) Three timer interrupts
  (3) One serial port interrupt
    * Priority allocated interrupt processing
    * Multi-level interrupt processing by software management
- CPU power down function
  (1) Idle mode: CPU stopped while oscillation continued.
  (Software setting)
  (2) PD mode: CPU and oscillation all stopped.
  (Software setting)
  (Setting I/O ports to floating status possible)
  (3) HPD mode: CPU and oscillation all stopped.
  (Hardware setting)
  (Setting I/O ports to floating status possible)
- CPU power down mode cancellation
  (1) Execution commenced from address 0 by CPU resetting.
  (IDLE, PD, and HPD mode cancellation)
    * RESET pin is used
  (2) Execution from interrupt address by interrupt request, or execution resumed from next address after the stop address. (IDLE and PD mode cancellation)
    * External, timer, and serial port interrupts
- I/O control registers (0F8H)
  b_0: Port 0, 1, 2, and 3 floating setting (PD, HPD)
  b_1: Port 1 high impedance input port setting
  b_2: Port 2 high impedance input port setting
  b_3: Port 3 high impedance input port setting
  b_4: Port 1, 2, and 3 pull-up resistance switching (10 kΩ pull-up resistance switch off to leave only 100 kΩ)
  b_5: Serial port reception error detector bit
  b_6: 32-bit timer mode setting (TL0+TH0+TL1+TH1)

- Timer/counters (three 16-bit timer/counters)
  (1) 8-bit timer with 5-bit prescalar
  (2) 16-bit timer
  (3) 8-bit timer with 8-bit auto-reloader
  (4) 8-bit separate timer
  (5) 16-bit timer with 16-bit auto-reloader
  (6) 16-bit capture timer
  (7) 16-bit baud rate generator timer
  (8) 32-bit timer
- Wide operating temperature range –40 to +85°C
- Wide operating voltage range
  (1) When operating: V_{CC}=+2.2 to 6V (varies according to frequency)
  (2) When stopped:
  V_{CC}=+2 to +6V (PD or HPD mode)
- Instruction execution cycle
  (1) 2-byte 1-machine cycle instructions
  (2) Multiplication/division instructions
- Direct initialization of ports 0, 1, 2, and 3 by input of reset signal even if oscillator have been stopped.
  (All ports output “1”.)
- High noise margin (with Schmitt trigger input for each I/O)
- 40-pin plastic DIP/44-pin plastic flat package/44-pin plastic PLCC/44/pin plastic TQFP
- Software compatibility with MSM80C31F and MSM80C51F

1.3 Additional Features in MSM80C154S/MSM83C154S/MSM85C154HVS

In addition to the basic operations of MSM80C31F/MSM80C51F, the MSM80C154S/MSM83C154S/MSM85C154HVS devices also include the following functions.

- ROM capacity increased from 4K bytes to 16K bytes
- RAM capacity increased from 128 bytes to 256 bytes
- An additional timer counter 2
- An additional timer interrupt 2
- An additional 8-bit timer 2 control register (T2CON 0C8H)
- An additional 8-bit I/O control register (IOCON 0F8H)
- Addition of two bits (bit 5, PT2 and bit 7, PCT) to the priority register (IP 0B8H)
- Addition of one bit (bit 5, ET2) to the interrupt enable register (IE 0A8H)
- Addition of two bits (bit 5, RPD and bit 6, HPD) to the power control register (PCON 87H)

Addition of these extra functions has further increased the performance and widen the range of application of these CPU devices.

2. SYSTEM CONFIGURATION

2. SYSTEM CONFIGURATION

2.1 MSM80C154S/MSM83C154S/MSM85C154HVS Logic Symbols

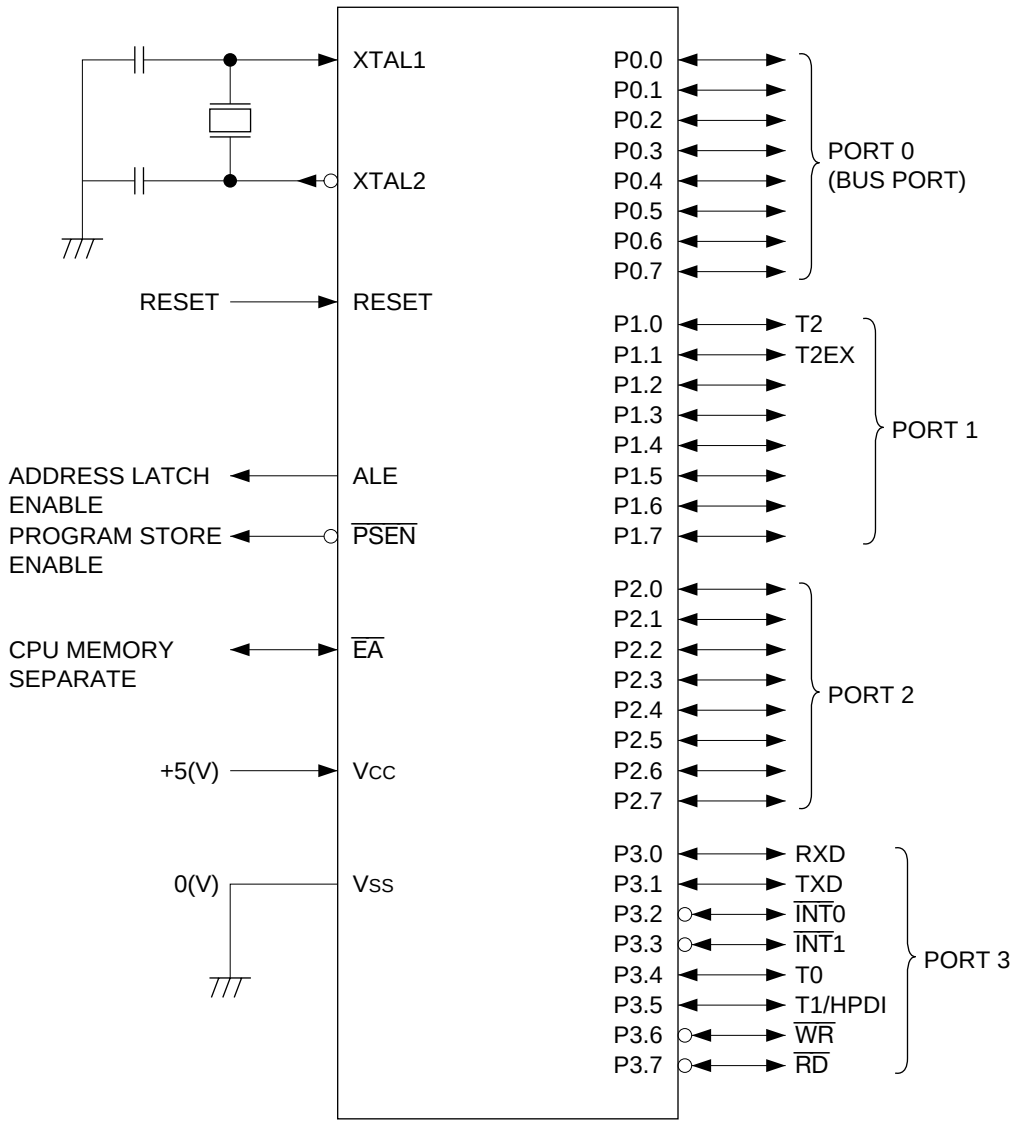
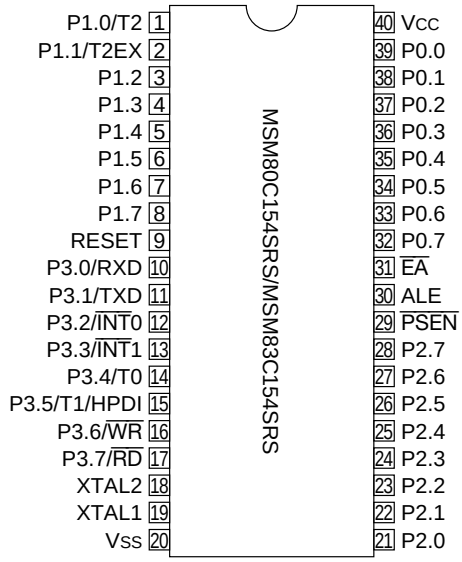


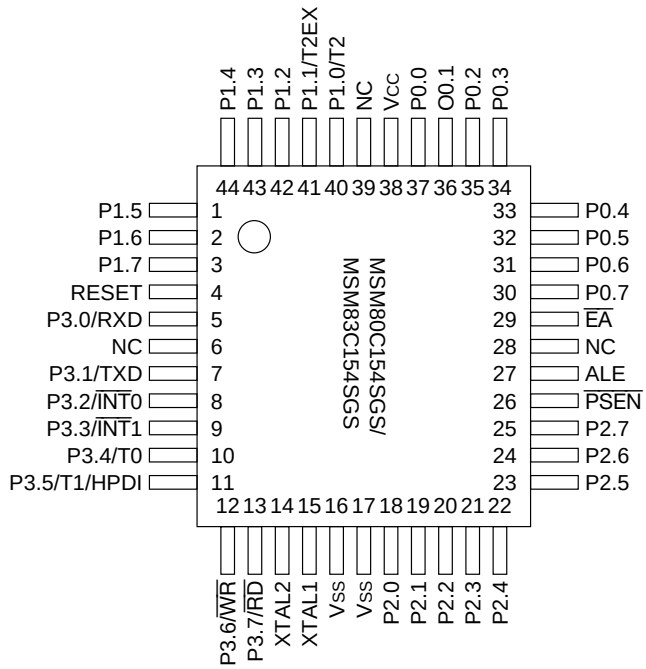
Figure 2-1 MSM80C154S/83C154S/85C154HVS logic symbols

2.2 MSM80C154S/MSM83C154S pin layouts

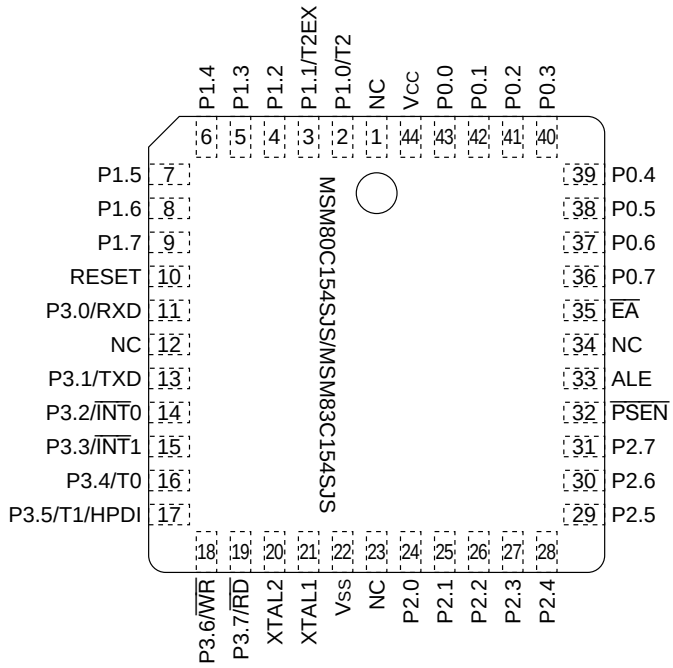
MSM80C154SRS/MSM83C154SRS
(Top View) 40 Pin Plastic DIP



MSM80C154SGS/MSM83C154SGS
(Top View) 44 Pin Plastic Package



MSM80C154SJS/MSM83C154SJS
(Top View) 44 Pin Plastic QFJ



MSM80C154STS/MSM83C154STS
(Top View) 44 Pin Plastic Package

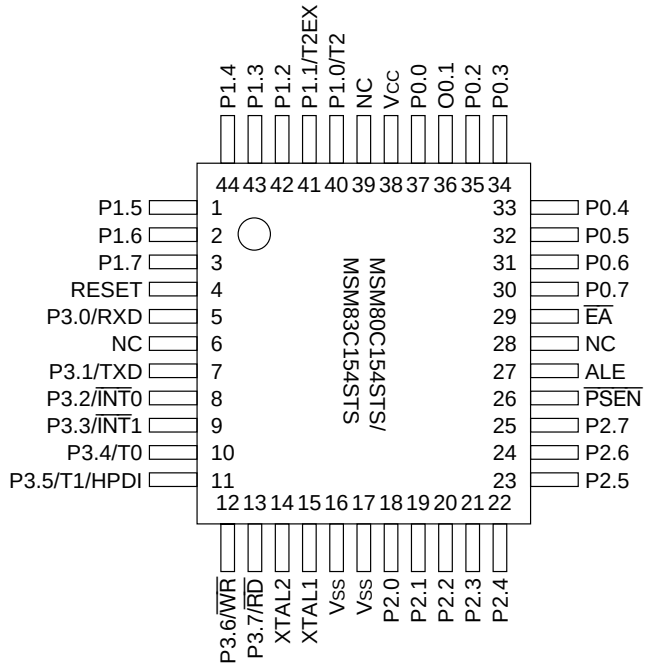


Figure 2-2 MSM80C154S/MSM83C154S pin layout (top view)

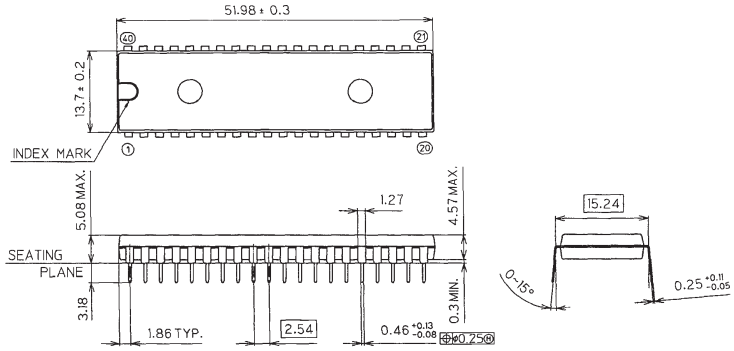
Applicable Packages

40-Pin Plastic DIP (DIP40-P-600-2.54)	MSM80C154S RS MSM83C154S-XXX RS
44-Pin Plastic QFJ (QFJ44-P-S650-1.27)	MSM80C154S JS MSM83C154S-XXX JS
44-Pin Plastic QFP (DFP44-P-910-0.80-2K)	MSM80C154S GS-2K MSM83C154S-XXX GS-2K
44-Pin Plastic TQFP (TQFP44-P-1010-0.80-K)	MSM80C154S TS-K MSM83C154S-XXX TS-K
40-Pin Ceramic Piggy Back (ADIP40-C-600-2.54)	MSM85C154HVS

2.2.1 MSM80C154S/MSM83C154S external dimensions

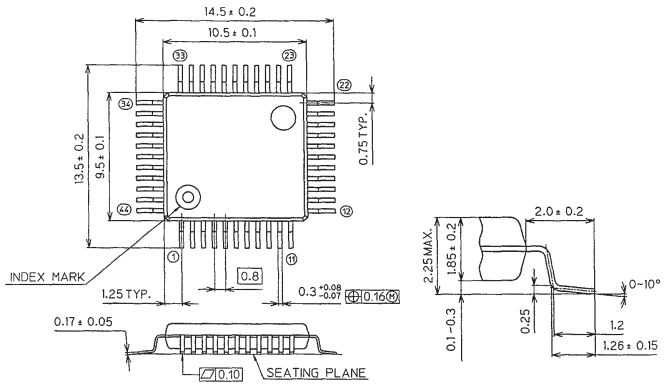
MSM80C154SRS/MSM83C154SRS

40-pin Plastic DIP (DIP40-P-600-2.54)



MSM80C154SGS/MSM83C154SGS

44-Pin Plastic QFP (QFP44-P-910-0.80-2K)



MSM80C154SJS/MSM83C154SJS

44-Pin Plastic QFJ (QFJ44-P-S650-1.27)

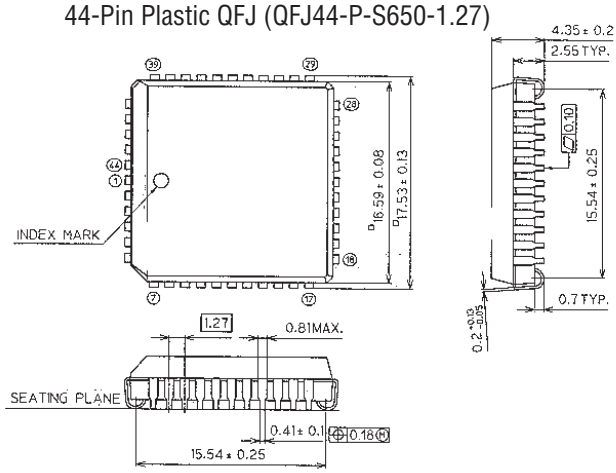
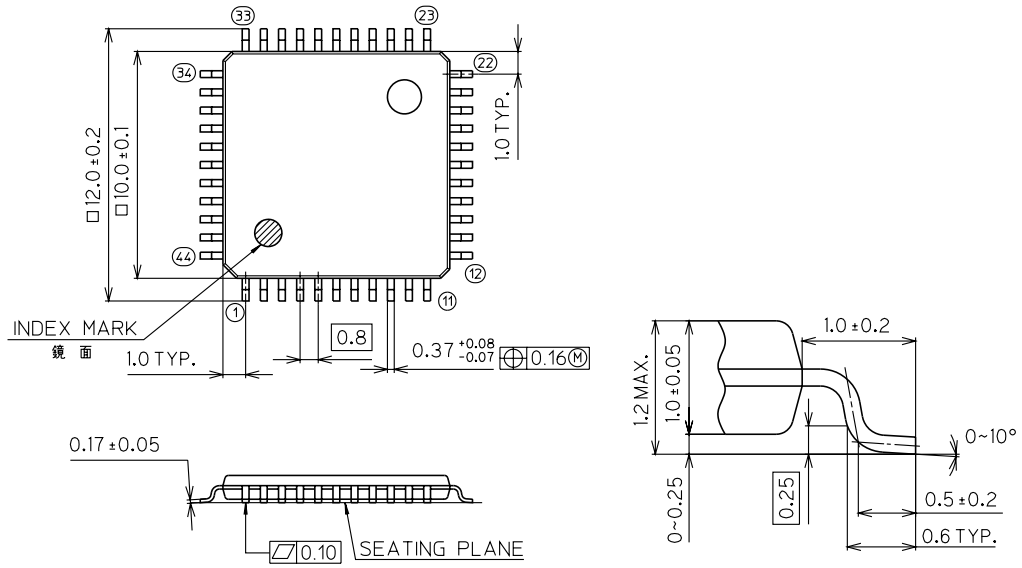


Figure 2-3 MSM80C154S/MSM83C154S external dimensions

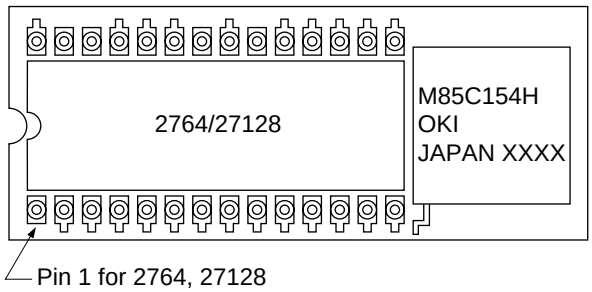
MSM80C154S/83C154S/85C154HVS

MSM80C154STS/MSM83C154STS

44-Pin Plastic TQFP (TQFP44-P-1010-0.80-K)



2.2.2 MSM85C154HVS pin layout and external dimensions



* The MSM85C154HVS pin layout of bottom side is the same as the pin layout for MSM83C154SRS.

* The 27C64/128 device should be used for EPROM.

40-Pin Ceramic Piggy Back (ADIP40-C-600-2.54)

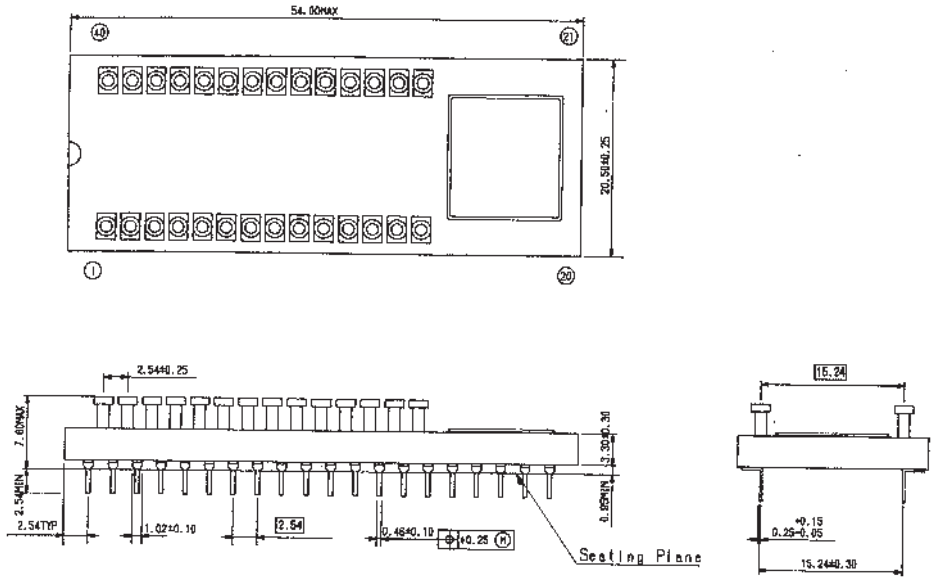


Figure 2-4 MSM85C154HVS pin layout and external dimensions

2.3 MSM80C154S Block Diagram

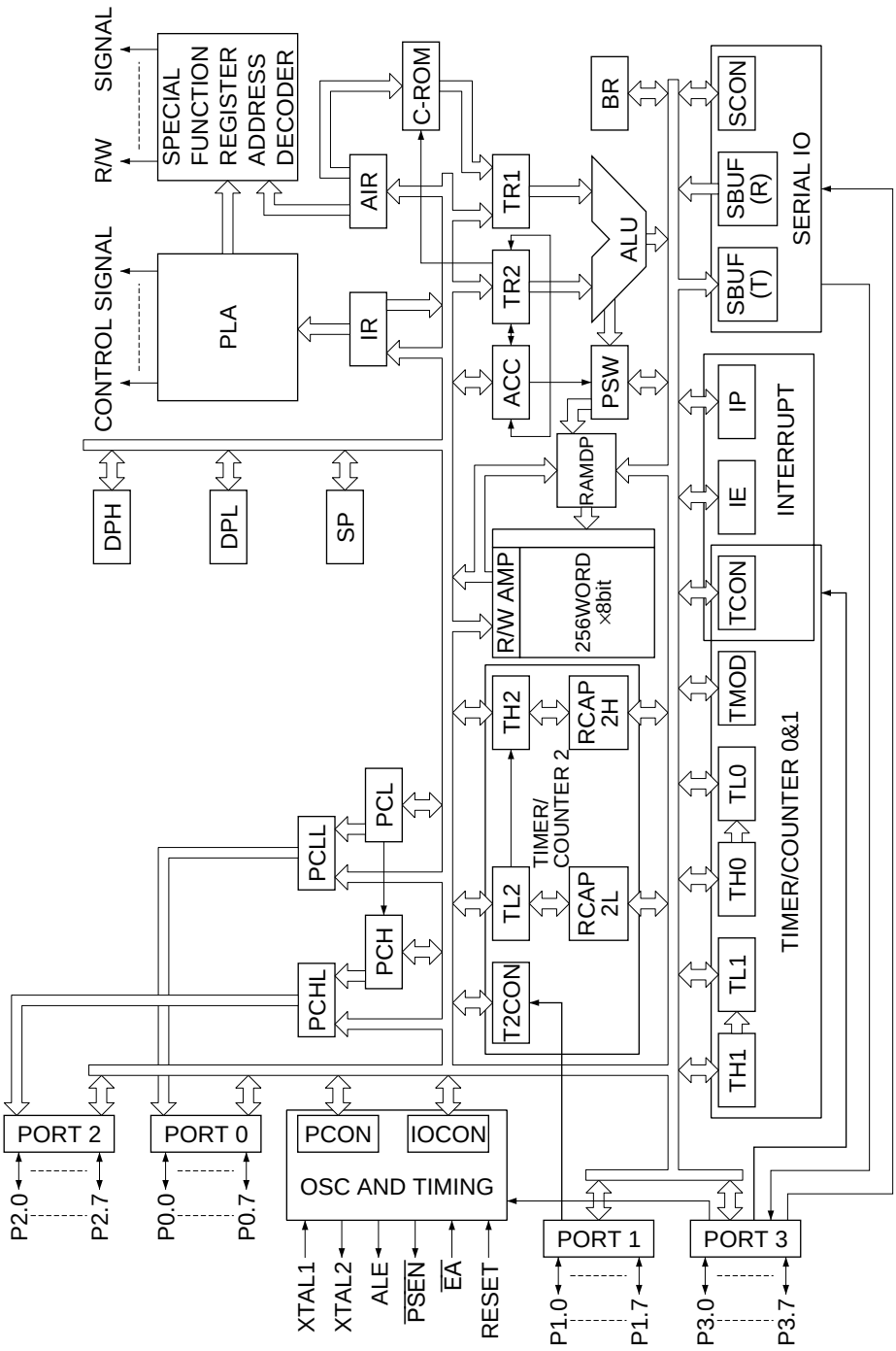


Figure 2-5 MSM80C154S block diagram

2.4 MSM83C154S Block Diagram

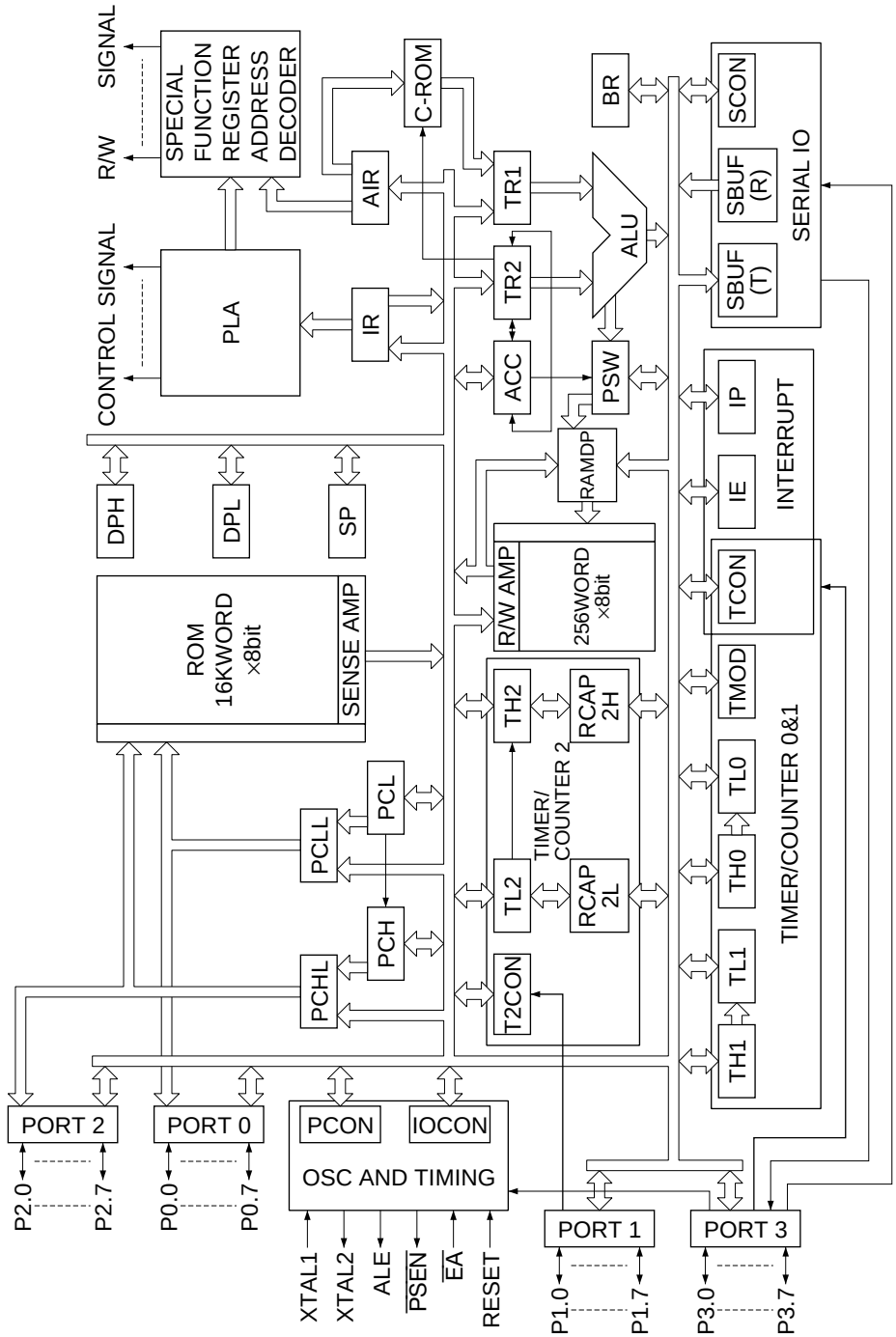


Figure 2-6 MSM83C154S block diagram

2.5 MSM85C154HVS Block Diagram

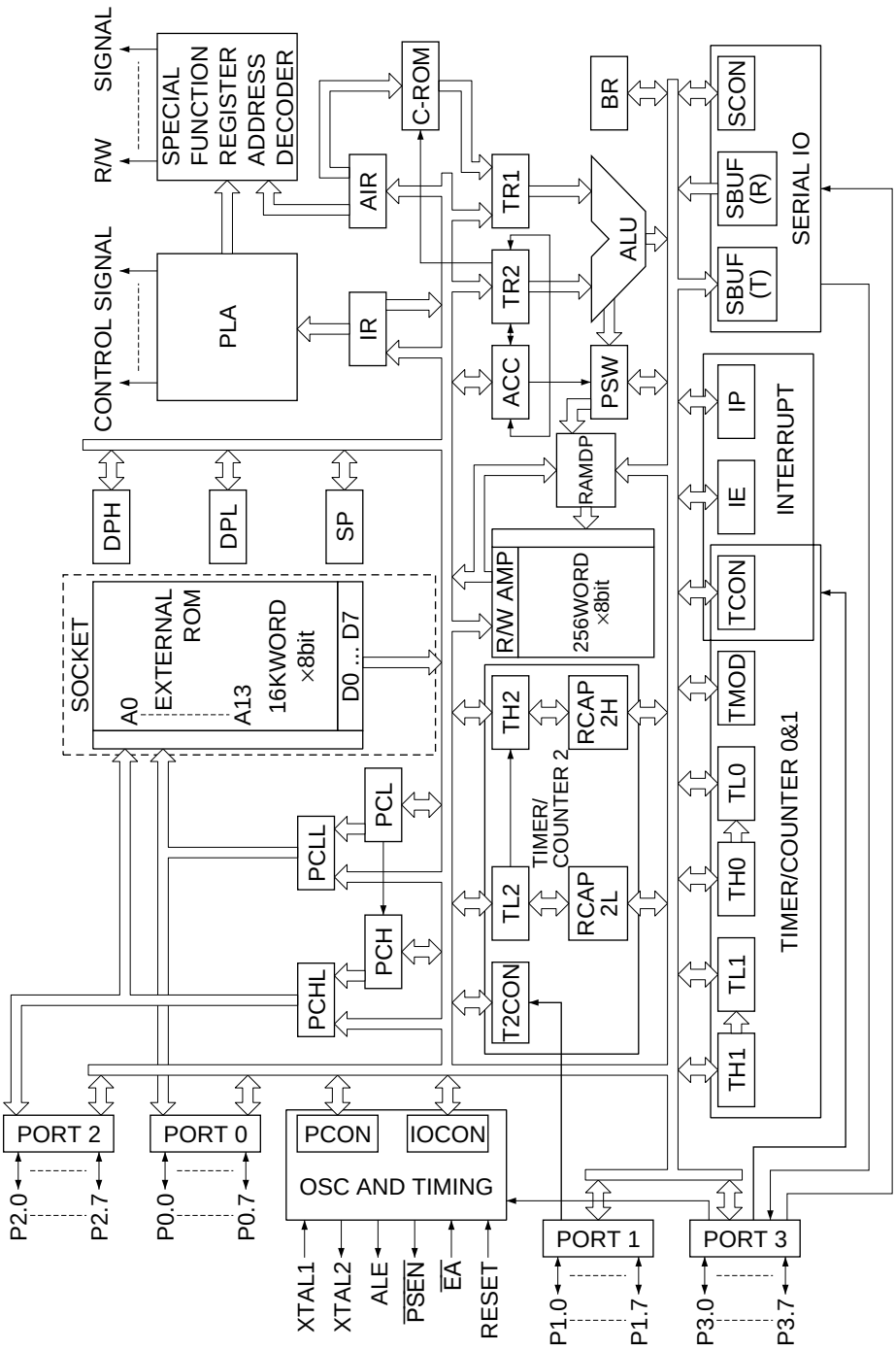


Figure 2-7 MSM85C154HVS block diagram

2.6 Timing and Control

2.6.1 Outline of MSM80C154S/MSM83C154S timing

The MSM80C154S/MSM83C154S devices are both equipped with a built-in oscillation inverter (see Figure 2-8) for use in the generation of clock pulses by external crystal or ceramic resonator. These clock pulses are passed to the timing counter and control circuits where the basic timing and control signals required for internal control purposes are generated.
The basic timing consists of state 1 (S1) thru state 6 (S6) (see Figure 2-9) where each state cycle is based on two XTAL1·2 fundamental clock pulses. The interval from S1 thru S6 forms a single machine cycle with a total of 12 fundamental clock pulses. 1-byte 1-machine cycle and 2-byte 1-machine cycle instructions are fetched into the instruction register during M1·S1, decoded during M1·S2, and executed during M1·S3 thru M1·S6. The second byte is fetched during M1·S4. 1-byte 2-machine cycle, 2-byte 2-machine cycle, and 3-byte 2-machine cycle instructions are also fetched during M1·S1, decoded during M1·S2, and executed during M1·S3 thru M2·S6. The second and third bytes are fetched during M1·S4, M2·S1, or M2·S4. The number of clocks used is 24. 1-byte 4-machine cycle instructions are involved in multiplication and division operations where 48 clocks are used.

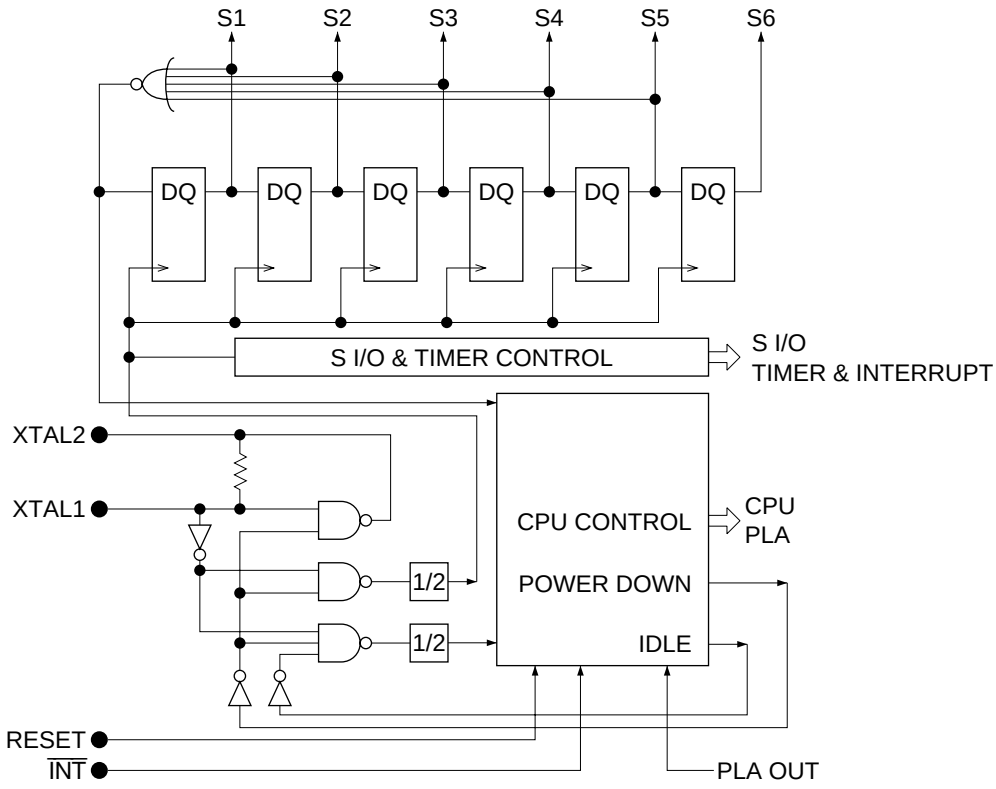


Figure 2-8 Oscillator, timing counter, and control stage block diagram

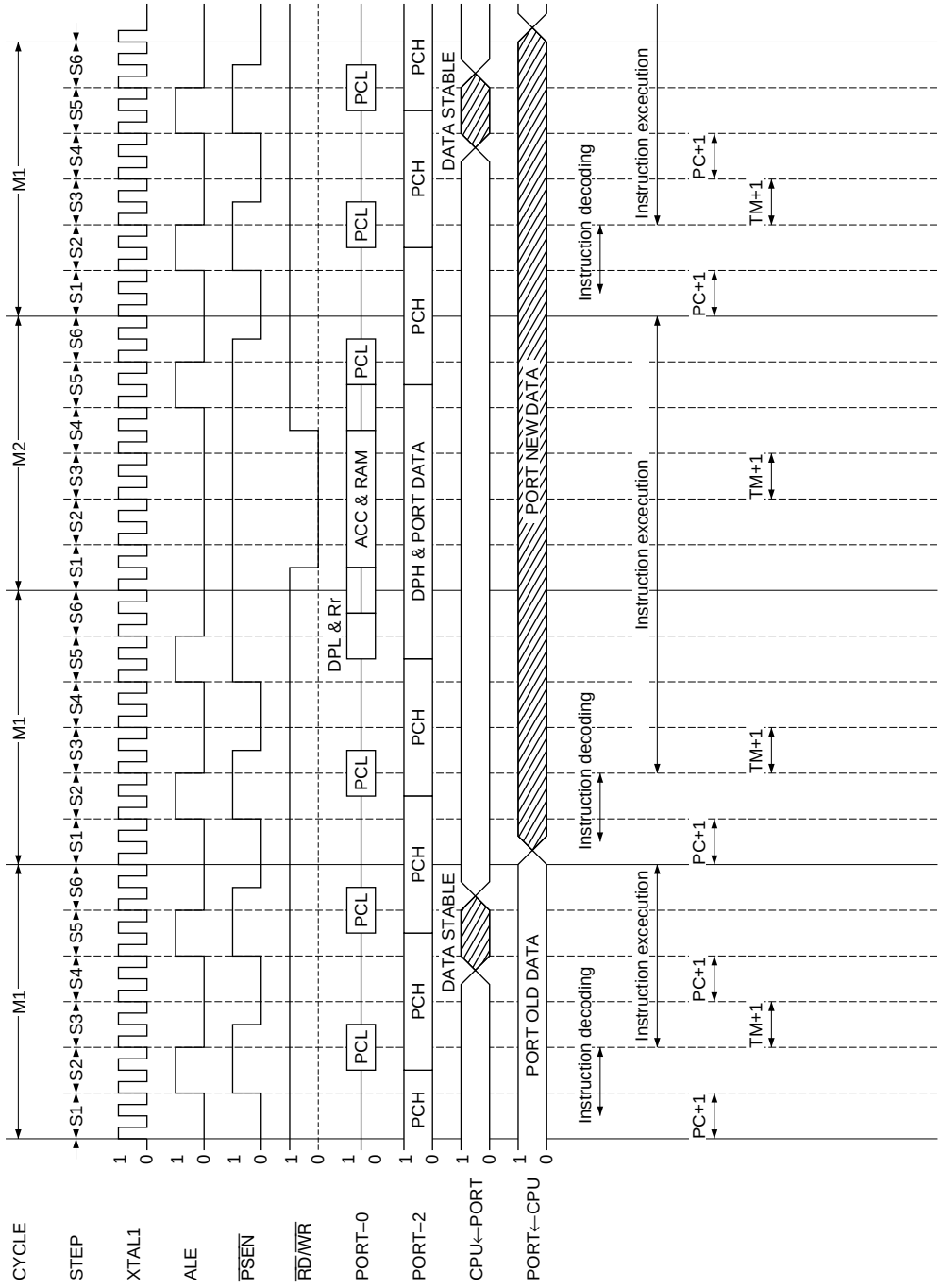


Figure 2-9 MSM80C154S/MSM83C154S fundamental timing

2.6.2 Major synchronizing signals

(1) ALE (Address Latch Enable)
The ALE signal is used as a clock signal where the address signals 0 thru 7 output from CPU port 0 can be latched externally when external program or external data memory (RAM) is used.
Although two ALE signal outputs are obtained in a single machine cycle during normal operations, no output is obtained during output of the $\overline{RD}$/$\overline{WR}$ signal when an external memory instruction (MOVX.....) is executed.

(2) $\overline{PSEN}$ (Program Store Enable)
The $\overline{PSEN}$ output signal is generated during execution of an external program. The output is obtained when an instruction or data is fetched.
The $\overline{PSEN}$ signal is valid when at "0" level, and external program data is enabled when in this valid state.
Although two $\overline{PSEN}$ signal outputs are obtained in a single machine cycle during normal operations, no output is obtained during output of the $\overline{RD}$/$\overline{WR}$ signal when an external data memory instruction (MOVX.....) is executed.

(3) $\overline{WR}$ (Write Strobe)
The $\overline{WR}$ output signal is obtained when an external data memory instruction (MOVX @Rr, A or MOVX @ DPTR, A) is executed.
CPU port 0 output data is written in the external RAM when the $\overline{WR}$ signal is at "0" level.

(4) $\overline{RD}$ (Read Strobe)
The $\overline{RD}$ output signal is obtained when an external data memory instruction (MOVX A, @ Rr or MOVX A, @ DPTR) is executed.
The external RAM is enabled and output data is passed to CPU port 0 when the $\overline{RD}$ signal is at "0" level.

2.6.3 MSM80C154S fundamental operation time charts

(1) External program memory read cycle timing chart

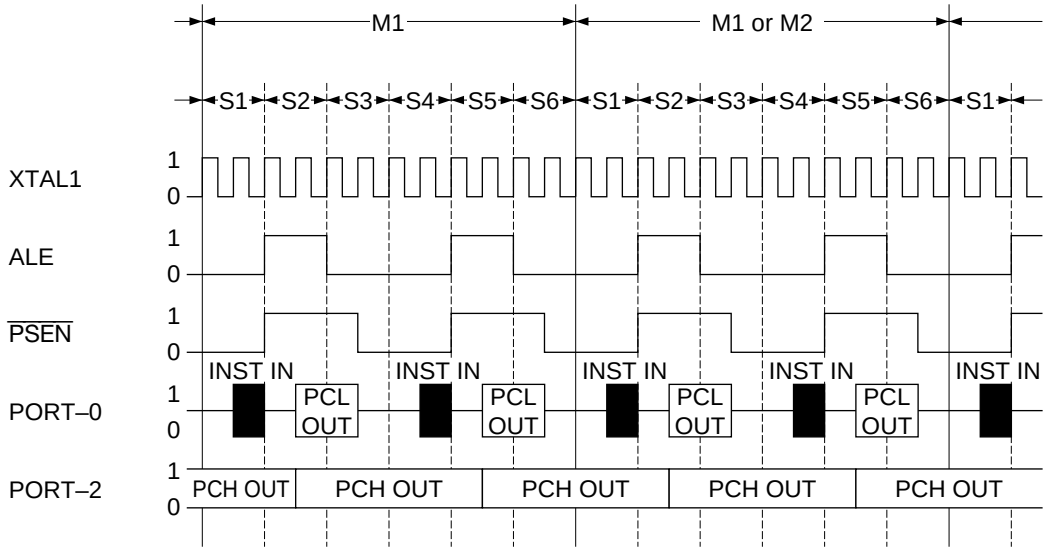


Figure 2-10 MSM80C154S external program memory read cycle timing chart

(2) MOVX A, @Rr

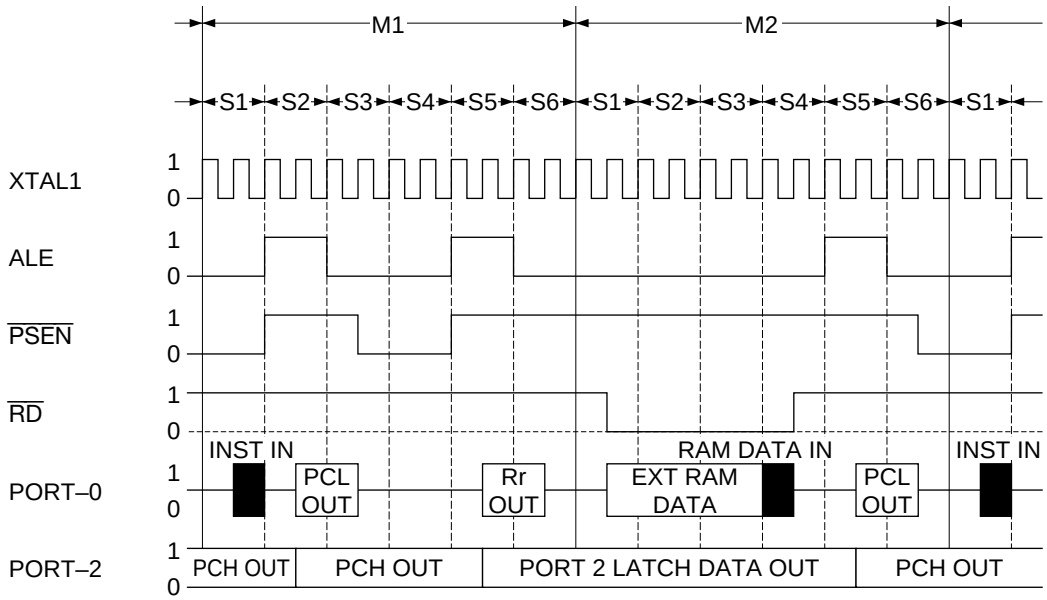


Figure 2-11 MSM80C154S MOVX A, @Rr execution

(3) MOVX @Rr, A

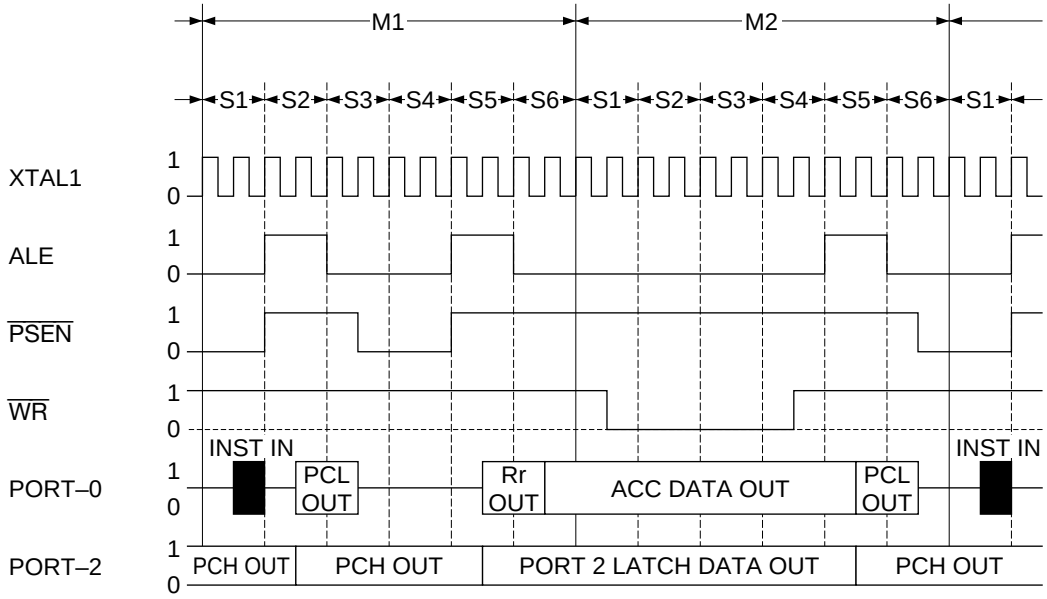


Figure 2-12 MSM80C154S MOVX @Rr, A execution

(4) MOVX A, @DPTR

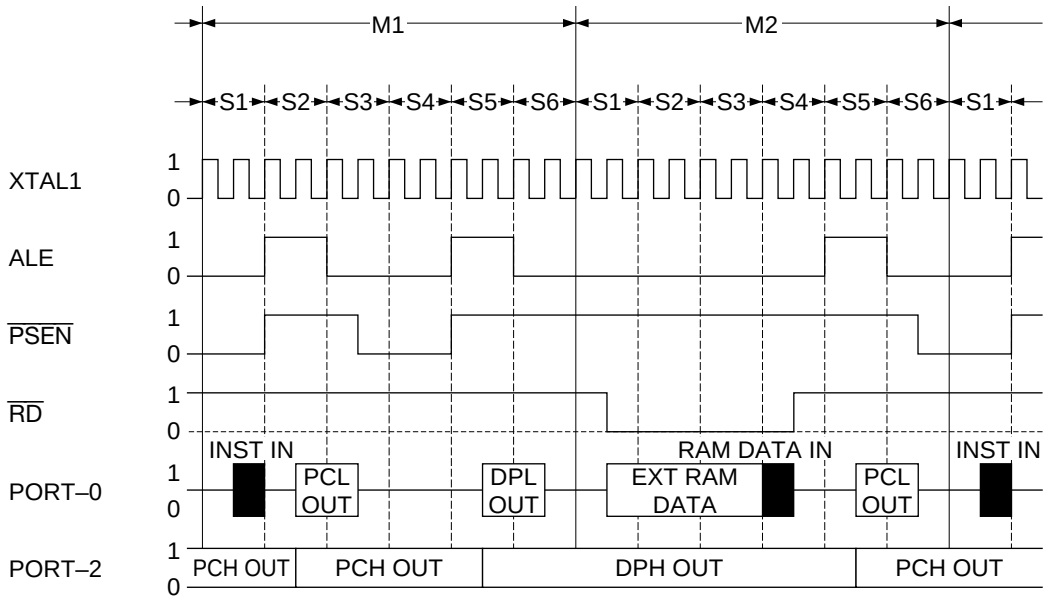


Figure 2-13 MSM80C154S MOVX A, @DPTR execution

(5) MOVX @DPTR, A

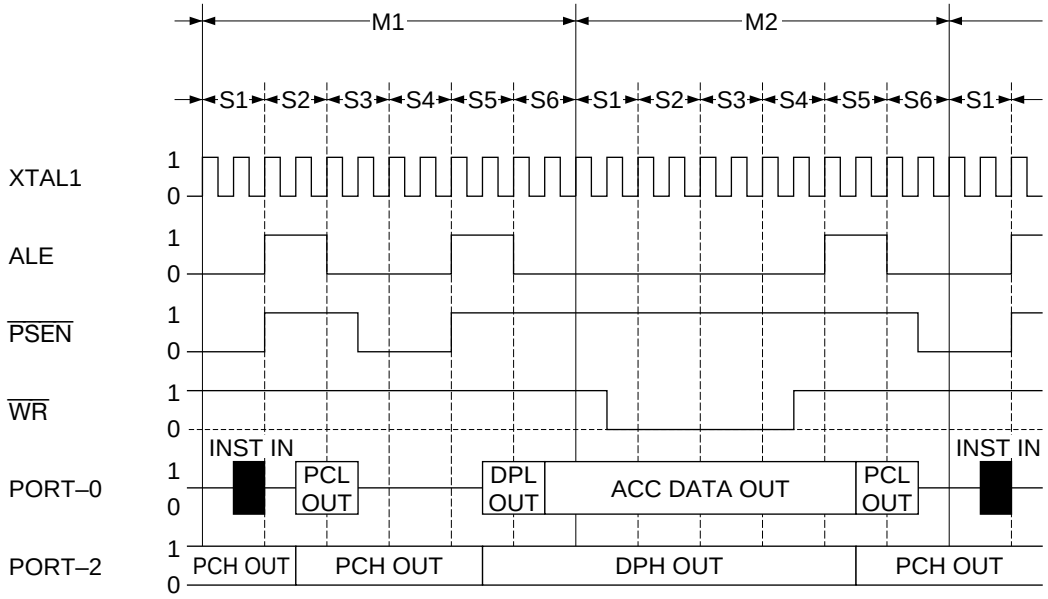


Figure 2-14 MSM80C154S MOVX @DPTR, A execution

(6) MOV direct, PORT [0, 1, 2, 3] execution

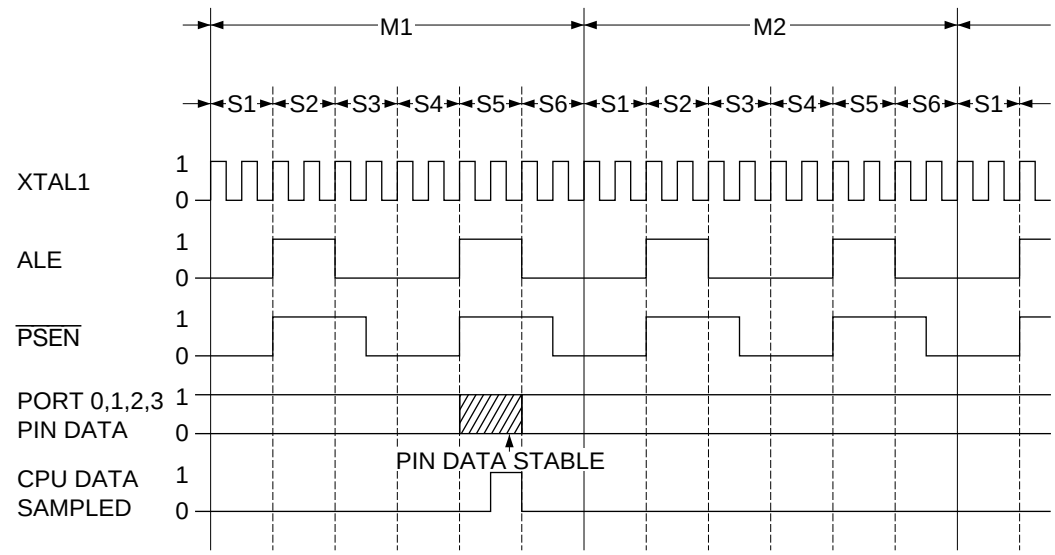


Figure 2-15 MSM80C154S MOV direct, PORT[0, 1, 2, 3] execution

2.6.4 MSM83C154S fundamental operation time charts

(1) MOVX A, @Rr

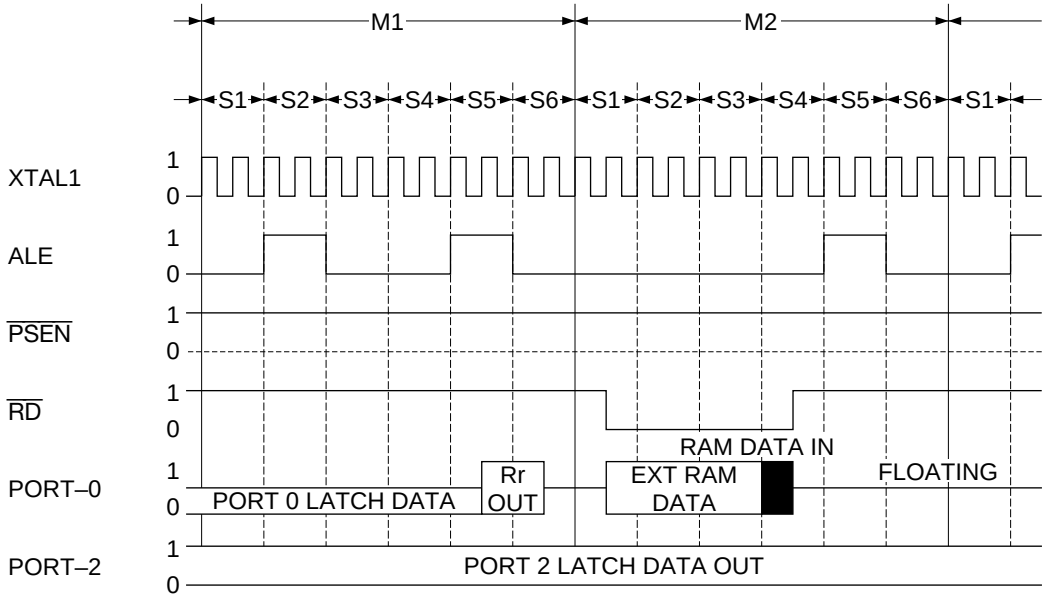


Figure 2-16 MSM83C154S MOVX A, @Rr execution

(2) MOVX @Rr, A

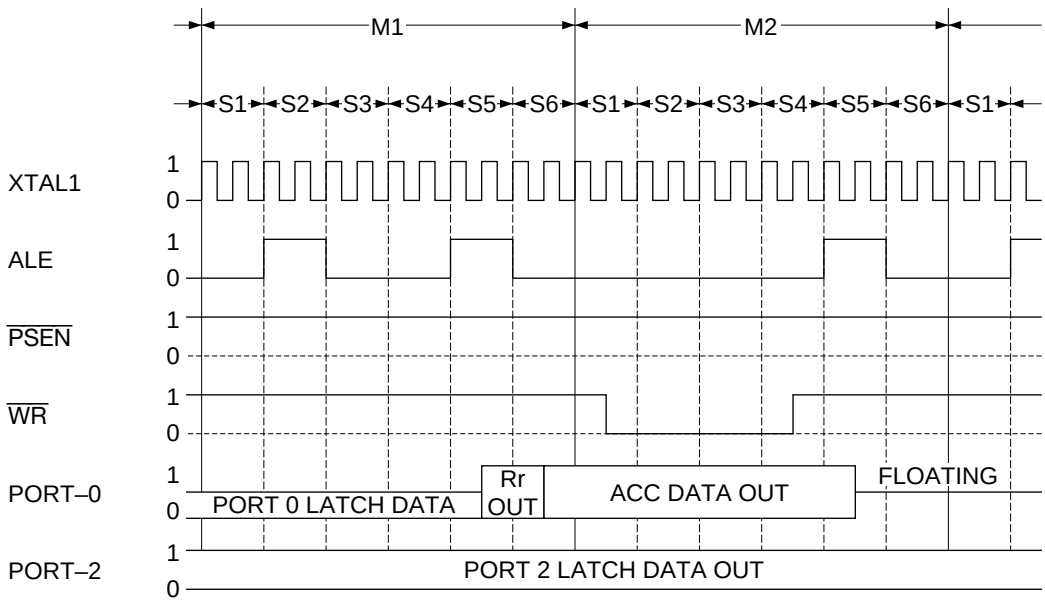


Figure 2-17 MSM83C154S MOVX @Rr, A execution

(3) MOVX A, @DPTR

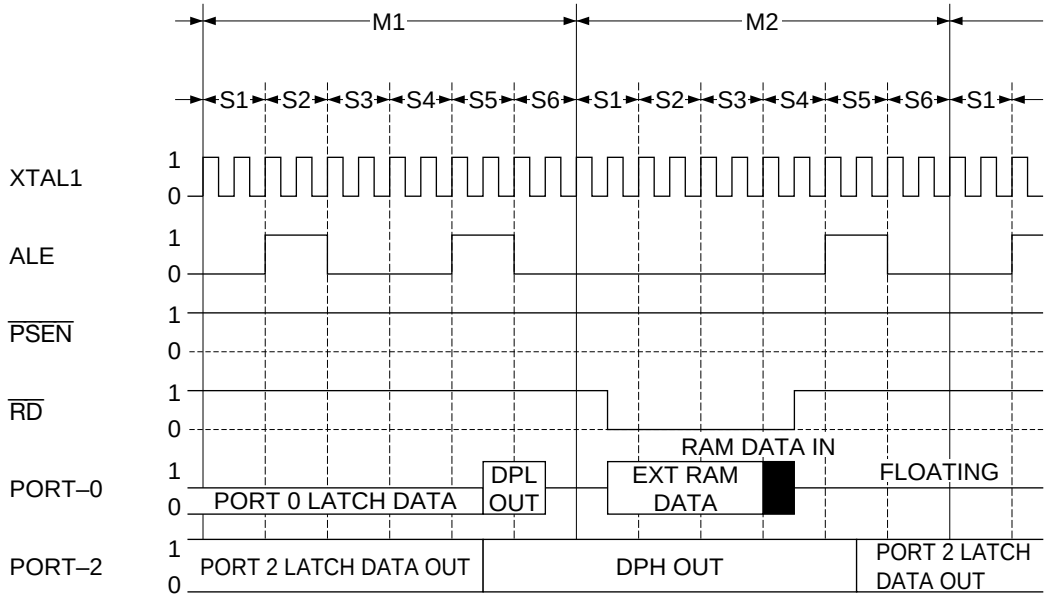


Figure 2-18 MSM83C154S MOVX A, @DPTR execution

(4) MOVX @DPTR, A

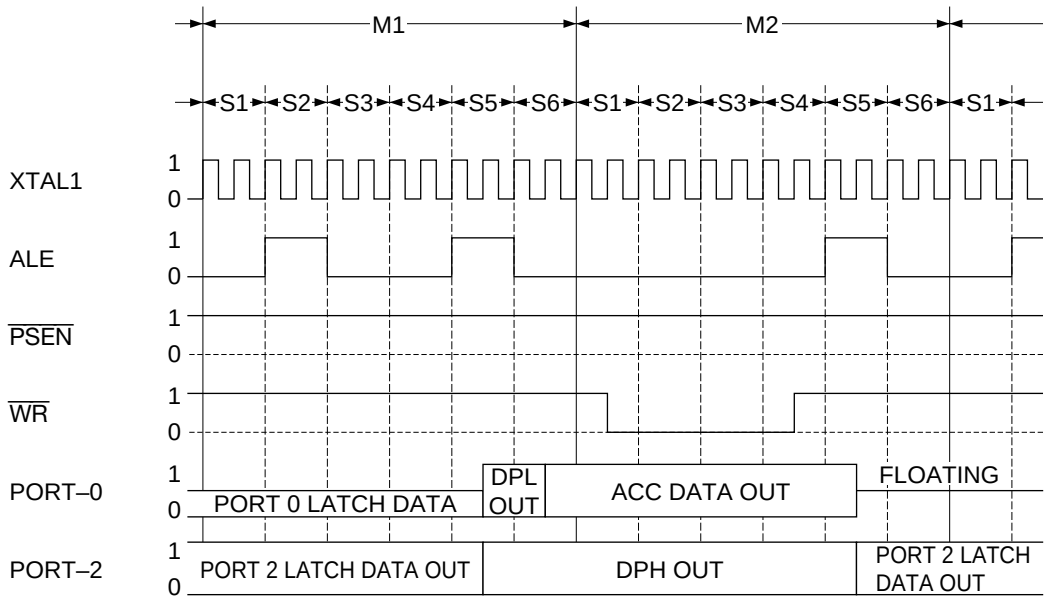


Figure 2-19 MSM83C154S MOVX @DPTR, A execution

(5) MOV direct, PORT [0, 1, 2, 3] execution

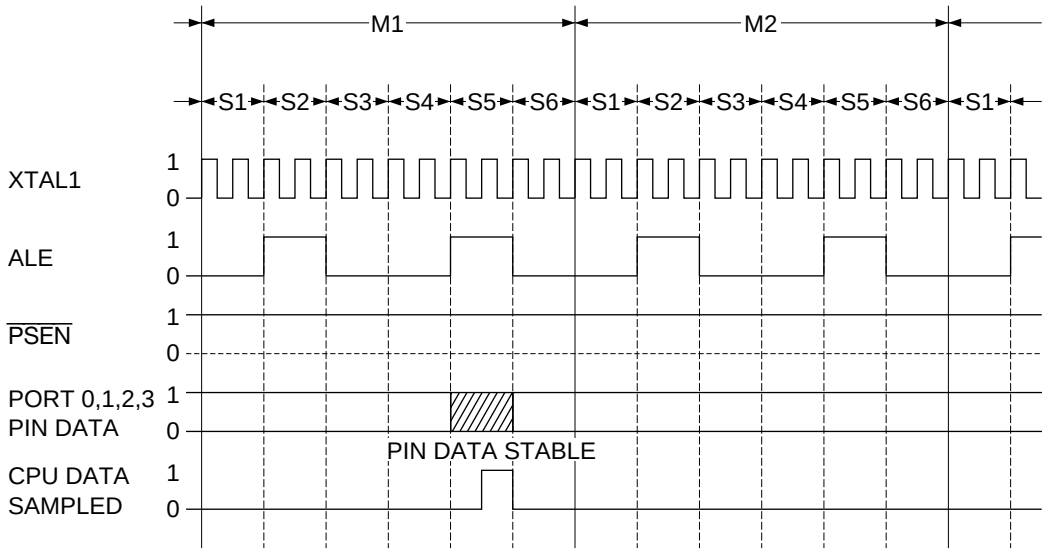


Figure 2-20 MSM83C154S MOV direct, PORT[0, 1, 2, 3] execution

2.7 Instruction Register (IR) and Instruction Decoder (PLA)

MSM80C154S/MSM83C154S operations are based on an instruction code address method. Hence, in addition to the instruction code instruction register (IR) and instruction decoder (PLA), these devices also include an instruction register (AIR) and register manipulation decoder (PLA) for data addresses and bit addresses.
Operation codes are passed to the IR, and data and bit addresses are passed to the AIR. CPU control signals are formed at the respective PLA for each instruction register, thereby activating the CPU. The block diagram is outlined in Figure 2-21.

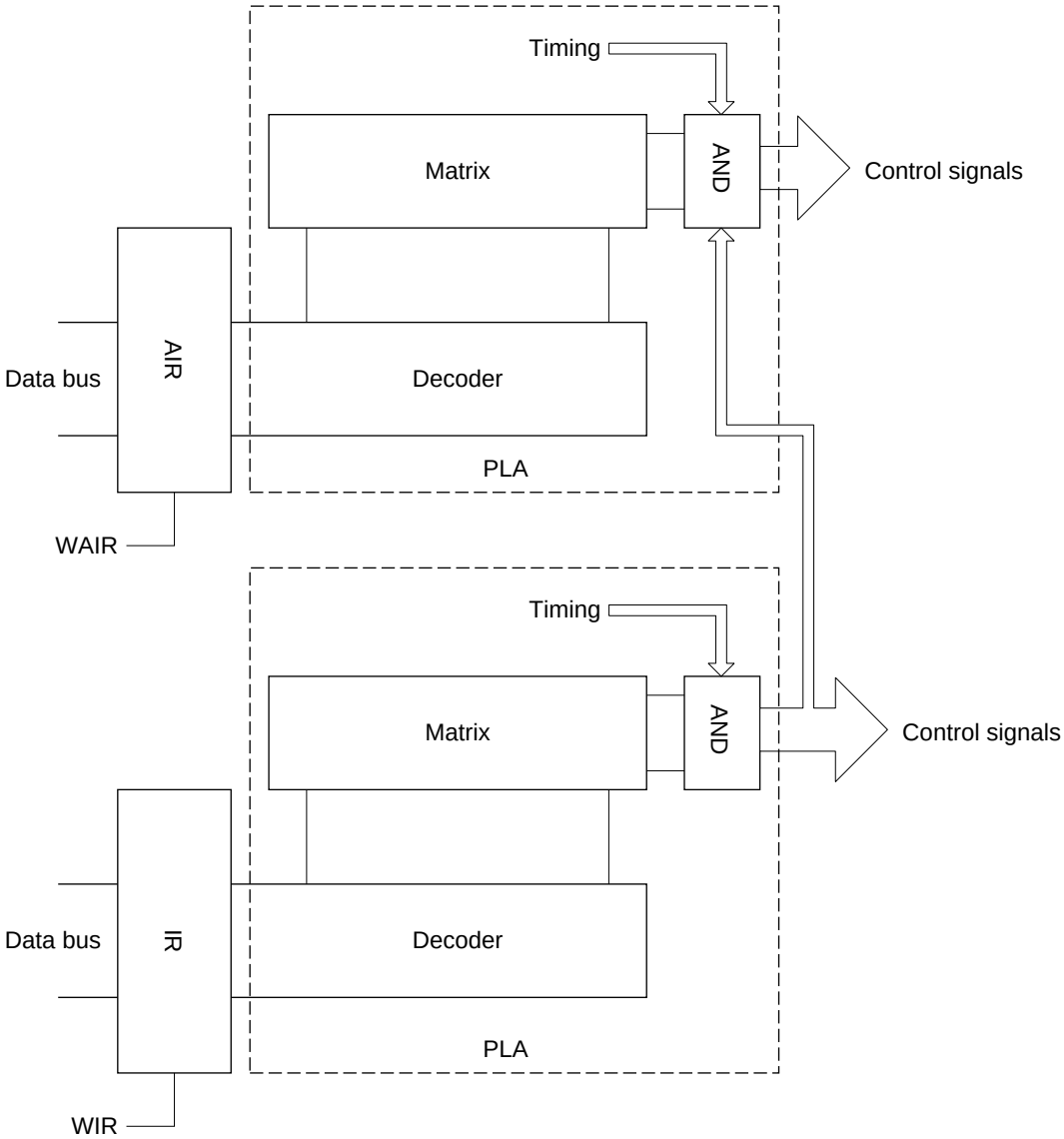


Figure 2-21 IR and PLA block diagram

2.8 Arithmetic Operation Section

(1) Outline
The MSM80C154S/MSM83C154S arithmetic operation section consists of
(1) an arithmetic operation instruction decoder, and
(2) an arithmetic and logic unit [ALU].

(2) Arithmetic operation instruction decoder:
Arithmetic operation instructions are passed to the instruction register (IR) and then to the PLA where they are converted into control signals.
The control signals from the PLA are used to control ALU peripheral circuits and ALU arithmetic operations (ADD, AND, OR, EOR).

(3) Arithmetic and logic unit [ALU]:
Upon reception of 8-bit data from one or two data sources the ALU processes that data in accordance with control signals from the PLA. The ALU is capable of executing the following processes:
- Additions and subtractions with and without carry
- Increments (+1) and decrements (–1)
- Bit complements
- Rotations (either direction with and without carry)
- BCD (decimal adjust)
- Carry, auxiliary carry, and overflow signal output
- Multiplications and divisions
- Bit detection
- Exchange of low and high order nibbles
- Logical AND, logical OR, and exclusive OR

If a bit-3 auxiliary carry (AC), a bit-7 carry (CY), or an overflow (OV) is generated as a result of the arithmetic operation executed by the ALU, that result is set in the program status word (PSW 0D0H).

PSW(0D0H)

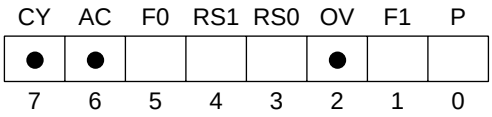


Figure 2-22 Program status word

2.9 Program Counter

The MSM80C154S/MSM83C154S program counter has a 16-bit configuration PC_0 thru PC_{15}, as shown in Figure 2-23.

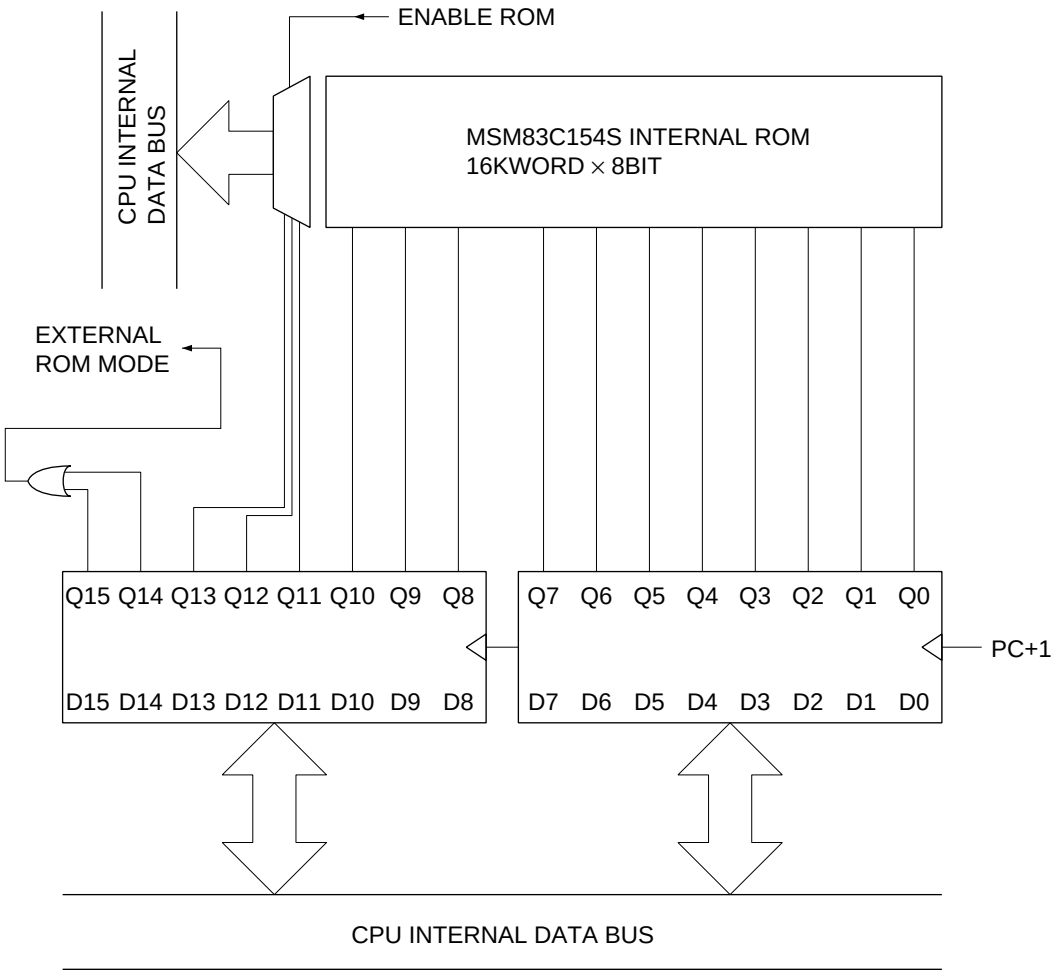


Figure 2-23 MSM80C154S/MSM83C154S program ounter

This program counter is a binary up-counter which is incremented by 1 each time one byte of instruction code is fetched. When the program counter is counted by 1 after counter contents have reached 0FFFFH, the counter is returned to 0000H. MSM83C154S is automatically switched to external ROM mode when the counter contents exceed 3FFFH.

2.10 Program Memory and External Data Memory

2.10.1 MSM80C154S/MSM83C154S program area and external ROM connections

Since MSM80C154S/MSM83C154S are equipped with a 16-bit program counter, these devices can execute programs of up to 64K bytes (including both internal and external programs).
Since the MSM80C154S is not equipped with an internal program ROM, however, only external instructions are executed. MSM83C154S, on the other hand, is equipped with a 16K byte program ROM which enables it to execute internal instructions from address 0 thru address 16383. External instructions are executed when the address is greater than 16383. The program area is outlined in Figure 2-24, and a diagram of ROM connections made when external instructions are executed is shown in Figure 2-25.

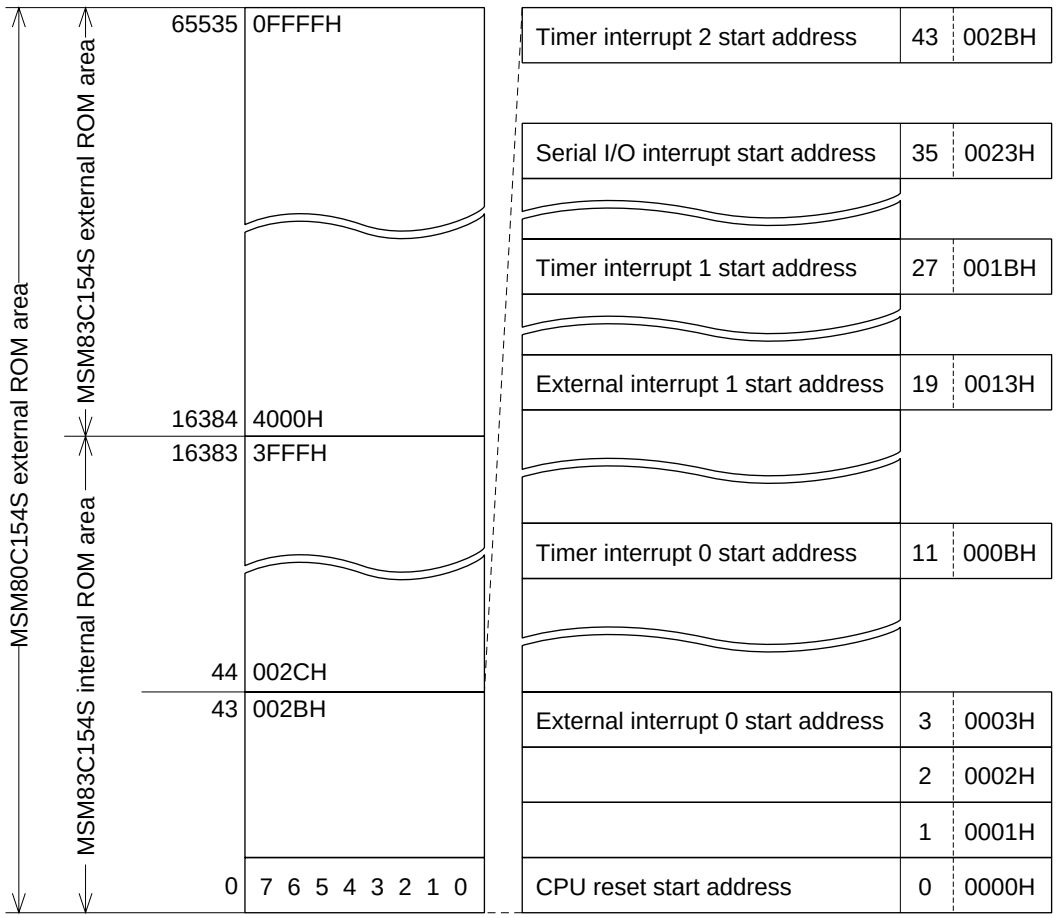


Figure 2-24 MSM80C154S/MSM83C154S program area

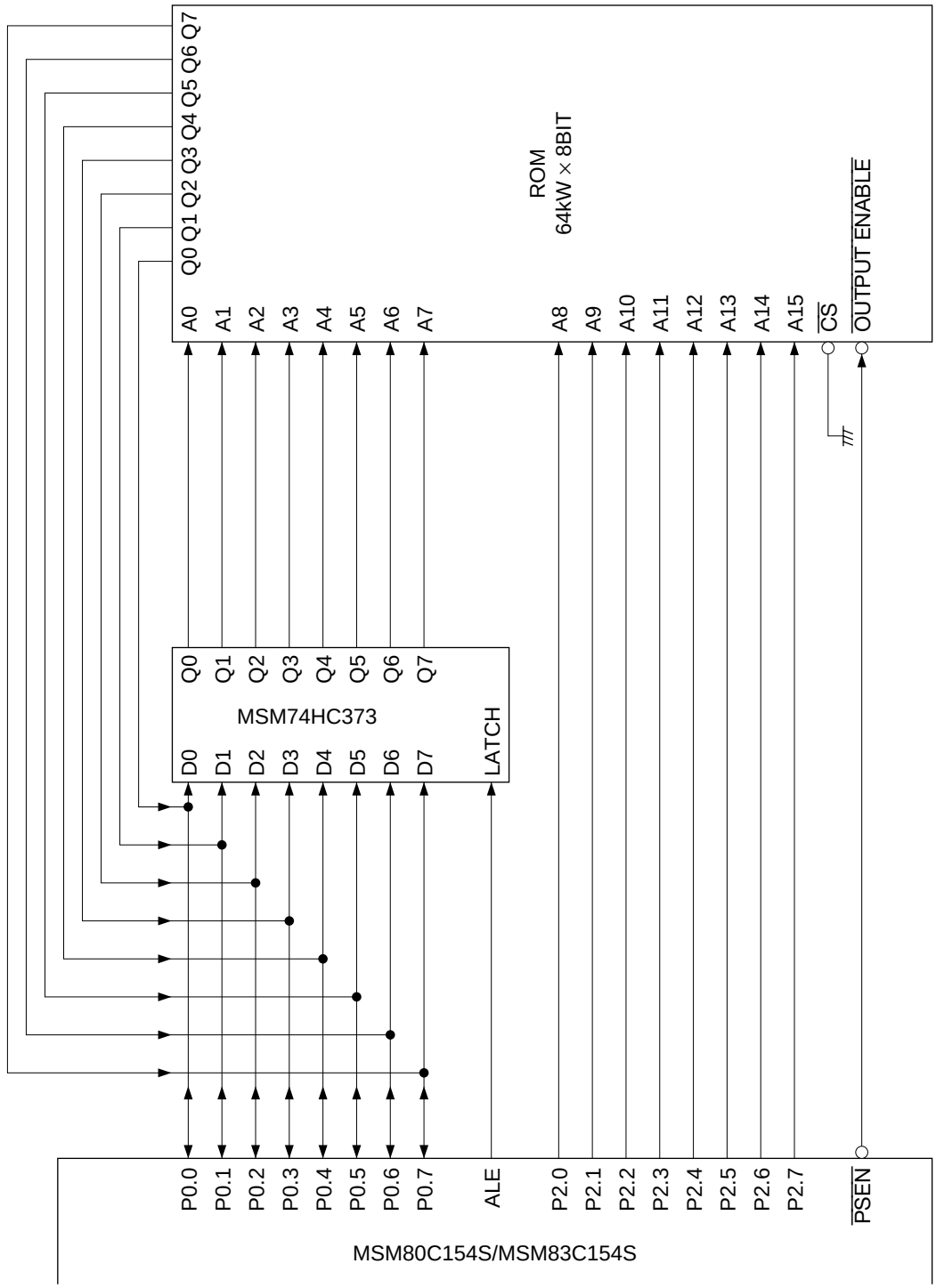


Figure 2-25 MSM80C154S/MSM83C154S external ROM connection diagram

2.10.2 Procedures and circuit connections used when external data memory (RAM) is accessed by data pointer (DPTR)

The MSM80C154S/MSM83C154S can be connected to an external 64K word × 8-bit data memory (RAM) when accessing the memory by data pointer (DPTR).
The data pointer (DPTR) consists of DPL and DPH registers. The DPL register contents serve as addresses 0 thru 7 of the external data memory, and the DPH register contents serve as addresses 8 thru 15.
The MOVX @DPTR, A instruction is used when accumulator contents are transferred to an external data memory, and the MOVX A, @DPTR instruction is used when external data memory contents are transferred to the accumulator. The external data memory connection diagram is shown in Figure 2-26 and the external data memory access time chart is shown in Figure 2-27.
When the data pointer indirect external memory instruction is executed, the CPU passes the DPL register contents to port 0, and the port 0 contents are latched externally by ALE signal. Data stored in the latch serves as the lower order addresses 0 thru 7 of the external data memory (RAM), and the DPH register contents passed to port 2 serve as the higher order addresses 8 thru 15 for addressing of the external data memory.
The $\overline{WR}$ or $\overline{RD}$ external data memory control signal is subsequently generated by the CPU to enable transfer of data between port 0 and the external data memory.

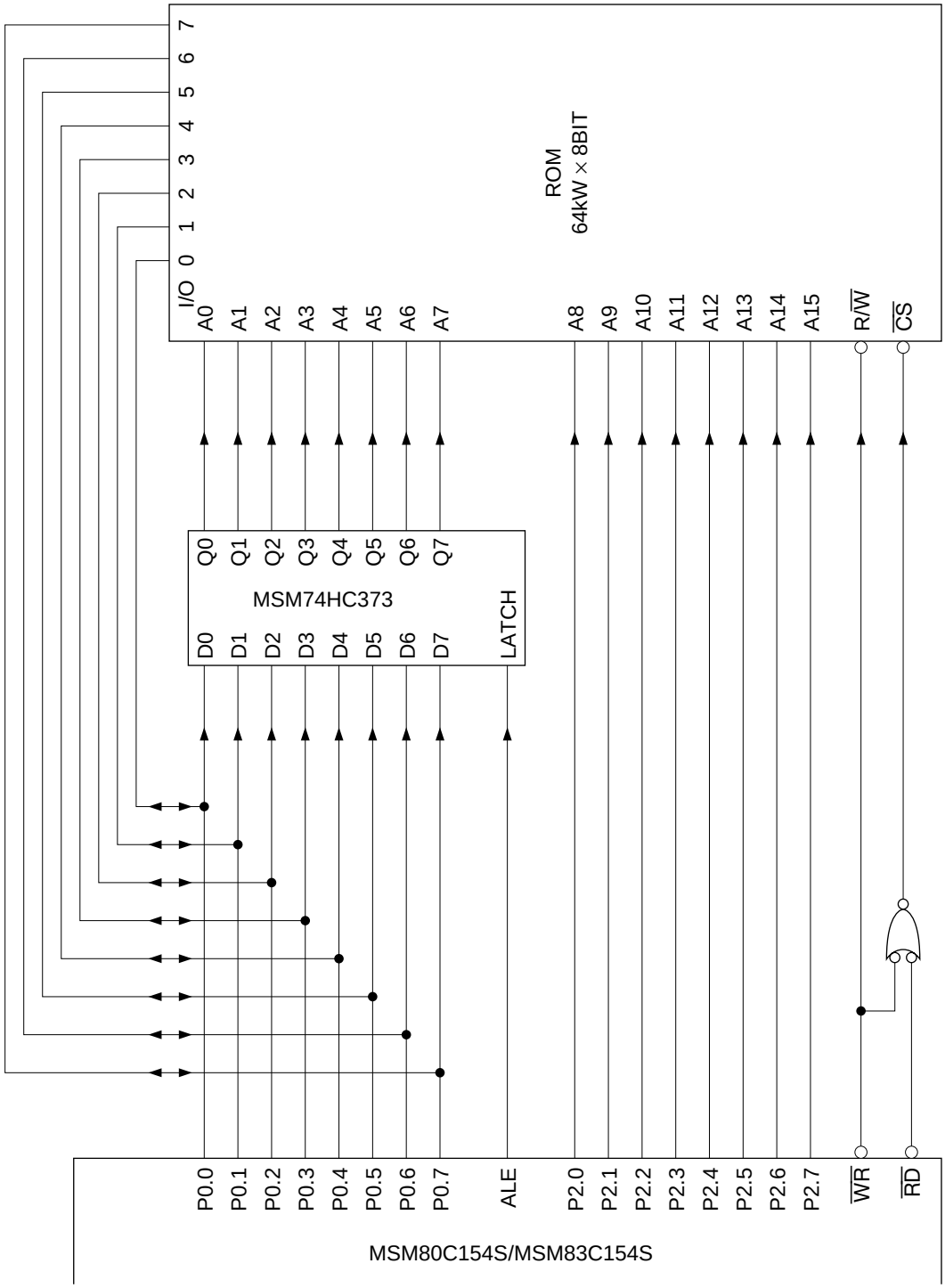


Figure 2-26 Connection circuit for external data memory addressed by DPTR

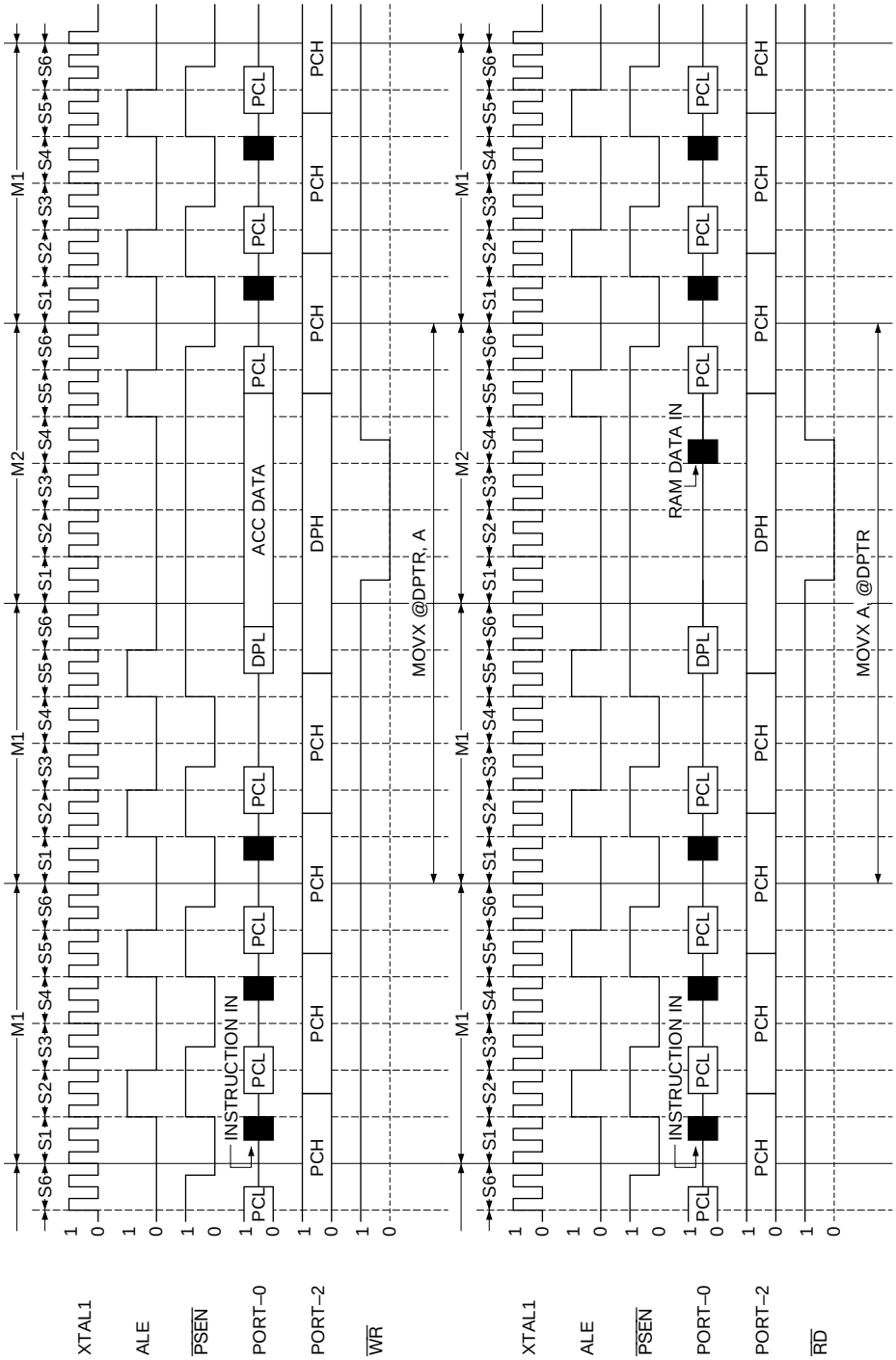


Figure 2-27 DPTR external data memory access timing

2.10.3 Procedures and circuit connections used when external data memory (RAM) is accessed by registers R0 and R1

The MSM80C154S/MSM83C154S can be connected to an external 256 word × 8-bit data memory (RAM) when addressing the memory according to the contents of registers R0 and R1 in the internal data memory (RAM).
The MOVX @Rr, A instruction is used when accumulator contents are transferred to an external data memory, and the MOVX A, @Rr instruction is used when external data memory contents are transferred to the accumulator. The external data memory connection diagram is shown in Figure 2-28 and the external data memory access time chart is shown in Figure 2-29.
When the indirect register external memory instruction is executed, the CPU passes the R0 or R1 register contents to port 0, and the port 0 contents are latched externally by the ALE signal. Data stored in the latch serves as the addresses 0 thru 7 of the external data memory. The $\overline{WR}$ or $\overline{RD}$ external data memory control signal is subsequently generated by the CPU to enable transfer of data between port 0 and the external data memory.
However, if the port 2 latched data is used in addresses 8 thru 15 of the external data memory, the circuit connections are the same as when the data pointer (DPTR) is used, thereby enabling a 64K byte × 8-bit data memory to be accessed.

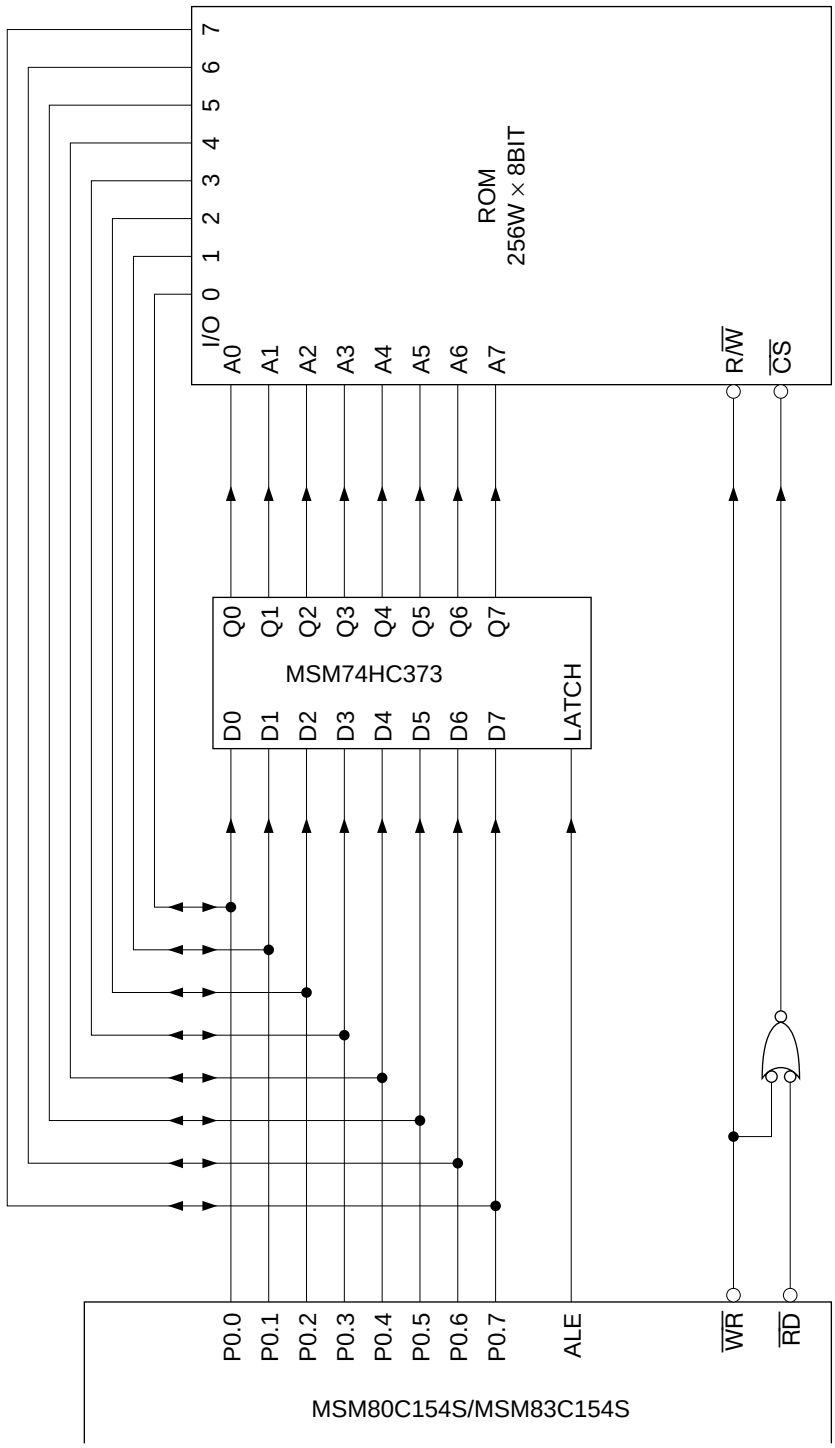


Figure 2-28 Connection circuit for external data memory addressed by register R0 or R1

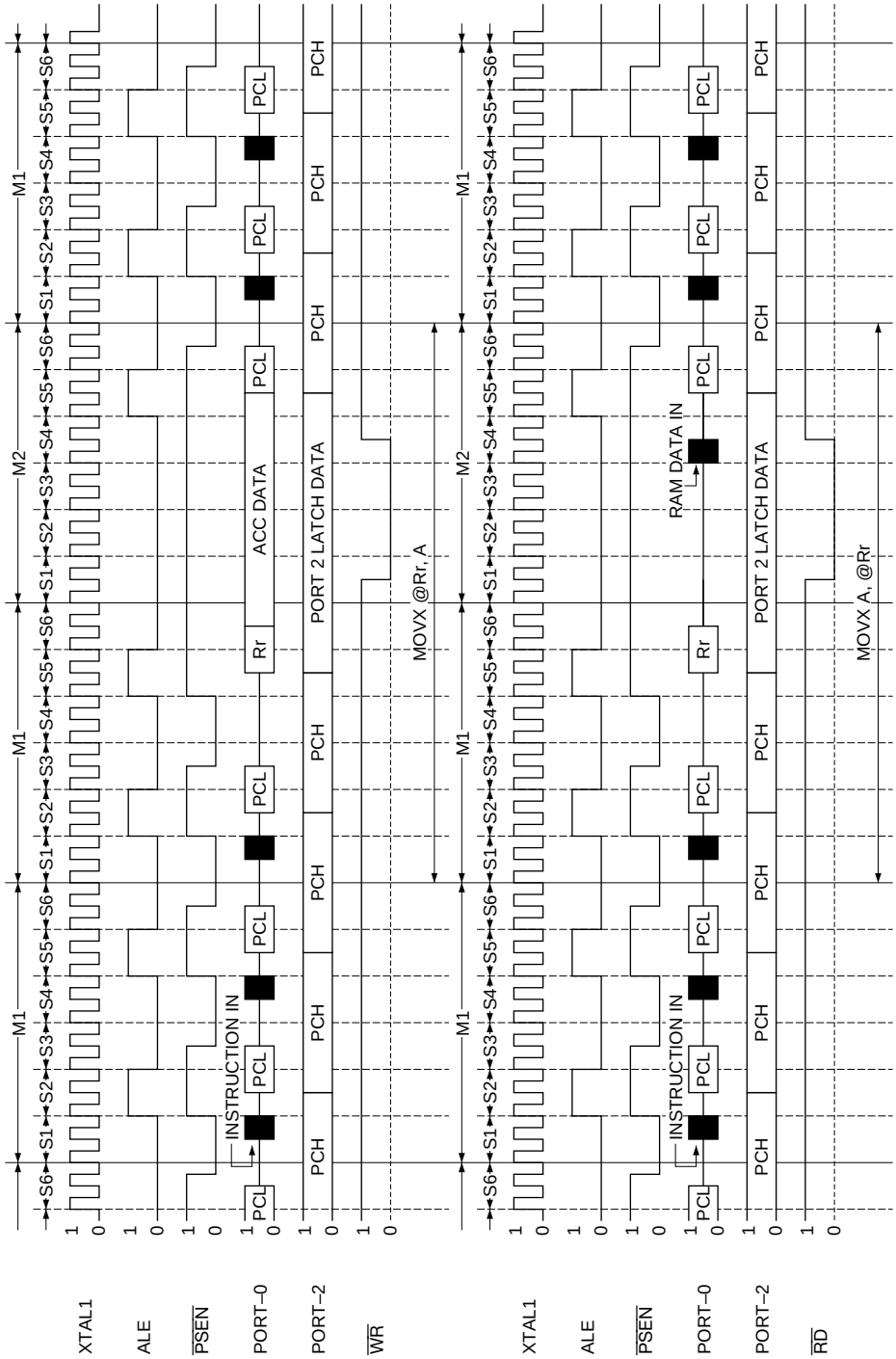


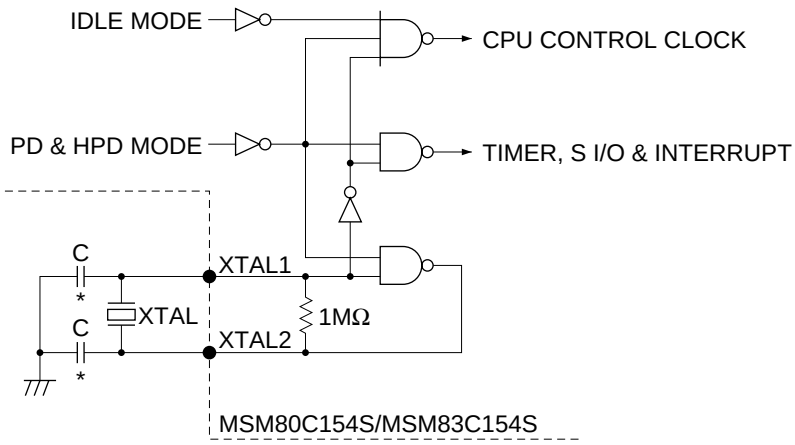
Figure 2-29 Register R0/R1 external data memory access timing

3. CONTROL

3. CONTROL

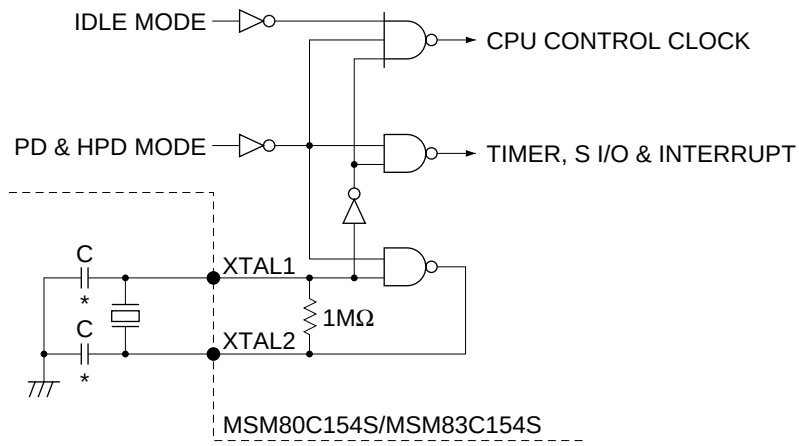
3.1 Oscillators: XTAL1
XTAL2

An oscillator is formed by connecting a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the MSM80C154S/MSM83C154S devices.
If an external clock is applied to XTAL1, the input should be at 50% duty and C-MOS level.



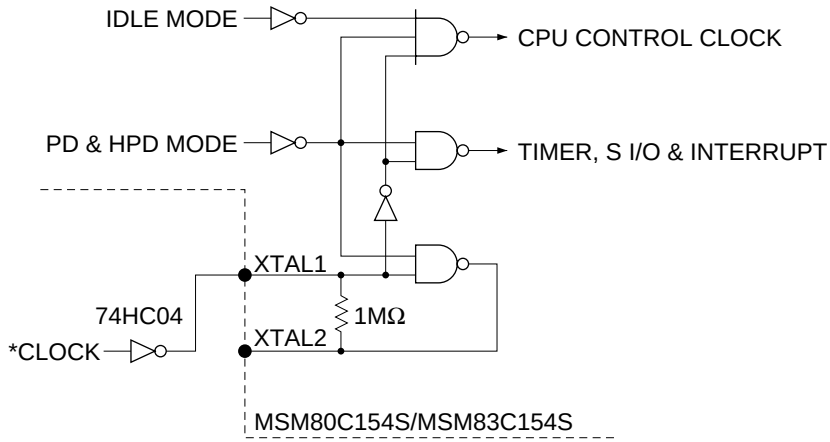
* The capacity of the compensating capacitor depends on the crystal resonator.
* The XTAL1·2 frequency depends on V_{CC}.

Figure 3-1 Crystal resonator connection diagram



* The capacity of the compensating capacitor depends on the ceramic resonator.
* The XTAL1·2 frequency depends on VCC.

Figure 3-2 Ceramic resonator connection diagram



* Supply of 50% duty clock

Figure 3-3 External clock supply circuit

3.2 CPU Resetting

3.2.1 Outline

If a reset signal (kept at “1” level for at least 1µsec) is applied to the RESET pin when the correct voltage (in respect to the various specifications) is applied to the MSM80C154S/ MSM83C154S VCC pin, a reset signal is stored in the CPU even if the XTAL1·2 oscillators have been stopped.
The internally stored reset signal is used in direct initialization (setting to “1”) of ports 0, 1, 2, and 3. All of the special function registers are then initialized (set to “0”) two machine cycles after the XTAL1·2 oscillator commences regular operation.
When the reset is released, instruction execution is started in the third machine cycle if the reset signal is changed from “1” level to “0” level before the M1·S1 signal leading edge, and in the fifth machine cycle if the reset signal is changed from “1” to “0” after the leading edge.
The reset circuit block diagram is shown in Figure 3-4, the reset start time charts in Figures 3-5 and 3-6, and the reset release time charts in Figures 3-7 and 3-8.

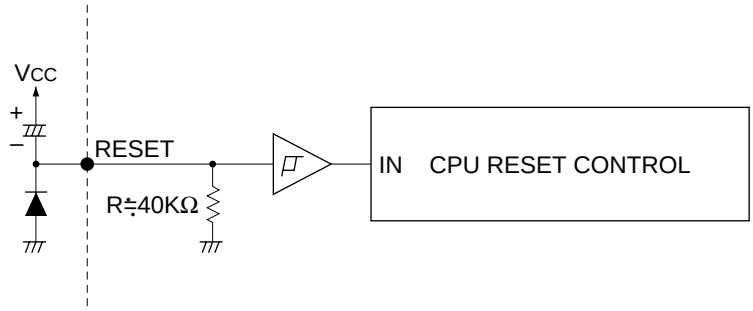


Figure 3-4 MSM80C154S/MSM83C154S reset circuit block diagram

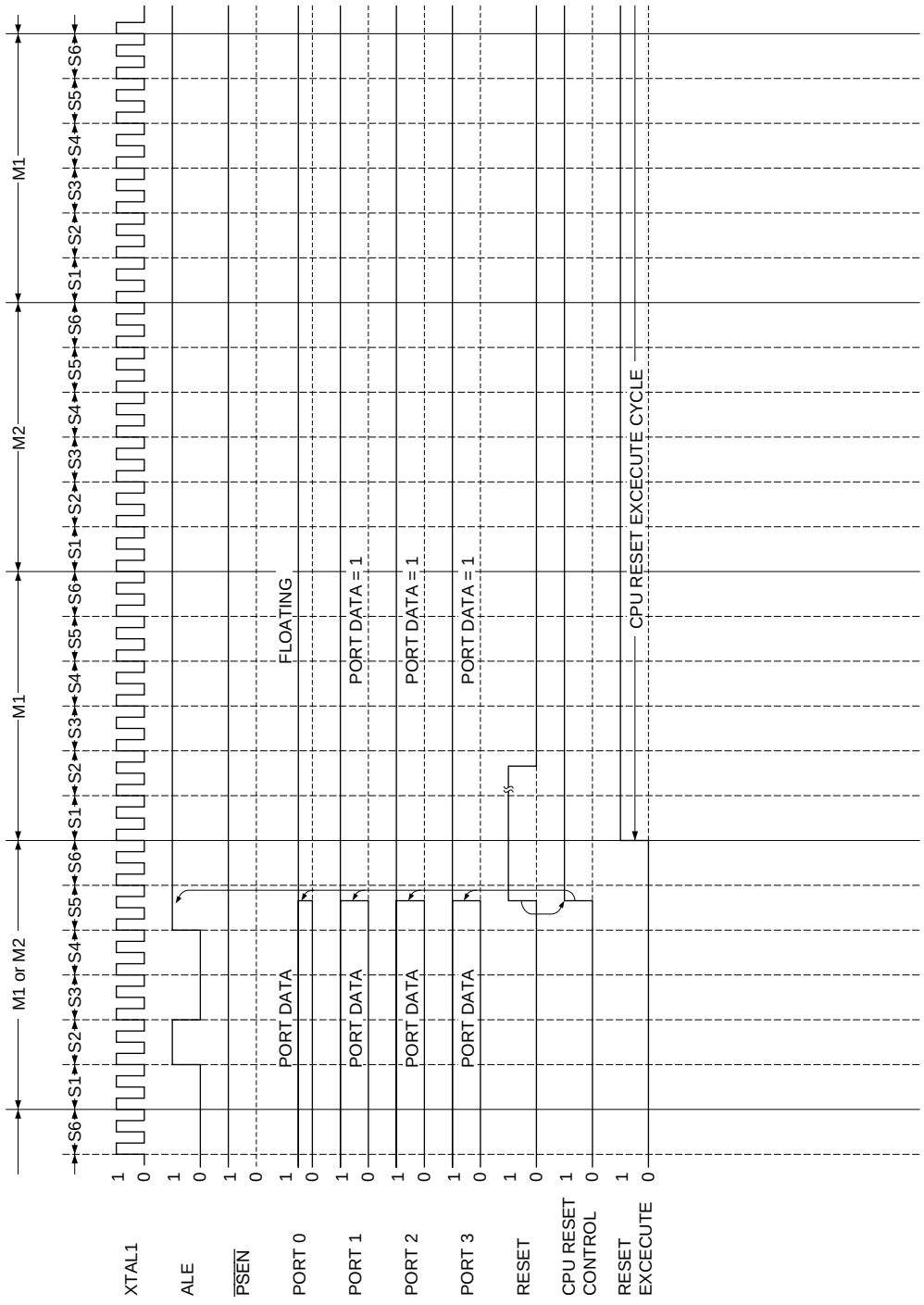


Figure 3-5 Reset execution time chart (internal ROM mode)

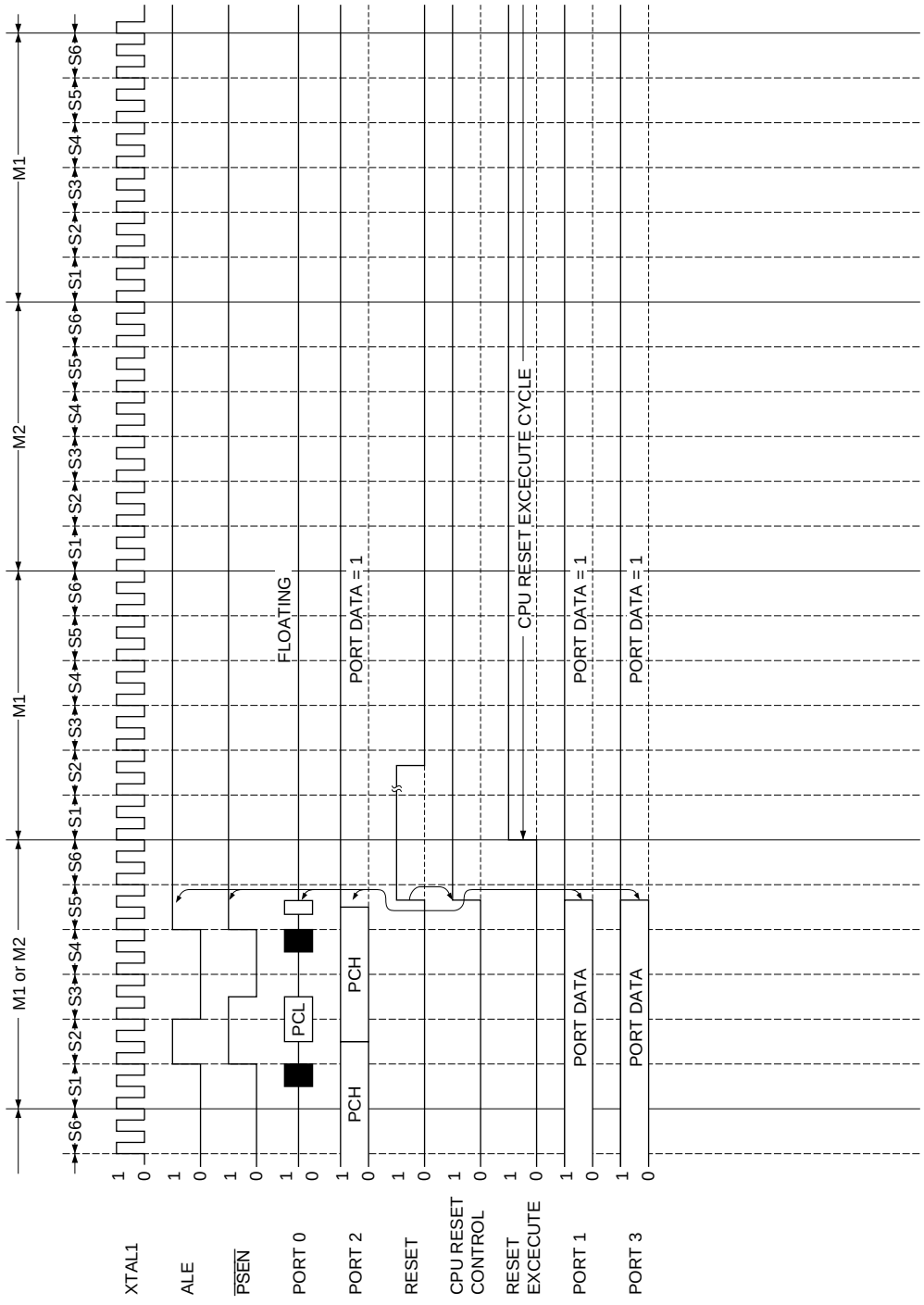


Figure 3-6 Reset execution time chart (external ROM mode)

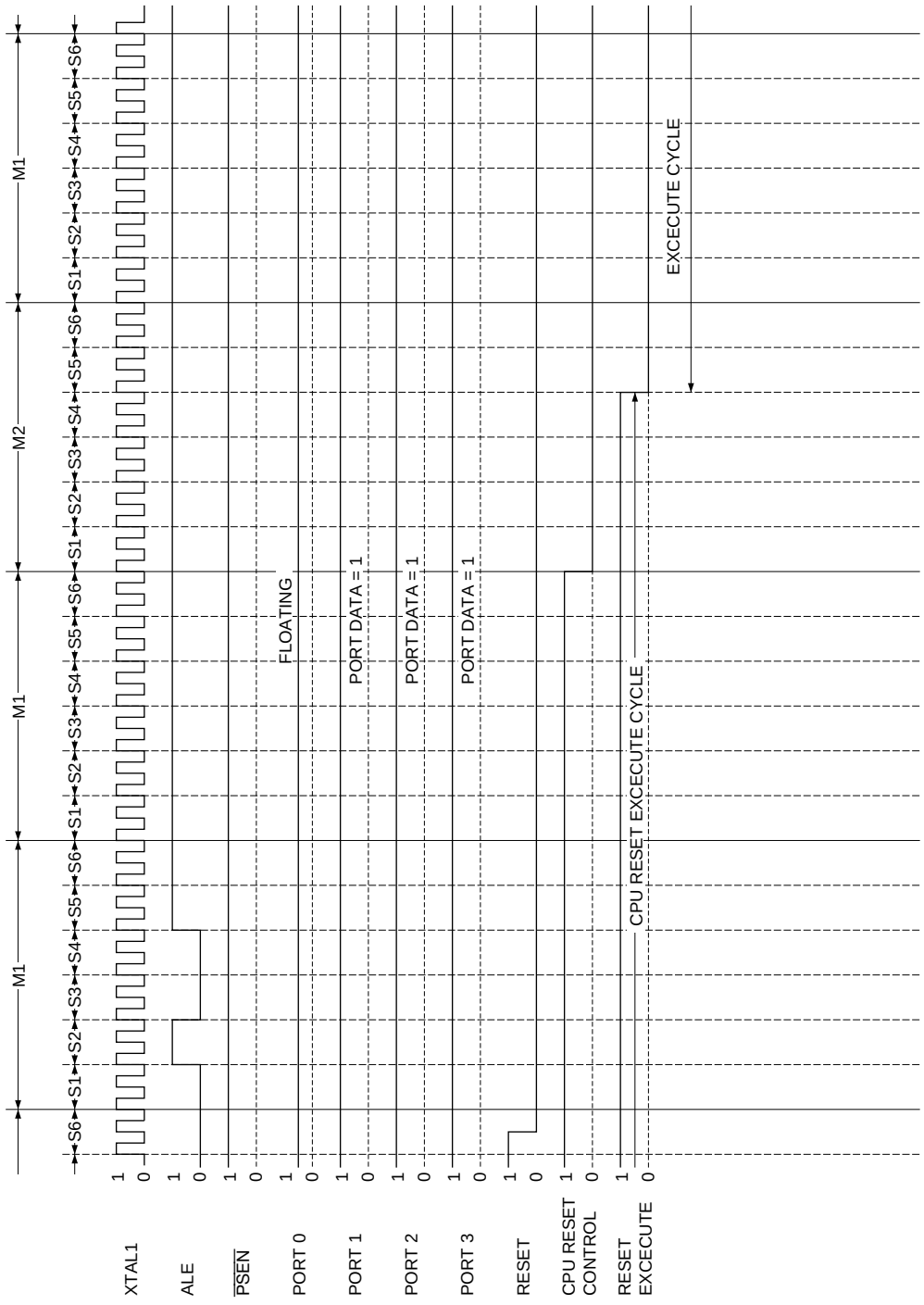


Figure 3-7 Reset release time chart (internal ROM mode)

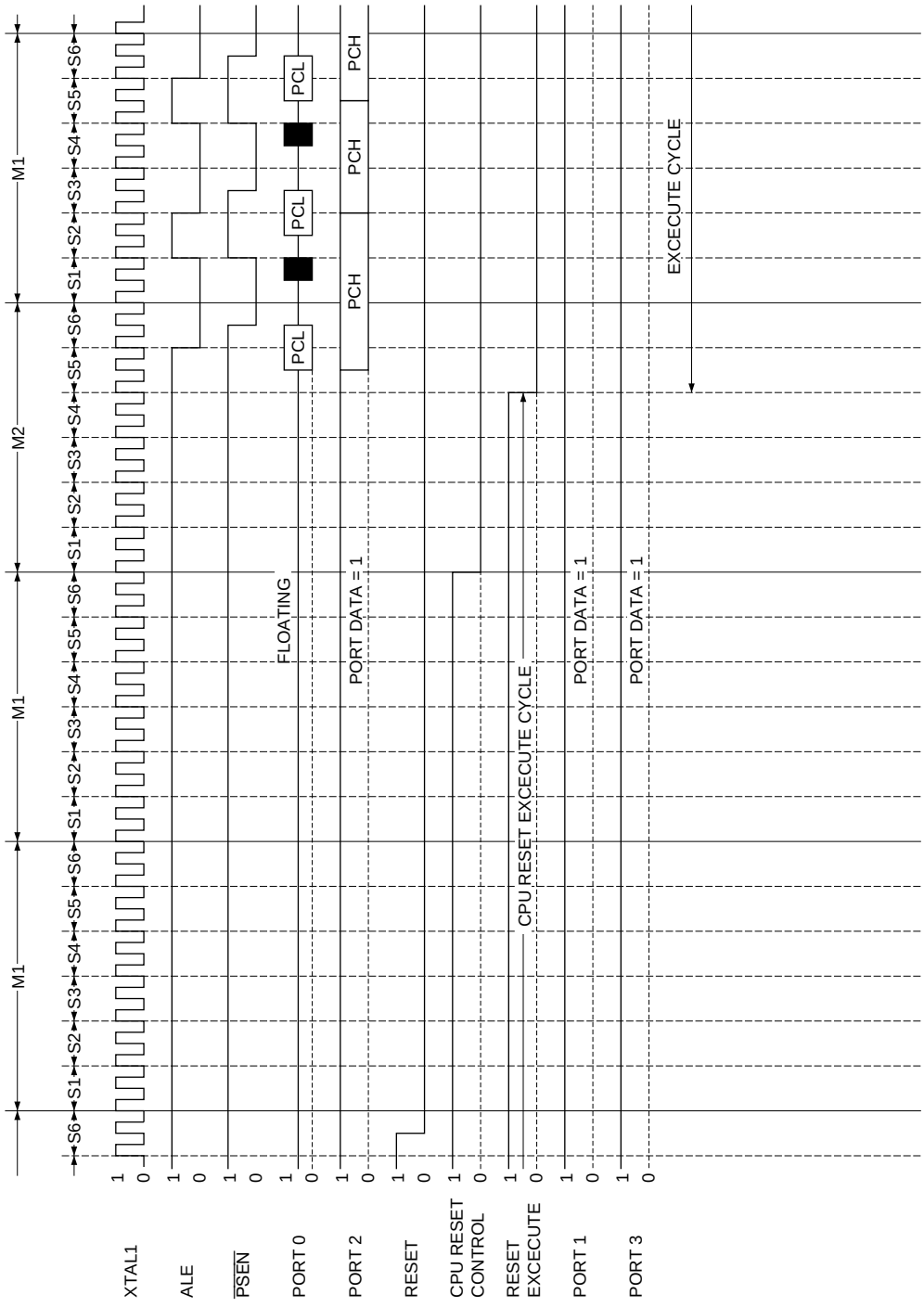


Figure 3-8 Reset release time chart (external ROM mode)

3.2.2 Reset Schmitt trigger circuit

The Schmitt trigger circuit connected to the RESET pin shown in the MSM80C154S/ MSM-83C154S reset circuit block diagram in Figure 3-4 operates in the following way when the V_{CC} power supply voltage is +5V.
If the voltage of the reset signal applied to the RESET pin exceeds 3V when the level of that signal is changed from “0” to “1”, the Schmitt trigger output level is changed from “0” to “1”, and the reset signal is set in the CPU reset control circuit, resulting in the reset operation being started by the CPU.
The CPU reset state is released when the “1” level on the RESET pin is changed to “0”. An input signal level below 1.5V is regarded as “0” level, and the Schmitt trigger output level is changed from “1” to “0”. When the reset signal is changed to “0” level, the CPU reset control circuit is ready for reset release. The Schmitt trigger circuit operation time chart for changes in the reset input voltage is outlined in Figure 3-9.

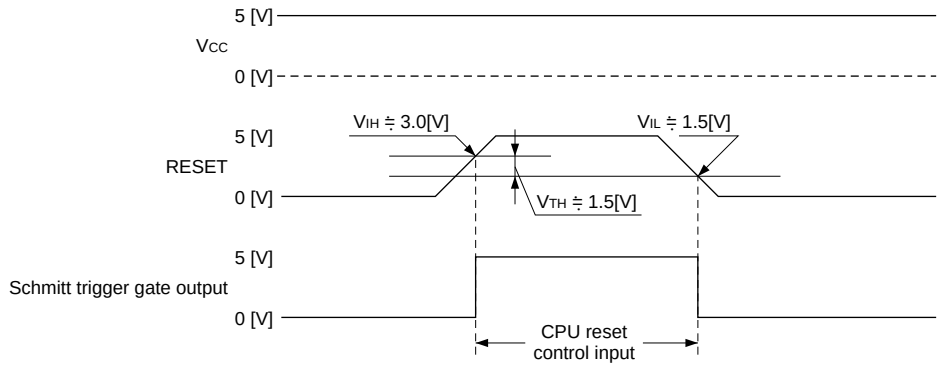


Figure 3-9 Reset Schmitt trigger gate detector time chart

3.2.3 CPU internal status by reset

When a reset signal is applied to the CPU with normal voltage applied to the MSM80C154S/MSM83C154S V_{CC} power supply pin, ports 0, 1, 2, and 3 are set to "1" (input mode) even if XTAL1·2 oscillation has been stopped. The output status of the ALE and $\overline{PSEN}$ pins also becomes "1". The CPU is then reset after normal XTAL1·2 oscillation has resumed. The internal CPU status when the CPU is reset is shown in Table 3-1.

Table 3-1 MSM80C154S/MSM83C154S reset internal status

Register Name	Register Reset Status
PC	0000H
SP	07H
IP	40H(0 × 000000)
IE	40H(0 × 000000)
PCON	10H(000 × 0000)
PSW, DPH, DPL, A, B	00H
SCON, TCON, TMOD	
T2CON, IOCON, TL0	
TL1, TL2, TH0, TH1	
TH2, RCAP2L, RCAP2H	
P1, P2, P3	*0FFH(input port)
P0	*0FFH(floating)
SBUF	Undefined
INTERNAL RAM	
ALE, $\overline{PSEN}$	*"1" OUT

* Denotes direct resetting even if XTAL1·2 has stopped.

3.3 $\overline{EA}$ (CPU Memory Separate)

3.3.1 Outline

The function of the $\overline{EA}$ pin is to determine whether a CPU internal program memory (ROM) instruction or an external program instruction is to be executed.

(1) Internal ROM mode
If the $\overline{EA}$ pin is connected to V_{CC} and a "1" reset signal is applied to the RESET pin to reset the CPU, an internal program memory (ROM) is executed from address 0. (MSM83C154S, MSM85C154HVS)

(2) External ROM mode
If the $\overline{EA}$ pin is connected to V_{SS} and a "1" reset signal is applied to the RESET pin to reset the CPU, an external program memory is executed from address 0.

4. INTERNAL SPECIFICATIONS

4. INTERNAL SPECIFICATIONS

4.1 Internal Data Memory (RAM) and Special Function Registers

4.1.1 Outline

MSM80C154S/MSM83C154S operation is based on an instruction code address method where operations are specified in an instruction code (OP) section, and the data memory (RAM) and special function registers (ACC, B, TCON, P0.....) are specified directly by part of the instruction code and the second or third byte of data following that instruction code.
According to this instruction code address method, all eight bits of data in the data memory and special function register may be specified, or one bit of data memory and one bit of data in the special function register may be specified. Direct designation of all eight bits of data is called data addressing, and direct designation of one bit of data is called bit addressing.
Since these CPU devices specify data memory (RAM) and special function register contents by the above method, specific addresses are assigned to the respective CPU data memory (RAM) and special function registers (ACC, B, TCON, P0, ...). Data addresses consist of eight bits, and range from 00 to 0FFH in binary (which correspond to 0 thru 255 in decimal). All data memory (RAM) and special function registers (ACC, B, TCON, P0, ...) exist in these 256 locations.
The data memory contains 256 bytes. The data memory between addresses 00 thru 7FH can be specified directly by data address, and the data memory from address 80H to 0FFH can be specified by indirect register instruction where R0 or R1 contents are set to 80H thru 0FFH. Note that the entire data memory (RAM) from 00 thru 0FFH can be specified by indirect register instruction.
Special function registers are located between addresses 80H thru 0FFH, and can also be specified directly by data address. Bit addresses consist of eight bits, the manipulation bits being specified by the three lower order bits and the data memory (RAM) or special function register (ACC, B, TCON, P0, ...) by the five higher order bits. Data memory between addresses 20 thru 2FH can be specified by bit addressing. Other areas cannot be specified by bit designation.
The special function registers which can be specified by bit address are P0, P1, P2, P3, TCON, SCON, IE, IP, T2CON, PSW, ACC, B, and IOCON, a total of 13 registers. The data memory (RAM) and special function register address space layout is shown in Figure 4-1.

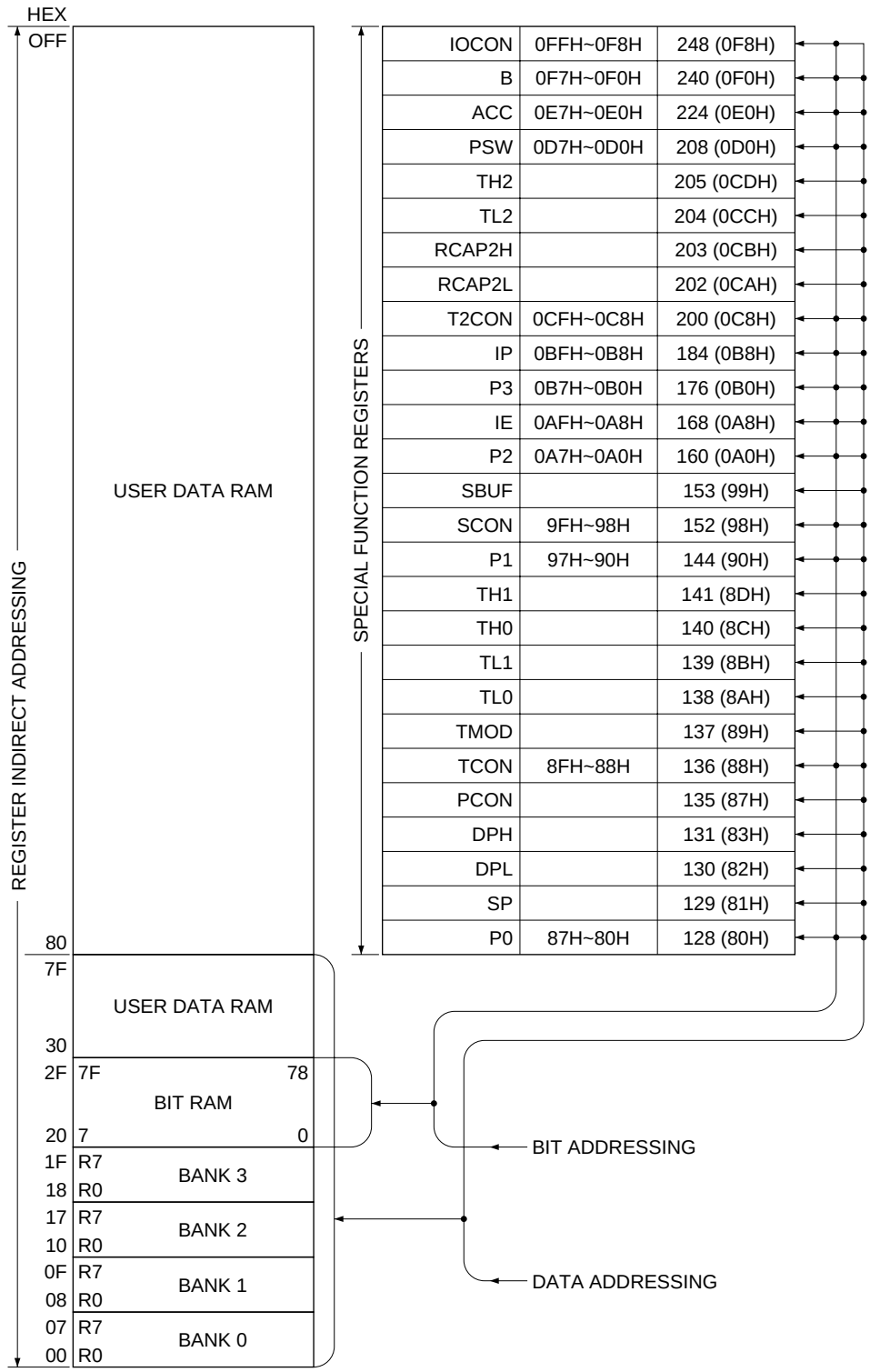


Figure 4-1 Data memory and special function register layout

4.2 Internal Data Memory (RAM)

4.2.1 Internal data memory (RAM)

The storage capacity of the MSM80C154S/MSM83C154S data memory is 256 words × 8 bits. The layout diagram is shown in Figure 4-2.
The data memory can be accessed (R/W) in four different ways - direct register designation, indirect register designation, data addressing, and bit addressing.
Four banks of registers group (R0 thru R7 × 4) exist within the data memory address range from 00 to 1FH. Banks are specified by RS0 and RS1 data combinations within the PSW.
The data memory address range from 20 to 2FH is an area where bit addressing is possible. One bit of data can be manipulated directly by bit manipulation instructions.
The data memory address range from 00 to 7FH is an area where data addressing is possible. 8-bit data manipulations can be handled directly by data address manipulation instructions.
The data memory address range from 80H to 0FFH is an area where data addressing is not possible. To manipulate data in this data memory area, the contents of register R0 or R1 are set in 80H thru 0FFH, then an indirect register instruction is used. (Indirect register instructions can be used to specify the entire data memory from address 00 to 0FFH.)
In addition to data storage in the CPU, the data memory is used as the place for saving stack data. This stack data storage area is addressed by a stack pointer (SP 81H).
Since the stack pointer can be set any desired value by software, the data memory can be used as stack from any data memory address. Note that 07H data is set automatically in the stack pointer when the CPU is reset.

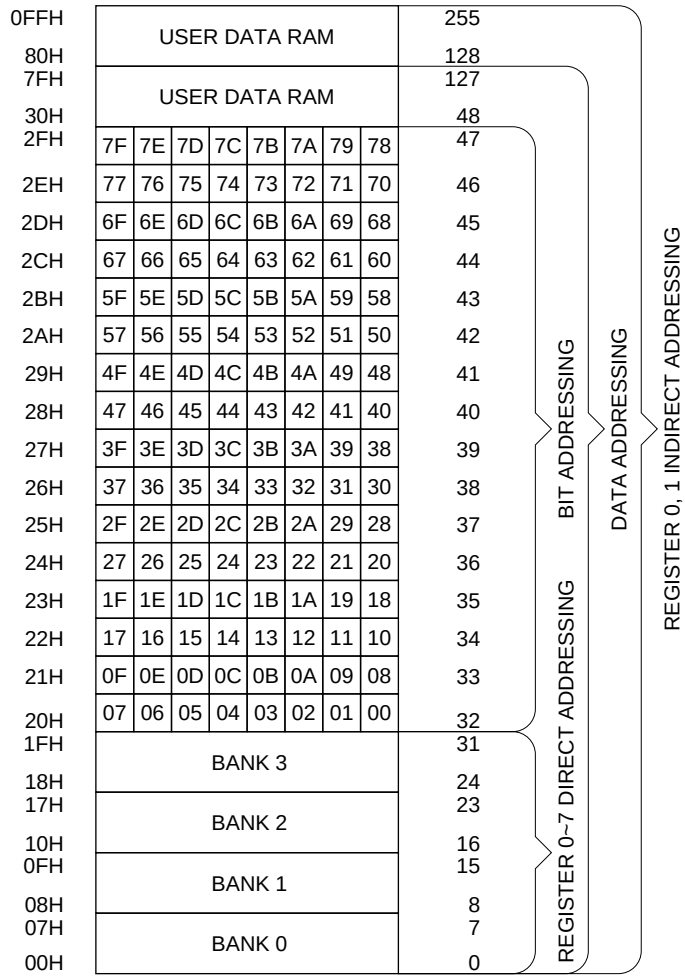


Figure 4-2 RAM layout diagram

4.2.2 Internal data memory registers R0 thru R7

Four banks of registers group exist in the data memory (RAM) between memory addresses 00 thru 1FH. Banks are specified by RS0 and RS1 bit combinations within the program status word (PSW). Note that the register area R0 thru R7 can also be used as normal data memory. The PSW table is shown in Table 4-1, and the data memory register bank layout in Figure 4-3.

Table 4-1 Program status word (PSW)

Bit	7	6	5	4	3	2	1	0
Flag	CY	AC	F0	RS1	RS0	OV	F1	P
Set				•	•			

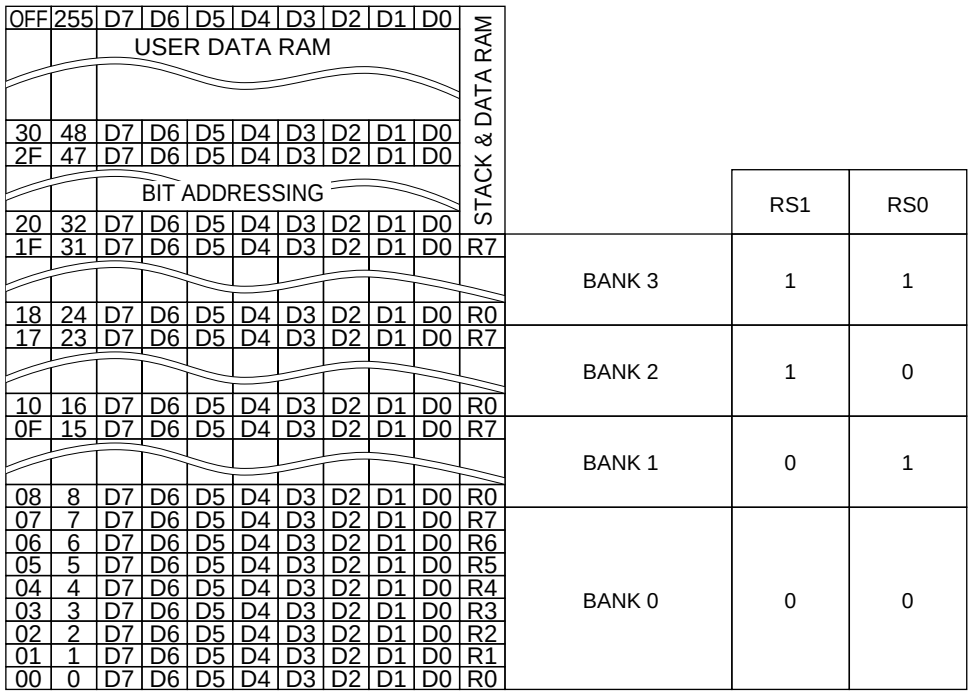


Figure 4-3 Internal data memory register bank layout

4.2.3 Stack

The stack data save (storage) area is in the internal data memory (RAM), and is specified by stack pointer (SP 81H).
Although 07H data is automatically set in the stack pointer when the CPU is reset, any desired data can be set by software to enable the data memory to be used as stack from any address. Two bytes of data memory are used when the stack is used by interrupt or CALL instruction, and a single byte of data memory is used when the PUSH instruction is used. The status where an interrupt is generated and the program counter contents are saved in the stack when the stack pointer contents are 7FH, and the status where accumulator contents are pushed during interrupt routine and are subsequently saved in the stack are shown in Table 4-2. The stack status up to completion of interrupt processing upon execution of POP and RETI instructions is also included.

Table 4-2 Stack storage layout

Stack processing	Stack pointer	RAM data bit							
		7	6	5	4	3	2	1	0
Before execution	7FH	D7	D6	D5	D4	D3	D2	D1	D0
Interrupt process	80H	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
(push PC)	81H	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8
PUSH process (ACC)	82H	A7	A6	A5	A4	A3	A2	A1	A0
POP process (ACC)	82H	A7	A6	A5	A4	A3	A2	A1	A0
RETI process (pop PC)	81H	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8
	80H	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
After execution	7FH	D7	D6	D5	D4	D3	D2	D1	D0

4.3 Internal Data Memory (RAM) Operating Procedures

4.3.1 Internal data memory indirect addressing

Operation of the internal data memory indirect increment instruction is described here as an example. This instruction (INC @Rr) is a 1-byte 1-machine cycle instruction (see Figure 4-4). The indirect address register is specified by instruction code bit 0 data r where r denotes either register 0 or 1 in the register group specified by PSW RS0 and RS1 bank data. Register 0 is specified when the r data is 0, and register 1 is specified when the data is 1.
When this instruction is executed, register data is read from the specified register 0 or 1, and the read out register data is written into the data pointer for the data memory.
The data memory contents specified by the data pointer are read by the CPU into a temporary register. Then a subsequent increment (+1) by the ALU is followed by a return to the data memory at the address where the data were read out. In this way, the contents of the data memory at the address specified by the contents of R0 or R1 are incremented.

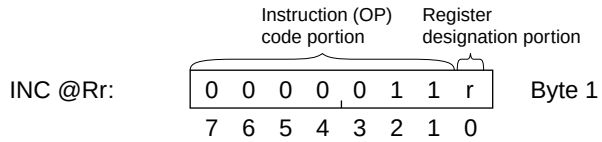


Figure 4-4 INC @Rr bit arrangement

4.3.2 Internal data memory register R0 thru R7 designation

Operation of the internal data memory register decrement instruction is described here as an example. This instruction (DEC Rr) is a 1-byte 1-machine cycle instruction (see Figure 4-5). Register R0 thru R7 is specified by r_0, r_1, and r_2 data of instruction code bit 0, 1, and 2. The r_0, r_1, and r_2 data is represented in binary code, r_0 being the LSB, and r_2 the MSB. The code is weighted 1, 2, and 4 from the LSB. Any one of the eight registers can be specified by combinations of this code. See Table 4-3 for the register designation combinations.
When this instruction is executed, one of the registers R0 thru R7 from the register group specified by the PSW RS0 and RS1 bank data is specified. The contents of the specified register is read by the CPU into a temporary register. Then a subsequent decrement (–1) by the ALU is followed by a return to the register where the data were read out. In this way, the register contents specified by r_0, r_1, and r_2 are decremented.

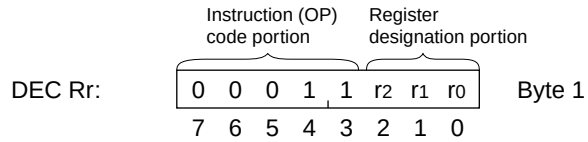


Figure 4-5 DEC Rr bit arrangement

Table 4-3 Register designation table

Register name	r_2	r_1	r_0
Register 0	0	0	0
Register 1	0	0	1
Register 2	0	1	0
Register 3	0	1	1
Register 4	1	0	0
Register 5	1	0	1
Register 6	1	1	0
Register 7	1	1	1

4.3.3 Internal data memory 1-bit data designation

In the MSM80C154S/MSM83C154S, 1-bit data manipulations (test, reset, set, complement, transfer) can be executed directly between internal data memory addresses 20 thru 2FH by bit manipulation instructions. The operation of a bit reset instruction is described below as an example.

This instruction (CLR bit address) is a 2-byte 2-machine cycle instruction (see Figure 4-6). The instruction code is indicated in byte 1, and the data memory address and bit designation are indicated in byte 2. The manipulation bit is specified by the b_0, b_1, and b_2 data in bits 0, 1, and 2 of byte 2. The b_0, b_1, and b_2 portion is expressed in binary code which is weighted 1, 2, and 4. Combinations of this code enable any one of eight bits to be specified. The bit designation combinations are listed in able 4-4.

The data memory is addressed by bits b_3, b_4, b_5, b_6 and b_7 of byte 2 with b_7 being "0". These bits can be expressed in binary by 0 thru 0FH, and a total of 16 designations of the data memory are possible.

When data memory addresses are specified, the data memory bit manipulation start address 20H is added to the b_3, b_4, b_5, and b_6 binary data to obtain the data memory address.

The data memory contents specified by the above method are read by the CPU into a temporary register, the specified bit data is reset to "0" by the ALU, and the CPU returns the result to the data memory where the data were read. One bit of specified data memory is thus reset to "0".

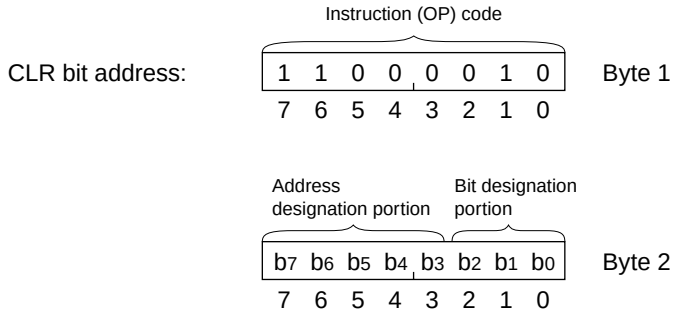


Figure 4-6 CLR bit address bit arrangement

Table 4-4 Bit designation table

Bit name	b_2	b_1	b_0
Bit 0	0	0	0
Bit 1	0	0	1
Bit 2	0	1	0
Bit 3	0	1	1
Bit 4	1	0	0
Bit 5	1	0	1
Bit 6	1	1	0
Bit 7	1	1	1

Table 4-5 Addressing combination table

	b_7	b_6	b_5	b_4	b_3	RAM address	
0	0	0	0	0	0	20H	32
1	0	0	0	0	1	21H	33
2	0	0	0	1	0	22H	34
3	0	0	0	1	1	23H	35
4	0	0	1	0	0	24H	36
5	0	0	1	0	1	25H	37
6	0	0	1	1	0	26H	38
7	0	0	1	1	1	27H	39
8	0	1	0	0	0	28H	40
9	0	1	0	0	1	29H	41
A	0	1	0	1	0	2AH	42
B	0	1	0	1	1	2BH	43
C	0	1	1	0	0	2CH	44
D	0	1	1	0	1	2DH	45
E	0	1	1	1	0	2EH	46
F	0	1	1	1	1	2FH	47

4.4 Special Function Registers (TCON, SCON,.... ACC, B)

4.4.1 Outline

As can be seen from the configuration shown in Table 4-6, the MSM80C154S/ MSM83C154S special function registers consist of 27 8-bit registers.
Special function registers can be accessed (R/W) by either data addressing or bit addressing. All 27 registers can be specified by data addressing. 13 registers (P0, P1, P2, P3, TCON, T2CON, SCON, IE, IP, PSW, ACC, B, and IOCON) can be specified by bit addressing.
If a register which does not exist at the data address is accessed when a special function register is used, the read data becomes 0FFH. And when data is written, none of the registers in the CPU are effected at all. Note, however, that since a jump is always executed when a bit test instruction which results in a relative jump at data condition "1" is executed, make sure that no instruction is executed for a register which does not exist.

Table 4-6 List of special function registers

Register name	Bit address								Data address
	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
IOCON	FF	FE	FD	FC	FB	FA	F9	F8	0F8H(248)
B	F7	F6	F5	F4	F3	F2	F1	F0	0F0H(240)
ACC	E7	E6	E5	E4	E3	E2	E1	E0	0E0H(224)
PSW	D7	D6	D5	D4	D3	D2	D1	D0	0D0H(208)
TH2									0CDH(205)
TL2									0CCH(204)
RCAP2H									0CBH(203)
RCAP2L									0CAH(202)
T2CON	CF	CE	CD	CC	CB	CA	C9	C8	0C8H(200)
IP	BF	BE	BD	BC	BB	BA	B9	B8	0B8H(184)
P3	B7	B6	B5	B4	B3	B2	B1	B0	0B0H(176)
IE	AF	AE	AD	AC	AB	AA	A9	A8	0A8H(168)
P2	A7	A6	A5	A4	A3	A2	A1	A0	0A0H(160)
SBUF									99H(153)
SCON	9F	9E	9D	9C	9B	9A	99	98	98H(152)
P1	97	96	95	94	93	92	91	90	90H(144)
TH1									8DH(141)
TH0									8CH(140)
TL1									8BH(139)
TL0									8AH(138)
TMOD									89H(137)
TCON	8F	8E	8D	8C	8B	8A	89	88	88H(136)
PCON									87H(135)
DPH									83H(131)
DPL									82H(130)
SP									81H(129)
P0	87	86	85	84	83	82	81	80	80H(128)

4.4.2 Special function registers

4.4.2.1 Timer mode register (TMOD)

Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
TMOD	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

Bit location	Flag	Function
TMOD.0	M0	M1 M0: Timer/counter 0 mode setting 0 0: 8-bit timer/counter with 5-bit prescalar 0 1: 16-bit timer/counter
TMOD.1	M1	1 0: 8-bit timer/counter with 8-bit auto reloading 1 1: Timer/counter 0 separated into TL0 (8-bit) timer/counter and TH0 (8-bit) timer/counter. TF0 is set by TL0 carry, and TF1 is set by TH0 carry.
TMOD.2	C/$\overline{T}$	Timer/counter 0 count clock designation control bit. XTAL1·2 divided by 12 clock is the input applied to timer/counter 0 when C/$\overline{T}$="0". The external clock applied to the T0 pin is the input applied to timer/counter 0 when C/$\overline{T}$="1".
TMOD.3	GATE	When this bit is "0", the TR0 bit of TCON (timer control register) is used to control the start and stop of timer/counter 0 counting. If this bit is "1", timer/counter 0 starts counting when both the TR0 bit of TCON and $\overline{INT}$0 pin input signal are "1", and stops counting when either is changed to "0".
TMOD.4	M0	M1 M0: Timer/counter 1 mode setting 0 0: 8-bit timer/counter with 5-bit prescalar 0 1: 16-bit timer/counter
TMOD.5	M1	1 0: 8-bit timer/counter with 8-bit auto reloading 1 1: Timer/counter 1 operation stopped
TMOD.6	C/$\overline{T}$	Timer/counter 1 count clock designation control bit. XTAL1·2 divided by 12 clock is the input applied to timer/counter 1 when C/$\overline{T}$="0". The external clock applied to the T1 pin is the input applied to timer/counter 1 when C/$\overline{T}$="1".
TMOD.7	GATE	When this bit is "0", the TR1 bit of TCON is used to control the start and stop of timer/counter 1 counting. If this bit is "1", timer/counter 1 starts counting when both the TR1 bit of TCON and $\overline{INT}$1 pin input signal are "1", and stops counting when either is changed to "0".

4.4.2.2 Power control register (PCON)

Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
PCON	87H	SMOD	HPD	RPD	—	GF1	GF0	PD	IDL
Bit location	Flag	Function							
PCON.0	IDL	IDLE mode set when this bit is set to "1". CPU operations are stopped when IDLE mode is set, but XTAL1·2, timer/counters 0, 1, and 2, the interrupt circuits, and serial port remain active. IDLE mode is cancelled when the CPU is reset or when an interrupt is generated.							
PCON.1	PD	PD mode set when this bit is set to "1". CPU operations and XTAL1·2 are stopped when PD mode is set. PD mode is cancelled when the CPU is reset or when an interrupt is generated.							
PCON.2	GF0	User flag. Testing this flag when IDLE mode is cancelled by an interrupt shows whether the interrupt is a normal interrupt or an IDLE mode release interrupt.							
PCON.3	GF1	User flag. Testing this flag when PD mode is cancelled by an interrupt shows whether the interrupt is a normal interrupt or a PD mode release interrupt.							
PCON.4	—	Reserved bit. The output data is "1" if the bit is read.							
PCON.5	RPD	Bit used to specify cancellation of CPU power down mode (IDLE or PD) by interrupt signal. Power down mode cannot be cancelled by interrupt signal if interrupt is not enabled by IE (interrupt enable register) when this bit is "0". If the interrupt flag is set to "1" by an interrupt request signal when this bit is "1" (even if interrupt is disabled), the program is executed from the next address of the power down mode setting instruction. The flag is reset to "0" by software.							
PCON.6	HPD	The hard power down setting mode is enabled when this bit is set to "1". If the level of the power failure detect signal applied to the HPDI pin (pin 3.5) is changed from "1" to "0" when this bit is "1", XTAL1·2 oscillation is stopped and the system is put into hard power down mode.							
PCON.7	SMOD	When the serial port is used in mode 1, 2 or 3, this bit has the following functions. The serial port operation clock is reduced by 1/2 when the bit is "0" for delayed processing. And when the bit is "1", the serial port operation clock is normal for faster processing.							

4.4.2.3 Timer control register (TCON)

Name	Address	MSB							LSB
		7	6	5	4	3	2	1	0
TCON	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Bit location	Flag	Function							
TCON.0	IT0	External interrupt 0 signal used in level detect mode when this bit is "0", and in trigger detect mode when "1".							
TCON.1	IE0	Interrupt request flag for external interrupt 0. Bit is reset automatically when interrupt is serviced. Bit can be set and reset by software when IT0="1".							
TCON.2	IT1	External interrupt 1 signal used in level detect mode when this bit is "0",and in trigger detect mode when "1".							
TCON.3	IE1	Interrupt request flag for external interrupt 1 . Bit is reset automatically when interrupt is serviced. Bit can be set and reset by software when IT1="1".							
TCON.4	TR0	Counting start and stop control bit for timer/counter 0. Timer/counter 0 starts counting when this bit is "1", and stops counting when "0".							
TCON.5	TF0	Interrupt request flag for timer interrupt 0. Bit is reset automatically when interrupt is serviced. Bit is set to "1" when carry signal is generated from timer/counter 0.							
TCON.6	TR1	Counting start and stop control bit for timer/counter 1. Timer/counter 1 starts counting when this bit is "1", and stops counting when "0".							
TCON.7	TF1	Interrupt request flag for timer interrupt 1 . Bit is reset automatically when interrupt is serviced. Bit is set to "1" when carry signal is generated from timer/counter 1.							

4.4.2.4 Serial port control register (SCON)

Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
SCON	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
Bit location	Flag	Function							
SCON.0	RI	"End of serial port reception" interrupt request flag. This flag must be reset by software during interrupt service routine. This flag is set after the eighth bit of data has been received when in mode 0, or by the STOP bit when in any other mode. In mode 2 or 3, however, RI is not set if the RB8 data is "0" with SM2="1". RI is set if STOP bit is received when SM2="1" in mode 1.							
SCON.1	TI	"End of serial port transmission" interrupt request flag. This flag must be reset by software during interrupt service routine. This flag is set after the eighth bit of data has been sent when in mode 0, or after the last bit of data has been sent when in any other mode.							
SCON.2	RB8	The ninth bit of data received in mode 2 or 3 is passed to RB8. The STOP bit is applied to R88 if SM2="0" when in mode 1. RB8 cannot be used in mode 0.							
SCON.3	TB8	The TB8 data is sent as the ninth data bit when in mode 2 or 3. Any desired data can be set in TB8 by software.							
SCON.4	REN	Reception enable control bit. No reception when REN="0". Reception enabled when REN="1".							
SCON.5	SM2	If the ninth bit of received data is "0" with SM2="1" in mode 2 or 3, the "end of reception" signal is not set in the RI flag. Nor is the "end of reception" signal set in the RI flag if the STOP bit is not "1" when SM2="1" in mode 1.							
SCON.6	SM1	SM0	SM1	MODE					
		0	0	0	8-bit shift register I/O				
		0	1	1	8-bit UART variable baud rate				
SCON.7	SM0	1	0	2	9-bit UART 1/32 XTAL1, 1/64 XTAL1 baud rate				
		1	1	3	9-bit UART variable baud rate				

4.4.2.5 Interrupt enable register (IE)

Name	Address	MSB							LSB
		7	6	5	4	3	2	1	0
IE	0A8H	EA	—	ET2	ES	ET1	EX1	ET0	EX0
Bit location	Flag	Function							
IE.0	EX0	Interrupt control bit for external interrupt 0. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".							
IE.1	ET0	Interrupt control bit for timer interrupt 0. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".							
IE.2	EX1	Interrupt control bit for external interrupt 1 . Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".							
IE.3	ET1	Interrupt control bit for timer interrupt 1 . Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".							
IE.4	ES	Interrupt control bit for serial port. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".							
IE.5	ET2	Interrupt control bit for timer interrupt 2. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".							
IE.6	—	Reserved bit. The output data is "1" if the bit is read.							
IE.7	EA	Overall interrupt control bit. All interrupts are disabled when bit is "0". All interrupts are enabled/disabled by IE.0 thru IE.5 when bit is "1".							

4.4.2.6 Interrupt priority register (IP)

Name	Address	MSB							LSB
		7	6	5	4	3	2	1	0
IP	0B8H	PCT	—	PT2	PS	PT1	PX1	PT0	PX0
Bit location	Flag	Function							
IP.0	PX0	Interrupt priority bit for external interrupt 0. Priority is assigned when bit is "1".							
IP.1	PT0	Interrupt priority bit for timer interrupt 0. Priority is assigned when bit is "1".							
IP.2	PX1	Interrupt priority bit for external interrupt 1 . Priority is assigned when bit is " 1 ".							
IP.3	PT1	Interrupt priority bit for timer interrupt 1 . Priority is assigned when bit is "1".							
IP.4	PS	Interrupt priority bit for serial port. Priority is assigned when bit is "1".							
IP.5	PT2	Interrupt priority bit for timer interrupt 2. Priority is assigned when bit is "1".							
IP.6	—	Reserved bit. The output data is "1" if the bit is read.							
IP.7	PCT	Priority interrupt circuit control bit. The priority register contents are valid and priority assigned interrupts can be processed when this bit is "0". When the bit is "1", the priority interrupt circuit is stopped, and interrupts can only be controlled by the interrupt enable register (IE).							

4.4.2.7 Program status word register (PSW)

Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
PSW	0D0H	CY	AC	F0	RS1	RS0	OV	F1	P

Bit location	Flag	Function
PSW.0	P	Accumulator (ACC) parity indicator. "1" when the "1" bit number in the accumulator is an odd number, and "0" when an even number.
PSW.1	F1	User flag which may be set to "0" or "1" as desired by the user.
PSW.2	OV	Overflow flag which is set if the carry C6 from bit 6 of the ALU or CY is "1" as a result of an arithmetic operation. The flag is also set to "1" if the resultant product of a multiplication instruction (MUL AB) is greater than 0FFH, but is reset to "0" if the product is less than or equal to 0FFH.
PSW.3	RS0	RAM register bank switch

RS1	RS0	BANK	RAM ADDRESS
0	0	0	00H – 07H
0	1	1	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

PSW.4	RS1	
PSW.5	F0	User flag which ma be set to "0" or "1" as desired by the user.
PSW.6	AC	Auxiliary carry flag. This flag is set to "1" if a carry C_3 is generated from bit 3 of the ALU as a result of executing an arithmetic operation instruction. In all other cases, the flag is reset to "0".
PSW.7	CY	Main carry flag. This flag is set to "1" if a carry C_7 is generated from bit 7 of the ALU as a result of executing an arithmetic operation instruction. In all other cases, the flag is reset to "0".

4.4.2.8 I/O control register (IOCON)

Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
IOCON	0F8H	—	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF
Bit location	Flag	Function							
IOCON.0	ALF	If CPU power down mode (PD, HPD) is activated with this bit set to "1", the outputs from ports 0, 1, 2, and 3 are switched to floating status. When this bit is "0", ports 0, 1, 2, and 3 are in output mode.							
IOCON.1	P1HZ	Port 1 becomes a high impedance input port when this bit is "1".							
IOCON.2	P2HZ	Port 2 becomes a high impedance input port when this bit is "1".							
IOCON.3	P3HZ	Port 3 becomes a high impedance input port when this bit is "1".							
IOCON.4	IZC	The 10 kohm pull-up resistance for ports 1, 2, and 3 is switched off when this bit is "1", leaving only the 100 kohm pull-up resistance.							
IOCON.5	SERR	Serial port reception error flag. This flag is set to "1" if an overrun or framing error is generated when data is received at a serial port. The flag is reset by software.							
IOCON.6	T32	Timer/counters 0 and 1 are connected serially to form a 32-bit timer/counter when this bit is set to "1". TF1 of TCON is set if a carry is generated in the 32-bit timer/counter.							
IOCON.7	—	The output data is "0" if the bit is read. This bit should not be set to "1".							

4.4.2.9 Timer 2 control register (T2CON)

Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
TMOD	0C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
Bit location	Flag	Function							
T2CON.0	CP/$\overline{RL2}$	Capture mode is set when TCLK+RCLK="0" and CP/$\overline{RL2}$ 16-bit auto reload mode is set when TCLK+RCLK="0" and CP/$\overline{RL2}$="0". CP/$\overline{RL2}$ is ignored when TCLK+RCLK="1".							
T2CON.1	C/$\overline{T2}$	Timer/counter 2 count clock designation control bit. The internal clocks (XTAL1·2÷12, XTAL1·2÷2) are used when this bit is "0", and the external clock applied to the T2 pin is passed to timer/counter 2 when the bit is "1".							
T2CON.2	TR2	Timer/counter 2 counting start and stop control bit. Timer/counter 2 commences counting when this bit is "1" and stops counting when "0".							
T2CON.3	EXEN2	T2EX timer/counter 2 external control signal control bit. Input of the T2EX signal is disabled when this bit is "0", and enabled when "1".							
T2CON.4	TCLK	Serial port transmit circuit drive clock control bit. Timer/counter 2 is switched to baud rate generator mode when this bit is "1", and the timer/counter 2 carry signal becomes the serial Port transmit clock. Note, however, that the serial ports can only use the timer/counter 2 carry signal in serial port modes 1 and 3.							
T2CON.5	RCLK	Serial port receive circuit drive clock control bit. Timer/counter 2 is switched to baud rate generator mode when this bit is "1", and the timer/counter 2 carry signal becomes the serial Port receive clock. Note, however, that the serial ports can only use the timer/counter 2 carry signal in serial port modes 1 and 3.							
T2CON.6	EXF2	Timer/counter 2 external flag. This bit is set to "1" when the T2EX timer/counter 2 external control signal level is changed from "1" to "0" while EXEN2="1". This flag serves as the timer interrupt 2 request signal. if an interrupt is generated, it must be reset to "0" by software.							
T2CON.7	TF2	Timer/counter 2 carry flag. This bit is set to "1" by a carry signal when timer/counter 2 is in 16-bit auto reload mode or in capture mode. This flag serves as the timer interrupt 2 request signal. if an interrupt is generated, it must be reset to "0" by software.							

4.5 Timer/Counters 0, 1 and 2

4.5.1 Outline

Timer/counters 0, 1 and 2 are all equipped with 16-bit binary up-counting and Read/Write functions, and can be operated independently.
All control of timer/counters 0 and 1 is handled by the timer control register (TCON 88H) and the timer mode register (TMOD 89H). And both timer/counters can be set independently to modes 0 thru 3 for a diversity of applications.
Timer/counters 0 and 1 can be operated by an external clock applied to the T0 and T1 pins (if external clock mode has been set) during soft power down mode (PD) and hard power down mode (HPD) where XTAL1·2 are stopped. Therefore, CPU power down mode can be cancelled by generating a timer/counter carry signal.
Timer/counter 2 can be fully controlled by timer 2 control register (T2CON 0C8H). There are three operational modes for a wide range of applications. Note that counting is stopped when XTAL1·2 are stopped.

4.5.2 Timer/counters 0 and 1

4.5.2.1 Outline

Timer/counters 0 and 1 are both equipped with a 16-bit binary counting function which can be operated independently.
All control of timer/counters 0 and 1 is handled by the timer control register (TCON) and the timer mode register (TMOD). And both timer/counters can be set independently to modes 0 thru 3 for a diversity of applications. The overall control circuit for timer/counters 0 and 1 is outlined in Figure 4-7 (excluding timer mode 3).

4.5.2.2 Timer/counter 0 and 1 counting control

Counting start and stop in timer/counters 0 and 1 is controlled by bit 4, TR0, and bit 6, TR1, in the timer control register (TCON 88H) as indicated in Table 4-7.
TR0 controls timer/counter 0, and TR1 controls timer/counter 1. Timer/counter operation is stopped when the bit data is “0”, and enabled when “1”.

Table 4-7 Timer control register (TCON 88H)

	Timer 1		Timer 0					
Bit	7	6	5	4	3	2	1	0
Flag	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Set		•		•				

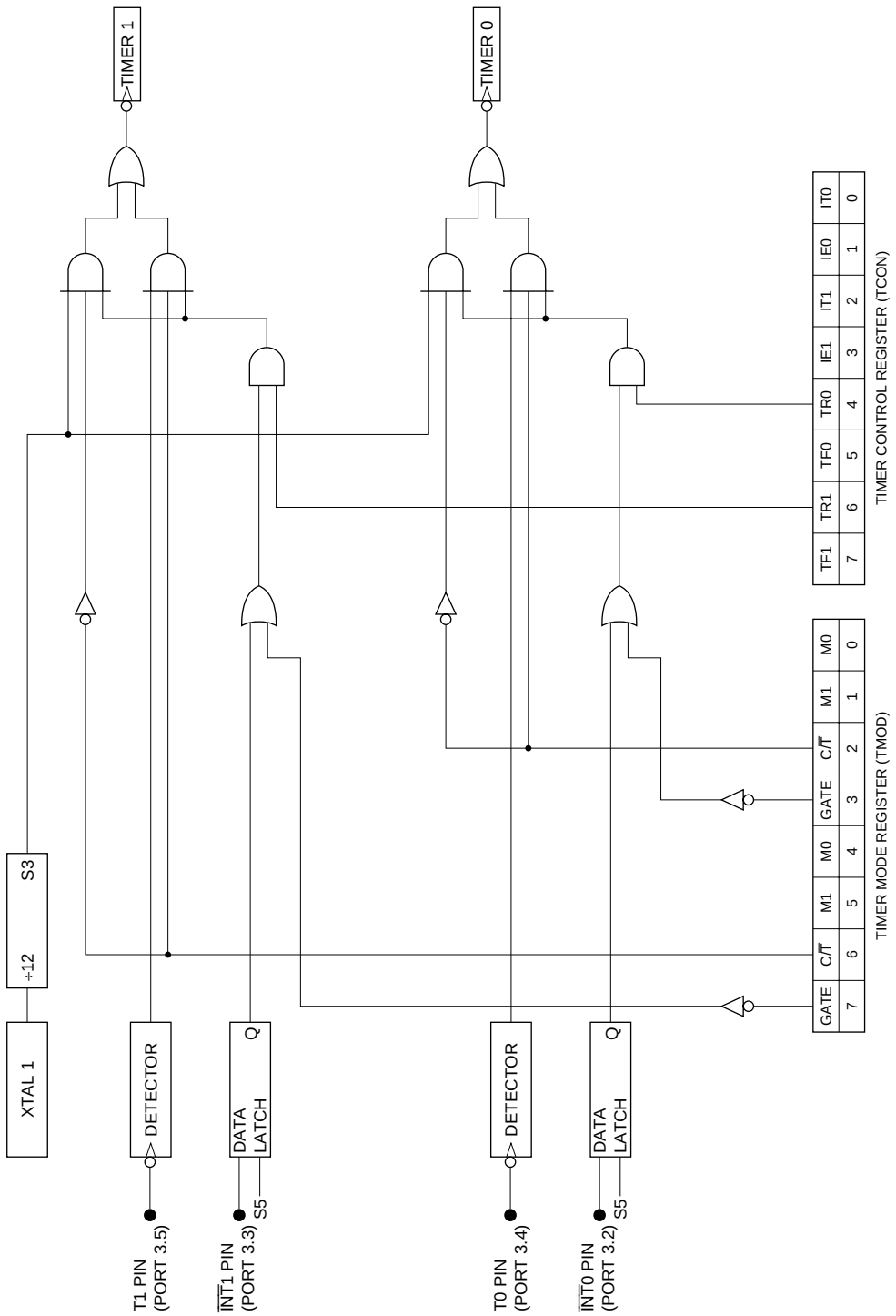


Figure 4-7 Overall clock input control circuit for timer/counters 0 and 1

4.5.2.3 Timer/counter 0 and 1 count clock designation

Designation of count clock inputs to timer/counters 0 and 1 is controlled by bit 2 and 6, C/$\overline{T}$, in the timer mode register (TMOD 89H).
Timer/counter 0 is controlled by bit 2, C/$\overline{T}$, and timer/counter 1 is controlled by bit 6, C/$\overline{T}$.
The internal clock is passed to the timer/counter when the C/$\overline{T}$ bit is “0”. This internal clock is the result of dividing XTAL1·2 by 12. The S3 timing signal (see Figure 2-9) becomes the clock.
The external clock is applied to the timer/counter when the C/$\overline{T}$ bit is “1”. The external clock applied to the T0 pin serves as the timer/counter 0 input, while the external clock applied to the T1 pin serves as the timer/counter 1 input.

Table 4-8 Timer mode register (TMOD 89H)

	Timer 1				Timer 0			
Bit	7	6	5	4	3	2	1	0
Flag	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Set		•				•		

4.5.2.3.1 External clock detector circuit for timer/counters 0 and 1

The detector circuit shown in Figure 4-8 is inserted between the timer/counters and the external clock pin.
This detector circuit operates in the following way. When the external clock applied to the T0 and T1 pins is changed from “1” to “0” level, that clock is fetched by F/Fl, and is then passed to F/F2 when the S5 timing signal appears. This F/F2 output is subsequently ANDed (logical product) with the S3 timing signal to form the timer/counter clock signal which then serves as the F/Fl reset signal. The reset F/Fl then waits for the next external clock. The “0” and “1” signal cycle widths of the respective external clocks applied to the T0 and T1 pins must have a minimum of period 12 times (12T) the XTAL1·2 oscillator clock cycle T. However, when the CPU is in PD mode or HPD mode the external clock applied to the T0 and T1 pins is input to timer/counters 0 and 1 directly. The operational time chart for this detector circuit is outlined in Figure 4-9.

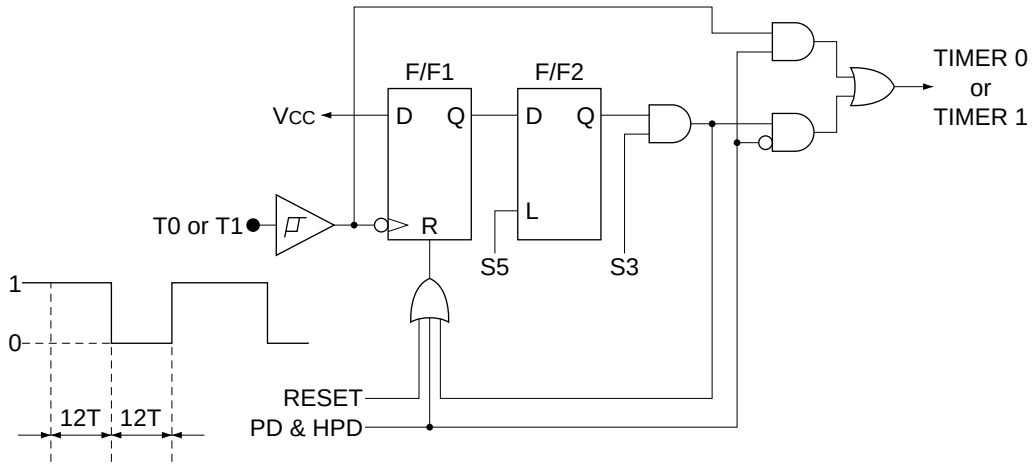


Figure 4- 8 T0 and T1 external clock detector circuit

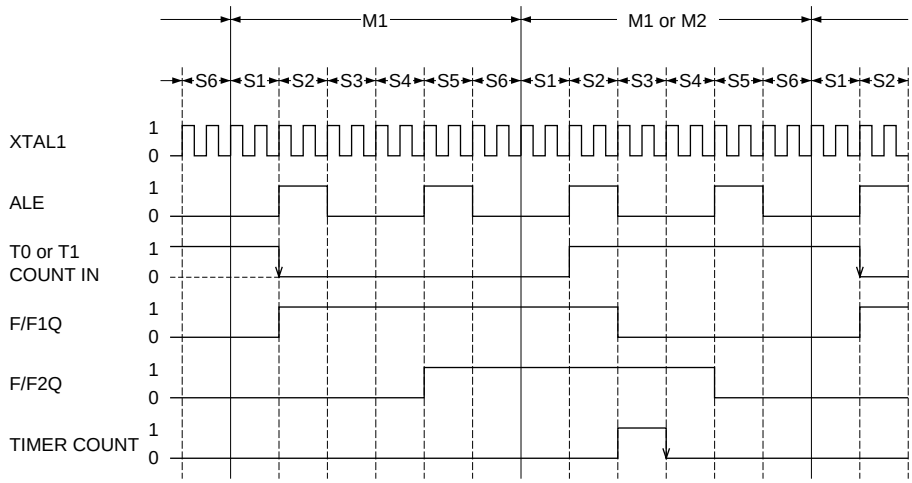


Figure 4-9 Detector circuit operational time chart

4.5.2.4 Counting control of timer/counters 0 and 1 by $\overline{\text{INT}}$ pin

In addition to control by TR0 and TR1 bits of timer control register (TCON), timer/counter 0 and 1 counting start and stop can also be controlled by the signal level applied to the external interrupt pin in accordance with the GATE data values of bits 3 and 7 in the timer mode register (TMOD 89H) indicated in Table 4-9.
Timer/counter 0 is controlled by the bit 3, GATE bit. When the GATE bit is “0”, counting is started and stopped only by TR0.
When the GATE bit is “1”, counting in timer/counter 0 is enabled if the TR0 bit and $\overline{\text{INT}}$0 pin input signal are both “1”. Counting is subsequently stopped if either is changed to “0” level.
Timer/counter 1 is controlled by the bit 7, GATE bit, the functional operation being the same as timer/counter 0. The GATE - $\overline{\text{INT}}$ timer/counter counting control circuit is outlined in Figure 4-10, and the control table is given in Table 4-10.

Table 4-9 Timer mode register (TMOD 89H)

	Timer 1				Timer 0			
Bit	7	6	5	4	3	2	1	0
Flag	GATE	C/$\overline{\text{T}}$	M1	M0	GATE	C/$\overline{\text{T}}$	M1	M0
Set	•				•			

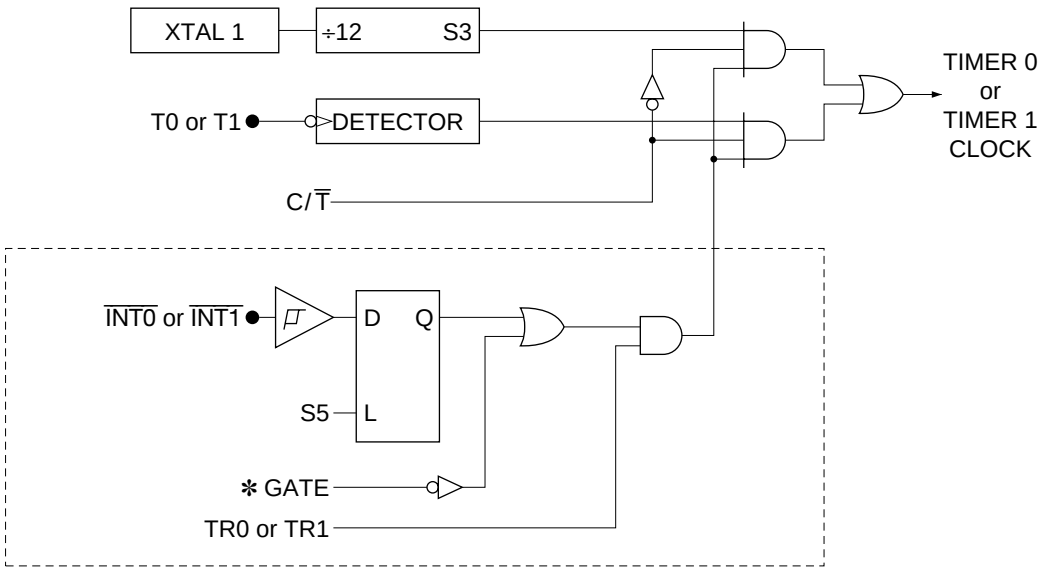


Figure 4-10 $\overline{\text{INT}}$0 and $\overline{\text{INT}}$1 timer/counter start/stop control circuit

Table 4-10 GATE·$\overline{\text{INT}}$·TR timer/counter control tables

	TIMER 0				
GATE	0	0	1	1	1
TR0	0	1	0	1	1
$\overline{\text{INT}}$0	×	×	0	0	1
RUN		•			•
STOP	•		•	•	

	TIMER 1				
GATE	0	0	1	1	1
TR1	0	1	0	1	1
$\overline{\text{INT}}$1	×	×	0	0	1
RUN		•			•
STOP	•		•	•	

4.5.2.5 Timer/counters 0/1 timer modes

4.5.2.5.1 Outline

The timer/counter 0 and 1 timer modes are set by combinations of M0 and M1 bit data in the timer mode register (TMOD 89H) shown in Table 4-11. The timer modes which can be set are 0, 1, 2, and 3.
Timer/counter 0 modes are specified by M0 and M1 of bits 0 and 1, and timer/counter 1 modes are specified by M0 and M1 of bits 4 and 5.

Table 4-11 Timer mode register (TMOD 89H)

	TIMER COUNTER 1				TIMER COUNTER 0			
Bit	7	6	5	4	3	2	1	0
Flag	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Set			•	•			•	•

4.5.2.5.2 Mode 0

M1	M0
0	0

In mode 0, timer/counters 0 and 1 both become 13-bit timer/counters by the circuit connection shown in Figures 4-11 and 4-12. TL0 and TL1 in timer/counters 0 and 1 serve as the counter for the five lower bits, and TH0 and TH1 serve as the counter for the eight upper bits.
TF0 of TCON is set by the timer/counter 0 carry signal, and TF1 of TCON is set by the timer/counter 1 carry signal. Note that the timer/counter 1 carry signal can also be used as the serial port transmission/reception clock.
Although the three upper bits of TL0 and TL1 are operative, they are invalid as signals.

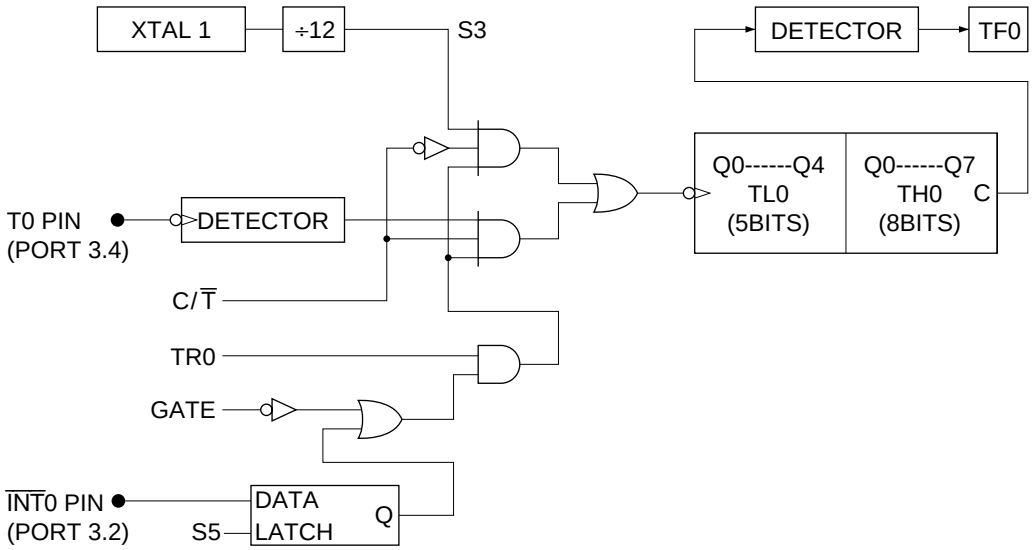


Figure 4-11 Timer/counter 0 mode 0

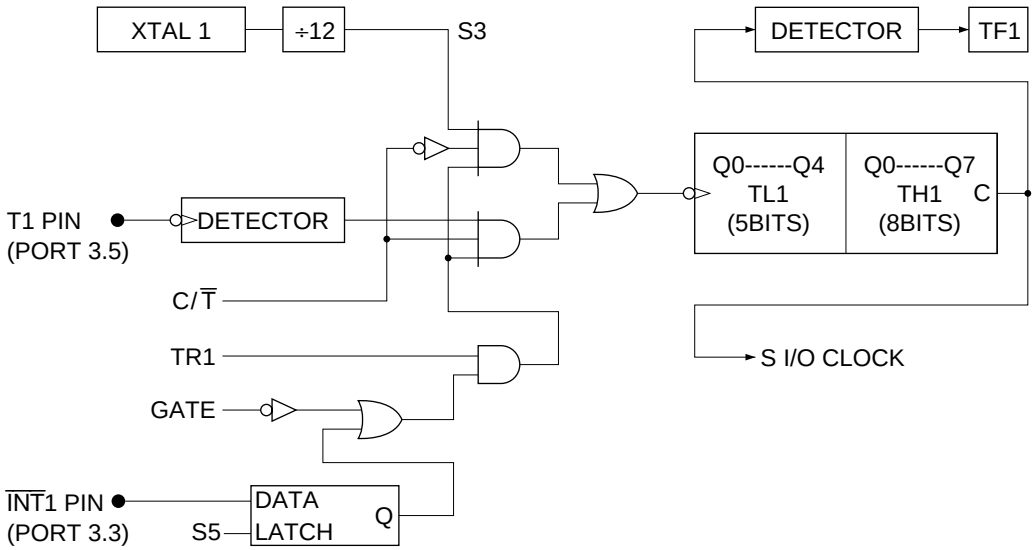


Figure 4-12 Timer/counter 1 mode 0

4.5.2.5.3 Mode 1

M1	M0
0	1

In mode 1, timer/counters 0 and 1 both become 16-bit timer/counters by the circuit connection shown in Figures 4-13 and 4-14.
TL0 and TL1 in timer/counters 0 and 1 serve as the counter for the eight lower bits, and TH0 and TH1 serve as the counter for the eight upper bits.
TL0 is set by the timer/counter 0 carry signal, and TF1 is set by the timer/counter 1 carry signal. Again note that the timer/counter 1 carry signal can also be used as the serial port transmission/reception clock.

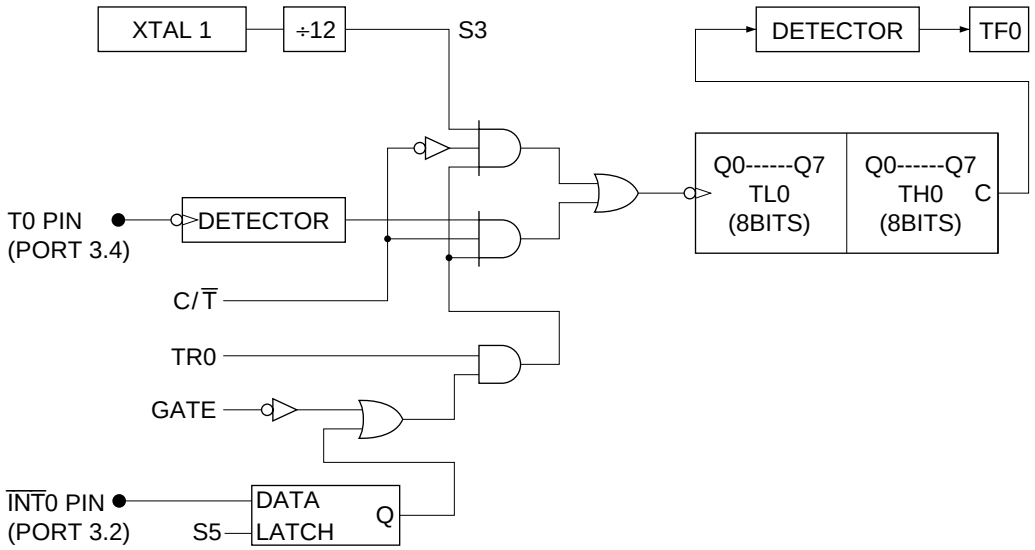


Figure 4-13 Timer/counter 0 model

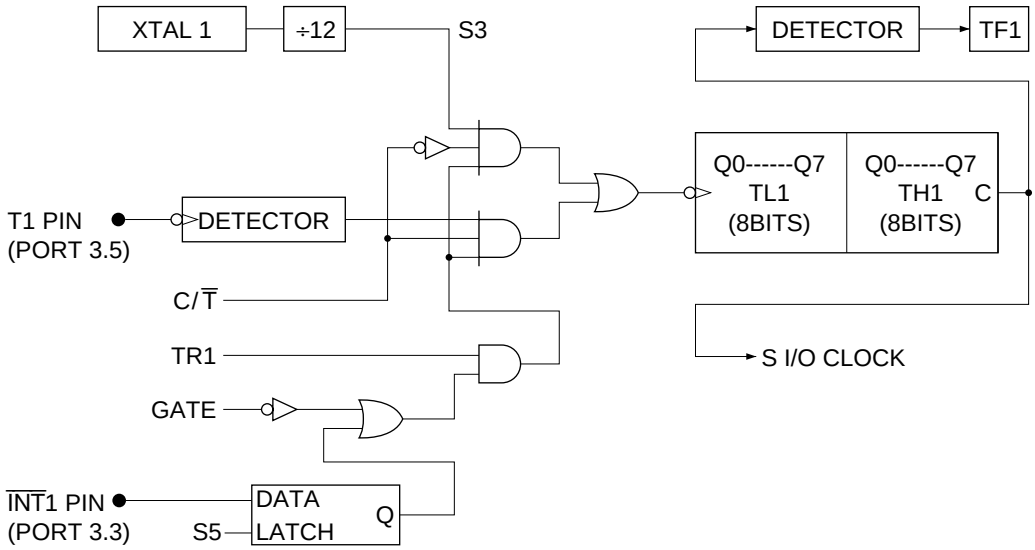


Figure 4-14 Timer/counter 1 model

4.5.2.5.4 Mode 2

M1	M0
1	0

In mode 2, timer/counters 0 and 1 both become 8-bit timer/counters with 8-bit auto reloader registers by the circuit connection shown in Figures 4-15 and 4-16. TH0 and TH1 in timer/counters 0 and 1 serve as the 8-bit auto reloader section, and TL0 and TL1 serve as the timer/counter section.

If a carry signal is generated by the 8-bit timer/counter TL0 and TL1, the respective auto reloader register data is preset into the timer/counter, and counting proceeds from the preset value.

TF0 is set by the timer/counter 0 carry signal, and TF1 is set by the timer/counter 1 carry signal. Note that the timer/counter 1 carry signal can also be used as the serial port transmission/reception clock.

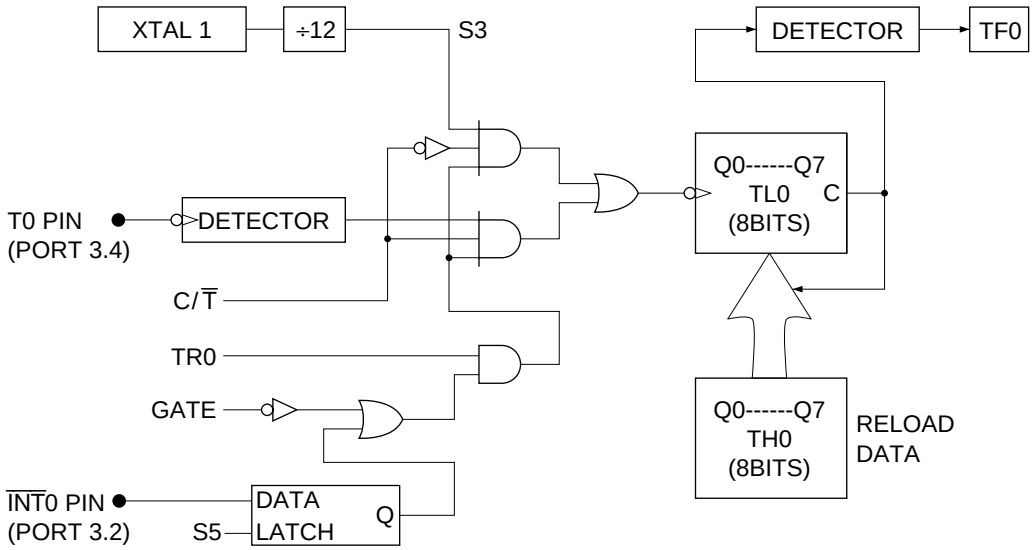


Figure 4-15 Timer/counter 0 mode 2

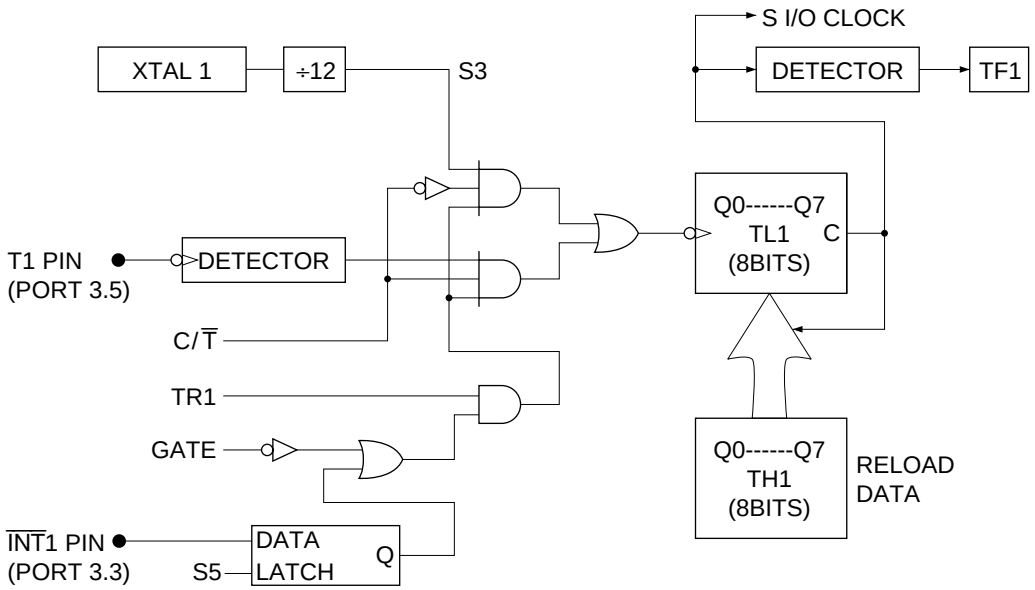


Figure 4-16 Timer/counter 1 mode 2

4.5.2.5.5 Mode 3

M1	M0
1	1

In mode 3, timer/counter 0 TL0 and TH0 become independent 8-bit timer/counters by the circuit connection shown in Figure 4-17. Timer/counter 1 does not operate when mode 3 is set. The TL0 8-bit timer/counter is controlled in the same way as the regular timer/counter 0, TF0 being set if a carry signal is generated by TL0.
The TH0 8-bit timer/counter is controlled only by TR1, and the control only covers count starting and stopping. TF1 is set by a carry signal generated by TH0.
When timer/counter 0 is set to mode 3, timer/counter 1 can operate in modes 0, 1, or 2, and be used by the serial port clock. Control of timer/counter 1 count starting and stopping in this case is handled between operating mode and mode 3. If mode 3 is set, the timer/counter 1 counting operation is stopped.

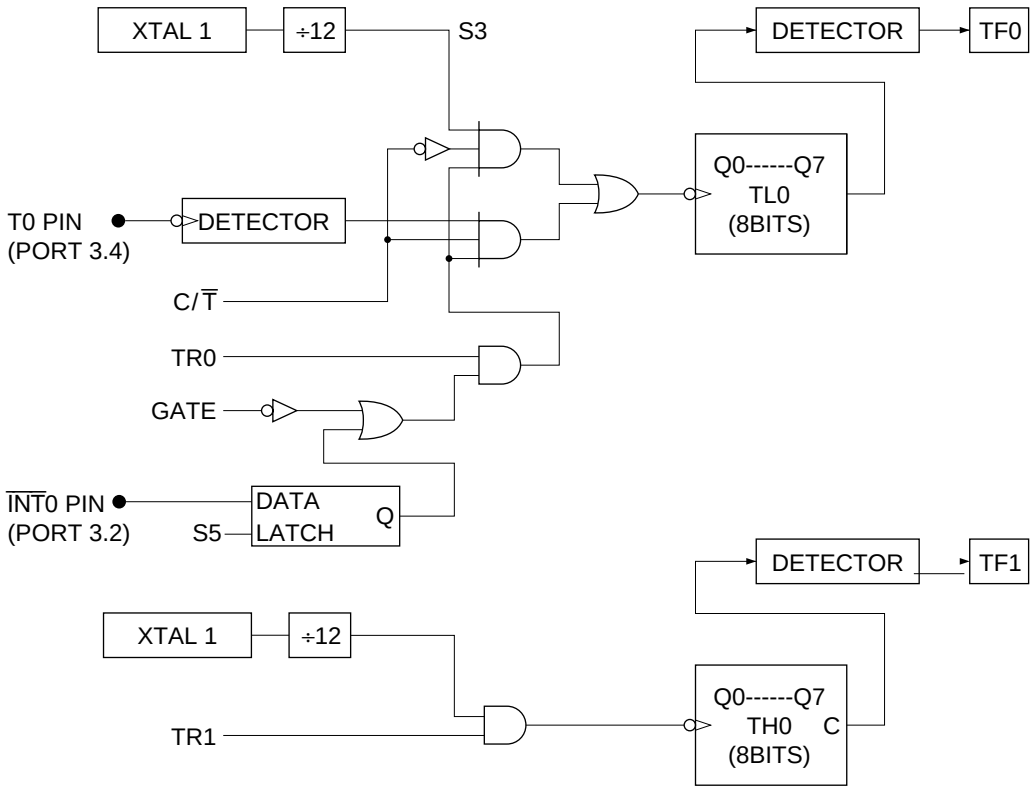


Figure 4-17 Timer/counter 0 mode 3

4.5.2.5.6 32-bit timer mode

When “1” is set in bit 6 (T32) of the I/O control register (IOCON 0F8H), timer/counters 0 and 1 are connected serially as indicated in Figure 4-18 to become a 32-bit timer/counter.
This 32-bit timer/counter is started by the following procedure. First, “0” is set in TR0, TR1, TF0, and TF1 of the timer control register (TCON 88H) to stop the timer/counter and reset the timer flag.
Next timer/counter preset data values are set in timer/counters 0 and 1, and a counter clock designation is set in bit 2 (C/$\overline{T}$) of the timer mode register (TMOD 89H).
If “1” is then set in bit 6 (T32) of the 1/0 control register (IOCON 0F8H) after completing the above procedure, the 32-bit timer/counter is established and counting is commenced. This 32-bit timer/counter is especially useful in cancelling CPU power down mode. (See power down mode cancellation.)

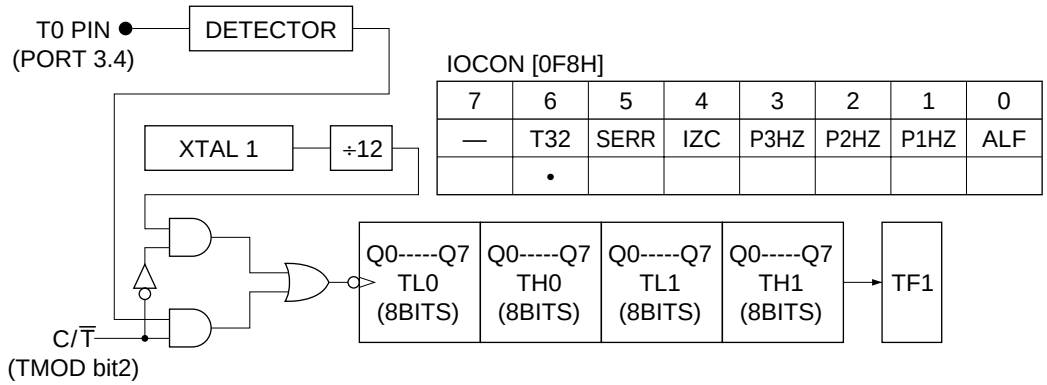


Figure 4-18 32-bit timer/counter

4.5.2.5.7 Caution about use of timer counters 0 and 1

Since the internal clock stops operation during soft power down mode (PD), the auto-reload operation is not executed if timer/counters 0 and 1 are set to mode 2 or mode 3.
If the power down mode is to be cancelled by the timer, timer/counters 0 and 1 must be set to mode 0 or mode 1.
When timers 0 and 1 are set to external clock mode, the external clock is taken in as shown in Figure 4-19 and the power down mode can be cancelled through the overflow of the timer.
If the external interrupt occurs when the T0 or T1 pin goes to “1” level and the soft power down mode (PD) is cancelled, the gate output (A) changes from “1” level to “0” level and the counter is incremented by 1.
In addition, “Q” of F/F1 is set on the trailing edge of T0 or T1.
Thus, the counter is incremented by additional 1.
The same event occurs not only by the external interrupt but also by the overflow of the timer. This is because the overflow signal of the timer is made up of the timer count value “FF” and the clock input signal “AND”. Therefore, the timer interrupt occurs when the T0 or T1 pin goes to “1” level, and the power down mode is cancelled and the counter is incremented by additional 1.
In cancelling the soft power down mode with the external interrupt, if the timer is set to external clock mode, the T0 or T1 pin must be set to “0” level. If the T0 or T1 pin is at “1” level or if the power down mode is cancelled by the overflow of the timer, the timer must be reset or the counter must be decremented by 1.

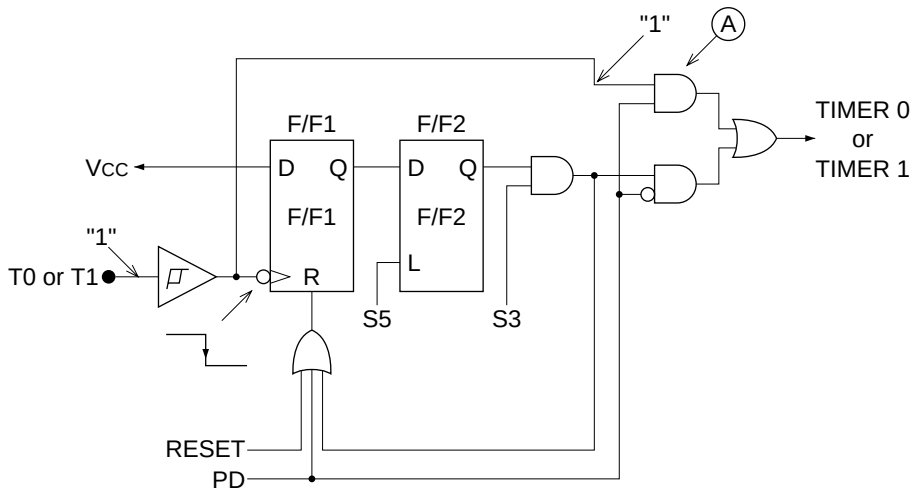


Figure 4-19 T0, T1 external clock detector circuit

4.5.2.5.8 Caution about use of timer counters 0 and 1 when setting software power down mode

When setting sofware power down mode, if the value of a timer counter by which a timer interrupt is set is immediately before overflow, the software power down mode can not be set.

(Example)

Timer 0 is in mode 1 of external clock.
Content of timer 0 is "FF".
Interrupt by timer 0 is enabled.
TO pin is "1".

If the above conditions all are established, the sofware power down mode cannot be set. This is because the AND output, shown as (A) of Fig. 4-19, becomes "1" when the software power down mode is set and timer interrupt is generated.
In this case, set the software power down mode after setting the TO pin to "0".

4.5.3 Timer/counter 2

4.5.3.1 Outline

Timer/counter 2 is equipped with 16-bit binary counting and Read/Write functions. This timer/counter is controlled entirely by timer 2 control register (T2CON 0C8H).
The operating modes are 16-bit auto reload mode, capture mode, and baud rate generator mode. Modes are specified by T2CON RCLK, TCLK, and CP/$\overline{RL2}$ bits combinations.
The internal or external clock applied to the timer/counter 2 is specified by the C/$\overline{T2}$ bit. And starting and stopping of timer/counter 2 counting is controlled by the TR2 bit. Note that timer/counter 2 counting is stopped in CPU power down mode where XTAL1·2 are stopped.

4.5.3.2 Timer 2 control register (T2CON)

The timer 2 control register (T2CON 0C8H) consists of the timer/counter 2 control bits, timer 2 internal flag (TF2), and timer 2 external flag (EXF2). The T2CON contents are outlined in Table 4-12.

Table 4-12 Timer 2 control register (T2CON 0C8H)

Bit	7	6	5	4	3	2	1	0
Flag	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

CP/$\overline{RL2}$: Capture mode is set when TCLK+RCLK=0 and CP/$\overline{RL2}$=1. The timer/counter 2 contents are passed to the capture register (RCAP2L/RCAP2H) when the level o the signal applied to the T2EX pin (bit 1 of port 1) is changed from “1” to “0” with EXEN2-1.
16-bit auto reload mode is set when TCLK+RCLK=0 and CP/$\overline{RL2}$=0. The CP/$\overline{RL2}$ data is ignored when TCLK+RCLK=1.

C/$\overline{T2}$: Timer/counter 2 clock input designation bit.
The internal clock is specified when this bit is “0” and the external clock is specified
when “1”.

TR2 : Timer/counter 2 counting start and stop control bit.
Timer/counter 2 operation is stopped when this bit is “0”, and enabled when “1”

EXEN2 : The T2EX pin control bit. The signal applied to the T2EX pin is invalid when this bit is “0”, and valid when “1”.

TCLK : Serial port transmit clock control bit. When this bit is set to “1”, timer/counter 2 is set to 16-bit auto reload operation mode, and the timer/counter 2 carry signal activates the serial port transmit circuit. This clock is only valid when serial port mode 1 or 3 has been set.

RCLK : Serial port receive clock control bit. When this bit is set to “1”, timer/counter 2 is set to 16-bit auto reload operation mode, and the timer/counter 2 carry signal activates the serial port receive circuit.
This clock is only valid when serial port mode 1 or 3 has been set.

EXF2 : Timer/counter 2 external flag bit which is set when the T2EX pin level (bit 1 of port 1) is changed from “1” to “0” at EXEN2=1. This flag serves as the timer interrupt 2 request signal. When an interrupt is generated, this flag must be reset to “0” by software.

TF2 : Timer/counter 2 internal flag bit which is set when a carry signal is generated by timer/counter 2 in 16-bit auto reload mode or capture mode. This flag serves as the timer interrupt 2 request signal. When an interrupt is generated, this flag must be reset to “0” by software.

4.5.3.3 Timer/counter 2 operation modes

Timer/counter 2 operation modes are set by combinations of the CP/$\overline{RL2}$, TCLK, and RCLK bits in timer 2 control register (T2CON 0C8H) shown in Table 4-13. The timer modes are listed in Table 4-14.

Table 4-13 Timer 2 control register (T2CON 0C8H)

Bit	7	6	5	4	3	2	1	0
Flag	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
Set			•	•				•

Table 4-14 Timer/counter 2 modes

RCLK	TCLK	CP/$\overline{RL2}$	TR2	Mode
0	0	0	1	16-bit auto reload
0	0	1	1	16-bit capture
RCLK + TCLK = 1		×	1	Baud rate generator
×	×	×	0	All operations stopped

4.5.3.3.1 16-bit auto reload mode

16-bit auto reload mode is set by making the circuit connection shown in Figure 4-20 by setting RCLK=0, TCLK=0, and CP/$\overline{RL2}$=0 as the bit conditions in timer 2 control register (T2CON). Timer/counter 2 operates in the following way when 16-bit auto reload mode is set. When a timer/counter 2 carry signal is generated, or when the signal applied to the T2EX pin (bit 1 of port 1) is changed from level “1” to “0”, the reload data in the RCAP2L and RCAP2H registers is preset in L2 and TH2 of timer/counter 2. The timer/counter thus starts counting from this preset value.

The timer/counter 2 carry signal is set in internal timer flag 2 (TF2), and the T2EX change is set in external timer flag 2 (EXF2). The TF2 and EXF2 serve as the timer interrupt 2 request signals with an interrupt call being made to address 43 (2BH) if the timer interrupt 2 has been enabled. If an interrupt routine is commenced, the TF2 and EXF2 flags must be reset to “0” by software.

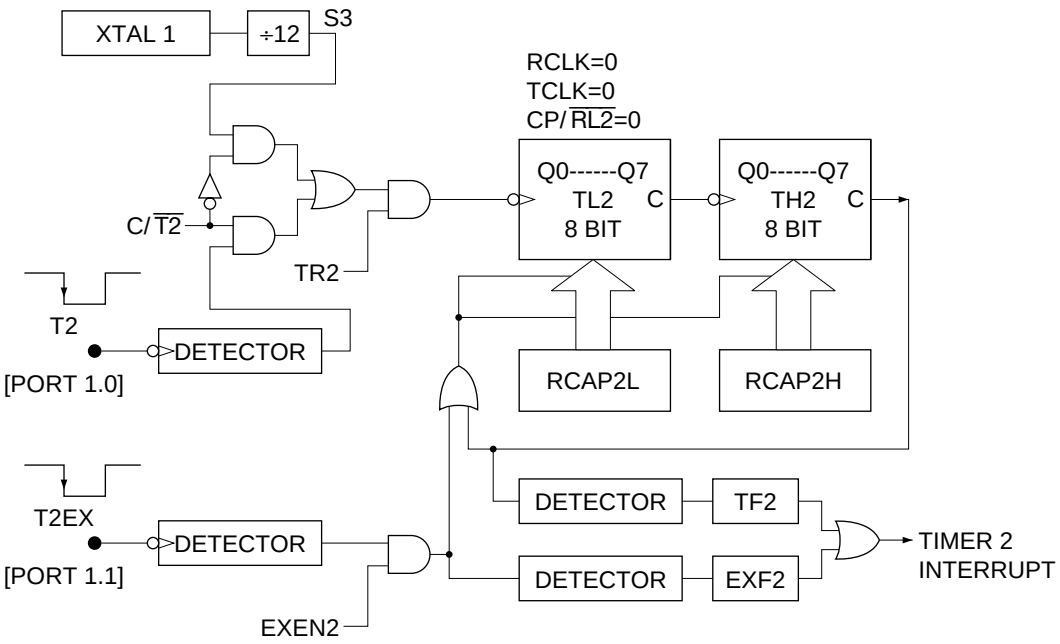


Figure 4-20 Timer/counter 2 16-bit auto reload mode circuit

4.5.3.3.2 16-bit capture mode

The 16-bit capture mode is set by making the connections shown in Figure 4-21 with the following timer 2 control register (T2CON) bit conditions, viz. RCLK=0, TCLK=0, and CP/$\overline{RL2}$=1.

Timer/counter 2 operates in the following way when 16-bit capture mode is set. When the signal applied to the T2EX pin (bit 1 of port 1) is changed from level “1” to “0”, the TL2 and TH2 count contents of timer/counter 2 are stored into capture registers RCAP2L and RCAP2H. The T2EX signal change is set in external timer flag 2 (EXF2) at this time, and a carry signal from timer/counter 2 is set in internal timer flag 2 (TF2). The EXF2 and TF2 serve as the timer interrupt 2 request signals with an interrupt call being made to address 43 (2BH) if timer interrupt 2 has been enabled. If an interrupt routine is commenced, the EXF2 and TF2 flags must be reset to “0” by software.

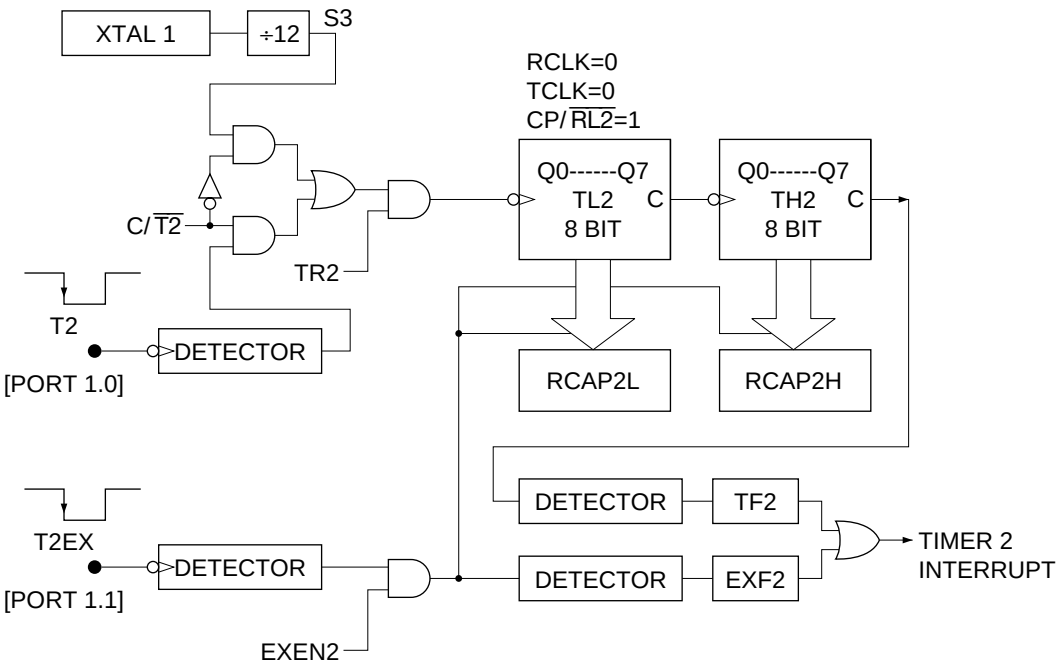


Figure 4-21 Timer/counter 2 16-bit capture mode circuit

4.5.3.3.3 16-bit baud rate generator mode

The 16-bit baud rate generator mode is set by making the connections shown in Figure 4-22 with the following timer 2 control register (T2CON) bit conditions, viz.
RCLK+TCLK=1.
Timer/counter 2 commences to operate in the following way when 16-bit baud rate generator mode is set.
Timer/counter 2 is put into 16-bit auto reload mode. When timer/counter 2 generates a carry signal, the reload data in the RCAP2L and RCAP2H registers is preset in the timer/counter 2 TL2 and TH2 and the timer/counter commences to count from that preset value. The carry signal is passed to a serial port.
The timer/counter 2 carry signal activates the serial port receive circuit when RCLK 1, and activates the transmit circuit when TCLK=1. Note, however, that the serial port can use these clocks only when the serial port is in mode 1 and 3.
When in this mode, the timer/counter 2 carry signal is not set in internal timer flag 2 (TF2). But since the change in level (from “1” to “0”) of the signal applied to the T2EX pin (bit 1 of port 1) is set in external timer flag 2, the T2EX pin can be used for ordinary external interrupt input pin. If an interrupt routine is commenced, the EXF2 flag must be reset to “0” by software.
Since timer/counter 2 is operated at 1/2 of the XTAL1·2 clock if the internal clock is used in this mode, only undefined data will be read from the timer/counter 2 TL2 and TH2 by software. Correct data, however, is read from the RCAP2L and RCAP2H registers.

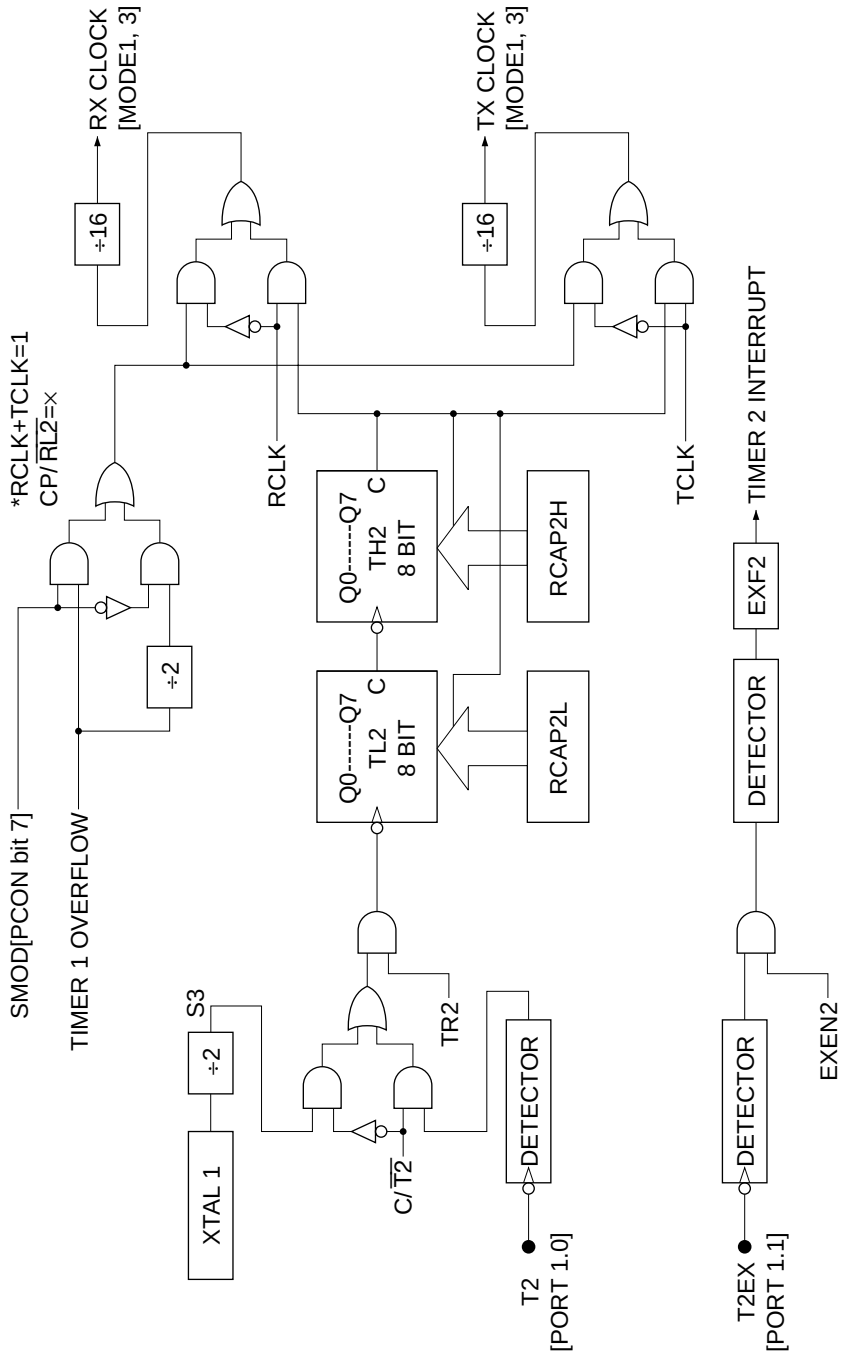


Figure 4-22 Timer/counter 2 baud rate generator mode circuit

4.5.3.4 Timer/counter 2 detector circuit

4.5.3.4.1 T2 (timer/counter 2 external clock detector)

The T2 detector circuit block diagram is shown in Figure 4-23. Operation of this circuit is outlined below. When the level of the signal applied to T2 (bit 0 of port 1) is changed from “1” to “0”, output of F/Fl becomes “1”. This output signal is then passed to F/F2 at S5 timing and F/F2 output also becomes “1”. The T2 signal change passed to F/F2 is synchronized with the S3 timing signal to become the external clock for timer/counter 2. At the same time, F/F1 is reset and waits for the next external clock input. Note that the “0” and “1” level cycle times of the external clock signal applied to the T2 pin must be at least 12 times (12T) the XTAL1·2 oscillator clock cycle time T.

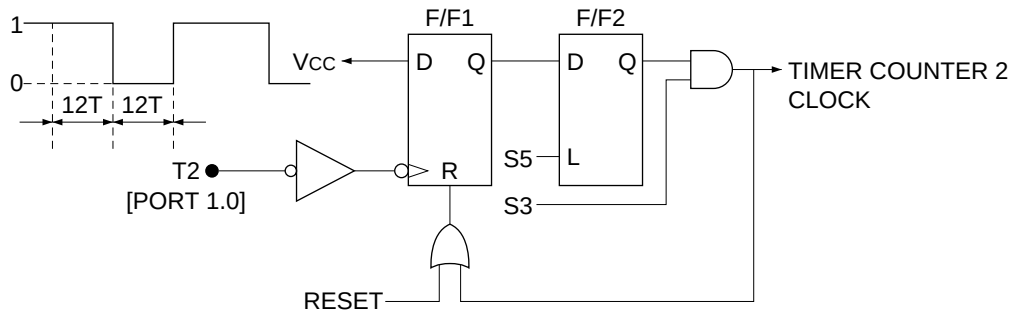


Figure 4-23 Timer/counter 2 external clock detector circuit

4.5.3.4.2 T2EX (timer/counter 2 external flag input detector)

T2EX detector circuit block diagram is shown in Figure 4-24. Operation of this circuit is outlined below. When the level of the signal applied to T2EX (bit 1 of port 1) is changed from “1” to “0”, output of F/F1 becomes “1”. This output signal is then passed to F/F2 at S2 timing and F/F2 output also becomes “1”. The T2EX signal change passed to F/F2 Q is synchronized with the S4 timing signal to become the T2EX signal for timer/counter 2. At the same time, F/Fl is reset and waits for the next T2EX input. Note that the “0” and “1” level cycle times of the external clock signal applied to the T2EX pin must be at least 12 times (12T) the XTAL1·2 oscillator clock cycle time T.

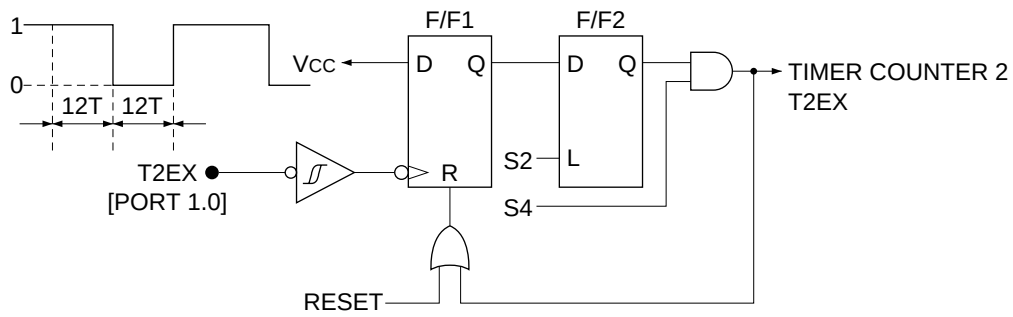


Figure 4-24 Timer/counter 2 T2EX detector circuit

4.5.3.5 Timer/counter carry signal detector circuit

The detector circuit shown in Figure 4-25 is inserted between the MSM80C154S/ MSM83C154S timer/counter carry output and the timer flag. The purpose of this detector is to prevent timer flags being set by the timer carry signal during execution of OR, AND, EOR, RESET bit, SET bit, or MOV bit instruction on the contents of the timer control register (TCON), and thereby prevent loss of timer flags by manipulated data by the time execution of instruction has been completed. Hence, even if a timer carry signal is generated during execution of an instruction, that flag will not be set while the instruction is still being executed. The flag is set at $\overline{M2}\cdot S1$ during execution of the next instruction. If a timer carry is generated during M1 thru M3 when executing a 4-machine cycle instruction, the timer flag is set during M3 or M4. See Figure 4-26 for the time chart.
In case of driving the timer/counters 0 and 1 with the external clock in the power down mode (PD, HPD), timer/counters 0 and 1 contents are incremented by falling edge of the external clock. However, after counting the maximum value of timer/counters 0 and 1, carry signals are generated and timer flags are set when the external clock level changes from “0” to “1“.

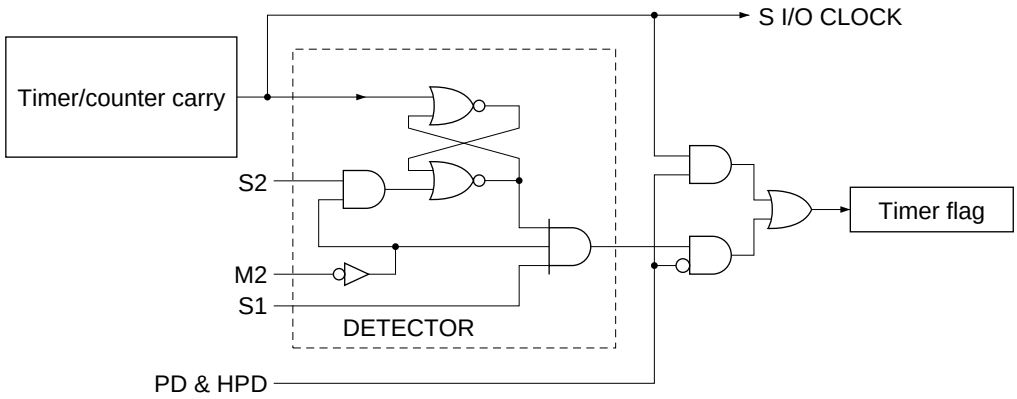


Figure 4-25 Timer/counter detector circuit

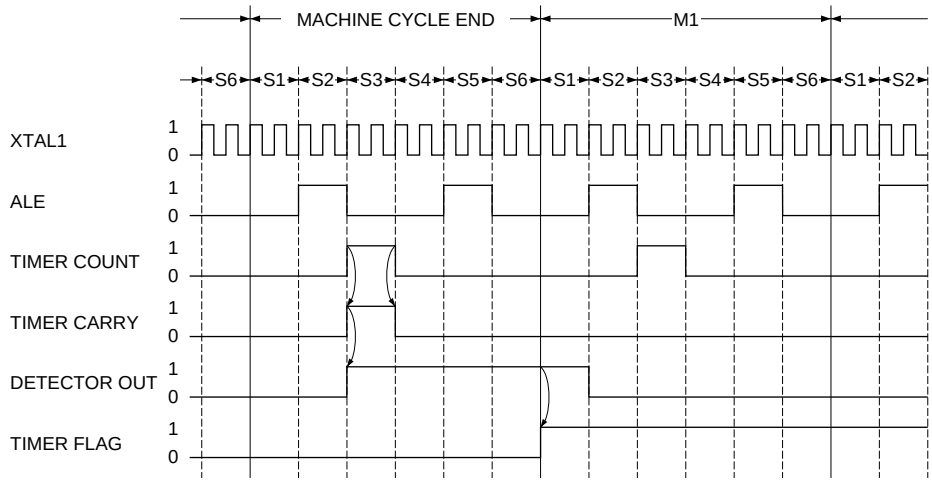


Figure 4- 26 Timer flag setting time chart

4.6 Serial Port

4.6.1 Outline

MSM80C154S/MSM83C154S is equipped with a serial port which can be used in I/O extension and UART (Universal Asynchronous Receiver/Transmitter) applications.

I/O extension mode
- Input and output of 8-bit serial data synchronized with the MSM80C154S/MSM83C154S output clock.

UART mode
- Independent transmitter and receiver circuits for full duplex communication.
- Double buffer in receiver circuit to provide a 1-frame time margin in processing received data.
- Selection of 10-bit and 11-bit frame lengths.
- Easier baud rate selection than in MSM80C31F/MSM80C51F
- Setting of different baud rates for transmitting and receiving possible Multi-processor system applications possible in 11-bit frame mode Framing and overrun error detect function

See Figure 4-27 for serial port block diagram.

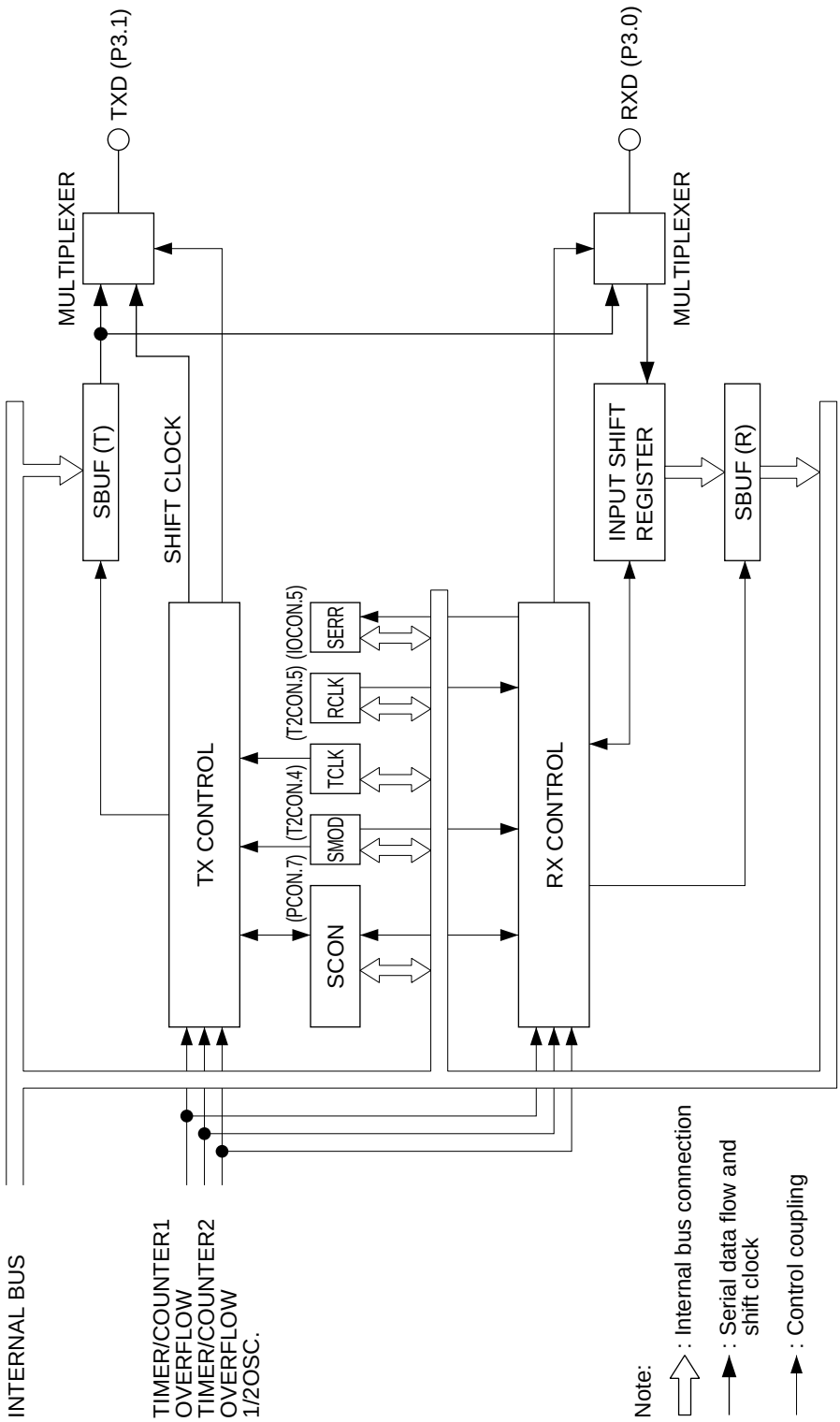


Figure 4-27 Serial port

4.6.2 Special function registers for serial port

4.6.2.1 SCON (Serial Port Control Register)

SCON is an 8-bit special function register consisting of control bits for specifying serial port operation modes and enabling/disabling data reception, storage bits for the ninth data bit transmitted and received during 11-bit frame UART mode, and the serial port status flag.
In addition to specifying SCON by data address 98H, each bit can be specified by bit addresses.
The functions of each SCON bit are listed in Table 4-15, and the functions of each operational mode specified by SCON are indicated in Table 4-16.

Table 4-15 SCON

Bit	Symbol	Function
0	RI	"End of reception" flag. This is the interrupt request flag set by hardware when reception of one frame has been completed. The interrupt is generated by ORing with the T1 flag. Since the flag cannot be cleared by hardware, it must be cleared by software.
1	TI	"End of transmission" flag. This is the interrupt request flag set by hardware when transmission of one frame has been completed. The interrupt is generated by ORing with the RI flag. Since the flag cannot be cleared by hardware, it must be cleared by software.
2	RB8	Storage of the 9th bit of the data received during 11-bit frame UART mode (mode 2 or 3). When in 10-bit frame UART mode (mode 1), the stop bit is stored.
3	TB8	Storage of the 9th bit of the data to be sent during 11-bit frame UART mode (mode 2 or 3).
4	REN	Receive enable bit. Reception is not activated if REN is not set..
5	SM2	If SM2 is set when in 11-bit frame UART mode (mode 2 or 3) and the 9th bit of the received data is "1", the received data is accepted and loaded into SBUF and RB8, and the RI flag is set. If the 9th bit of the received data is "0", on the other hand, the received data is disregarded and the SBUF, RB8, and RI flags remain unchanged. This function is used to enable communication between processors in multi-processor systems. If SM2 is set when in 10-bit frame UART mode (mode 1) and the normal stop bit cannot be received (stop bit "0"), the received data is disregarded, and the SBUF, RB8, and RI flags remain unchanged. When SM2="0", however, the sent data is received irrespective of the "0"/"1" status of the stop bit. SM2 must be cleared when in I/O extension mode (mode 0).
6	SM1	Used in setting serial port operation mode. See Table 4-16.
7	SM0	Used in setting serial port operation mode. See Table 4-16.

Table 4-16 Serial port operation modes

SM0	SM1	Mode	Function	Baud rate
0	0	0	I/O extension	1/12 FOSC
0	1	1	10-bit frame UART	Vareable
1	0	2	11-bit frame UART	1/32 FOSC or 1/64 FOSC
1	1	3	11-bit frame UART	Vareable

Note: FOSC denotes frequency of fundamental oscillator (XTAL1·2).

4.6.2.2 SBUF (serial port buffer register)

SBUF is an 8-bit special function register used to store transmitting and receiving data. Although the SBUF is specified by the same data address 99H for both writing and reading, physically separate registers are specified. That is, the sending circuit SBUF is specified by instructions where SBUF is used as a destination operand, and the receiving circuit SBUF is specified by instructions where SBUF is used as a source operand.

4.6.2.3 TCLK

TCLK controls selection of the baud rate clock source for the transmitting circuit when in mode 1 or 3.
The timer/counter 2 overflow becomes the transmitting circuit baud rate clock source when TCLK is set in mode 1 or 3. And the timer/counter 1 overflow becomes the transmitting circuit baud rate clock source if TCLK is cleared.
TCLK has no effect on the baud rate clock source when in mode 0 or 2. TCLK is located at bit 4 of T2CON (timer/counter 2 control register) specified by data address 0C8H. This bit can also be specified by bit address 0CCH.

4.6.2.4 RCLK

RCLK controls selection of the baud rate clock source for the receiving circuit when in mode 1 or 3.
The timer/counter 2 overflow becomes the receiving circuit baud rate clock source when RCLK is set in mode 1 or 3. And the timer/counter 1 overflow becomes the receiving circuit baud rate clock source if RCLK is cleared.
RCLK has no effect on the baud rate clock source when in mode 0 or 2. RCLK is located at bit 5 of T2CON (timer/counter 2 control register) specified by data address 0C8H. This bit can also be specified by bit address 0CDH.

4.6.2.5 SMOD

SMOD controls the division of the baud rate clock source when the serial port is in UART mode (mode 1, 2, or 3).
If SMOD is cleared when in mode 1 or 3, the timer/counter 1 overflow frequency divided by 2 becomes the baud rate clock source. And if SMOD is set, the timer/counter 1 overflow becomes the baud rate clock source.
When TCLK is set in mode 1 or 3, however, and timer/counter 2 is the baud rate clock source for the transmitting circuit, SMOD has no effect on the transmitting baud rate. And if RCLK has been set, timer/counter 2 becomes the baud rate source for the receiving circuit, and SMOD has no effect on the receiving baud rate.
If SMOD is cleared in mode 2, 1/2 OSC (oscillator frequency divided by 2) divided by 2 becomes the baud rate clock source. And if SMOD is set, 1/2 OSC becomes the baud rate clock source.
SMOD is located at bit 7 of PCON (power control register) specified by data address 87H. Designation by bit address is not possible.
See Table 4-17 for the corresponding baud rate clock sources for TCLK, RCLK, and SMOD.

Table 4-17 Corresponding baud rate clock sources for TCLK, RCLK, and SMOD

Mode	TCLK or RCLK	SMOD	Baud rate colck source
0	X	X	MSM83C154S fundamental timing
1	0	0	T/C1 overflow divided by 2
	0	1	T/C1 overflow
	1	X	T/C2 overflow
2	X	0	1/2 OSC divided by 2
	X	1	1/2 OSC
3	0	0	T/C1 overflow divided by 2
	0	1	T/C1 overflow
	1	X	T/C2 overflow

Note: X : Don't care
T/C1 : Timer/counter1
T/C2 : Timer/counter2
1/2 OSC : Oscillator frequency (XTAL1•2) divided by 2

4.6.2.6 SERR

SERR is the status flag set when a framing error or overrun error is generated during UART mode (mode 1, 2, or 3).

Framing error:

The SERR flag is set when no stop bit is detected in UART mode. Framing error is detected irrespective of the data reception conditions set by SM2.

Overrun error:

The SERR flag is also set when the next data is ready to be transferred from the input shift register to the SBUF which is already full in UART mode. Note that an overrun error is only detected when the data reception conditions set by SM2 have been satisfied. Although the SERR flag is set by hardware when a framing or overrun error is generated, it is not an interrupt request flag. The flag must be checked by software to determine whether it has been set or not. The flag must also be cleared by software. Since the SERR flag is set by the logical OR of framing and overrun errors, it is not possible to determine whether the error is a framing or overrun error simply by checking the flag.

SERR is located at bit 5 of IOCON (I/O control register) specified by data address 0F8H. This bit can also be specified by bit address 0FDH.

4.6.3 Operating modes

4.6.3.1 Mode 0

4.6.3.1.1 Outline

Mode 0 is the I/O extension mode where input and output of 8-bit data via RXD (P3.0) is synchronized with the output clock from TXD (P3.1).
The baud rate in mode 0 is fixed to 1/12th of the fundamental oscillator (XTAL1·2) frequency to enable the serial port to operate synchronized with the basic MSM80C154S/MSM83C154S timing.
A block diagram of the mode 0 serial port is shown in Figure 4-28, the operational timing chart is shown in Figure 4-29, and the serial port operation timing in relation to the basic MSM80C154S/MSM83C154S timing is shown in Figure 4-30.

4.6.3.1.2 Mode 0 baud rate

In mode 0, the baud rate is determined by the following equation to synchronize operations with the basic MSM80C154S/MSM83C154S timing.

$$B = F_{OSC} \times \frac{1}{12}$$

where B is baud rate, and F_{OSC} is the fundamental (XTAL1·2) frequency.

4.6.3.1.3 Mode 0 transmit operation

Data output is commenced by writing data in SBUF.
The SBUF data is obtained sequentially from RXD about one machine cycle after completion of the SBUF data writing instruction, the LSB appearing first.
Two states after commencing the LSB output, output of the TXD synchronized clock is commenced. This synchronized clock is at level “0” from the latter half of S3 thru to the first half of S6, and at “1” level from the latter half of S6 thru to the first half of S3. The transmit circuit is initialized immediately following completion of output of the MSB, and the TI flag is set at the first M1·S3 after that.

4.6.3.1.4 Mode 0 receive operation

Data input is commenced when REN=“1” and R1=“0” is achieved by an instruction used to set REN or by an instruction used to clear the RI flag (or by an instruction which does both simultaneously).
Output of the TXD synchronizing clock is commenced following nine states after REN=“1” and R1=“0” is attained. The synchronized clock is at level “0” from the latter half of S3 thru to the first half of S6, and at level “1” from the latter half of S6 thru to the first half of S3.
The RXD data is read sequentially into an input shift register in the serial port just before the synchronized clock is changed from “0” to “1”.
When input of the 8-bit data is completed, loading of the input shift register data into SBUF (with the LSB at the beginning of the input data) occurs at the same time that receiving circuit is initialized. The RI flag is then set at the first M1·S3 after completion of input of the 8-bit data.

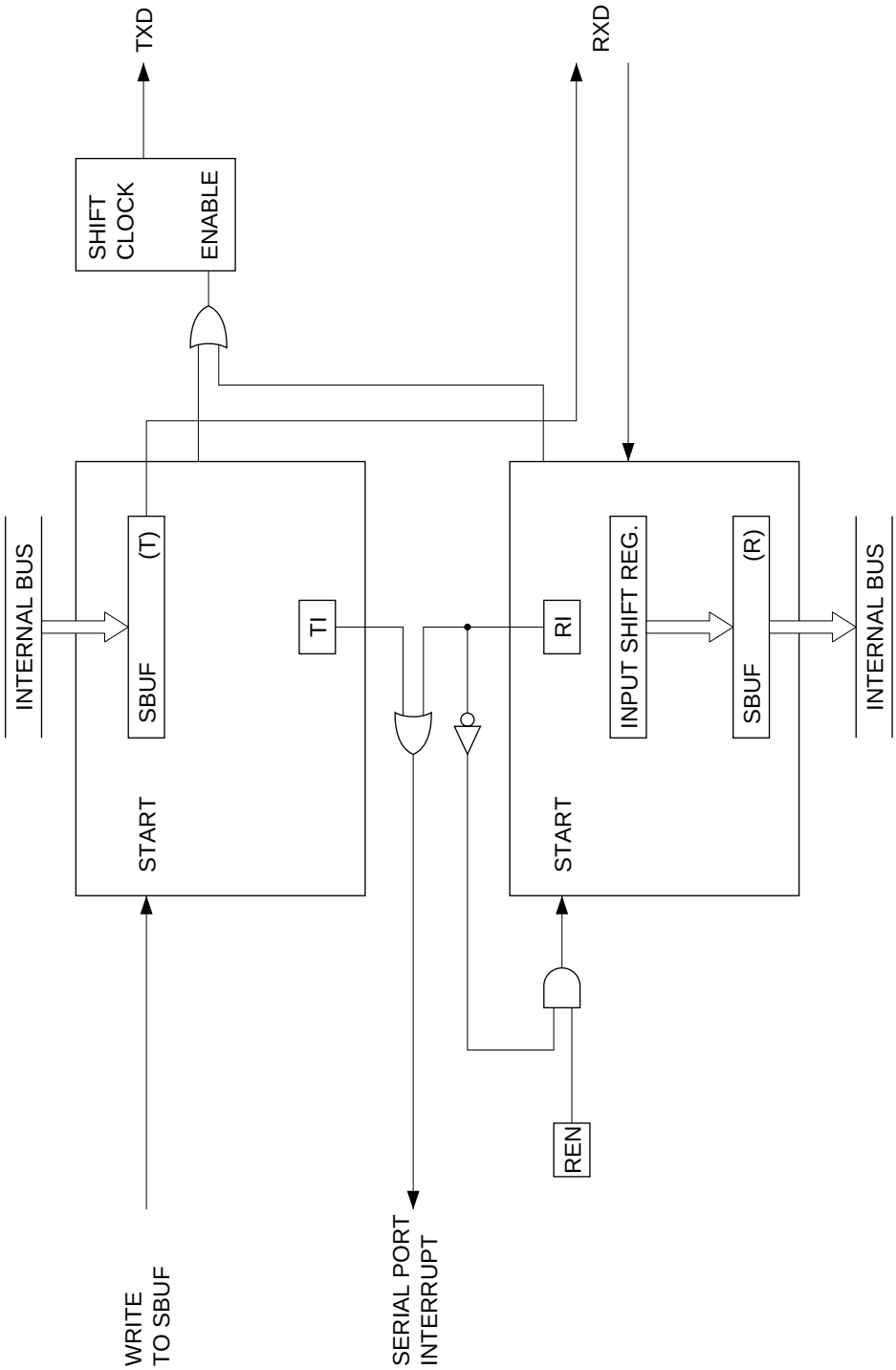


Figure 4-28 Serial port (mode 0)

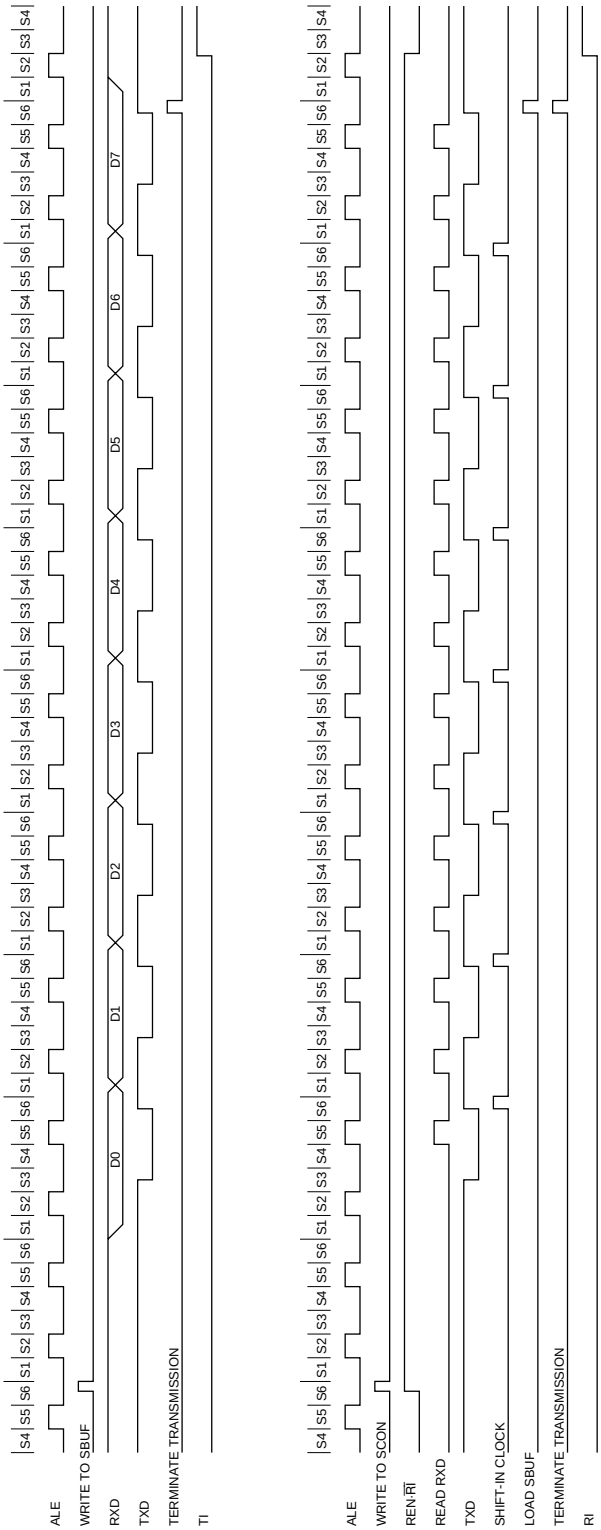


Figure 4-29 Serial port (mode 0) timing chart

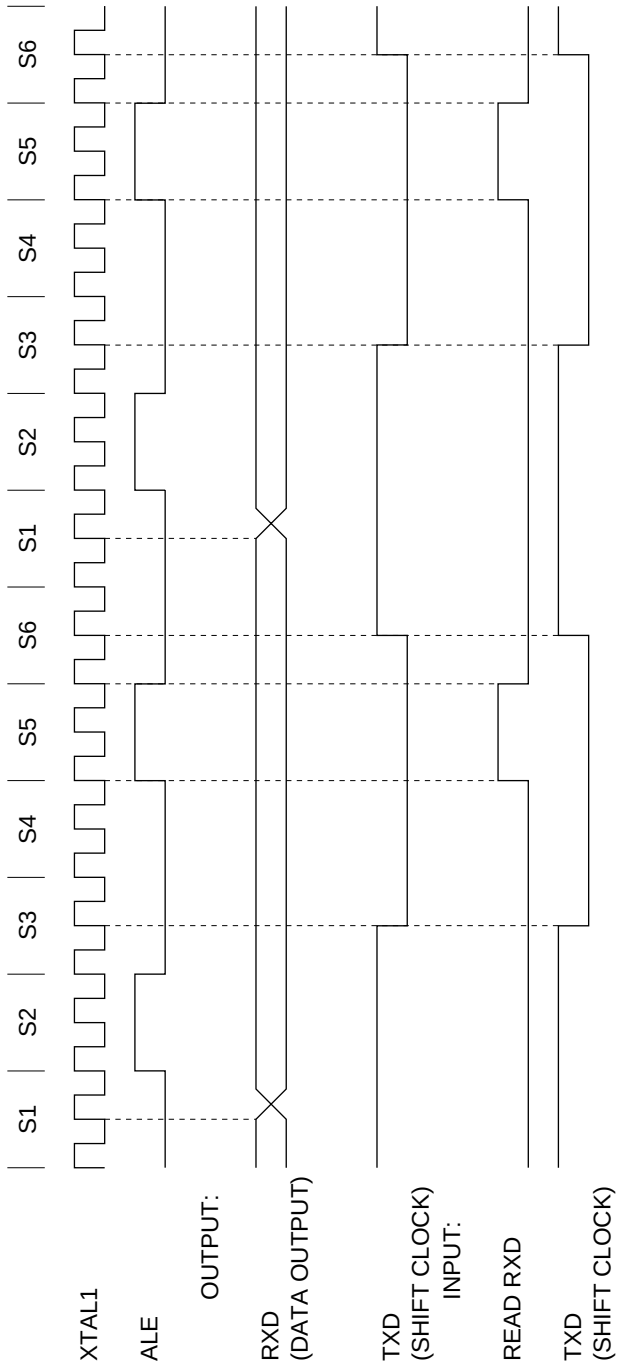


Figure 4-30 Serial port (mode 0) timing and corresponding basic MSM80C154S/ MSM83C154S timing

4.6.3.2 Mode 1

4.6.3.2.1 Outline

Mode 1 is the 10-bit frame UART mode (with one start bit, eight data bits, and one stop bit) where the baud rate may be set to any value depending on the timer/counter 1 or timer/counter 2 setting.
A block diagram of the serial port in mode 1 is shown in Figure 4-31, and the operational timing chart is given in Figure 4-32.

4.6.3.2.2 Mode 1 baud rate

The timer/counter 1 or timer/counter 2 overflow can be set as the baud rate clock source in mode 1 by independent TCLK and RCLK setting for the transmit and receive circuits.
Where the baud rate is determined by the timer/counter 1 overflow, baud rate is determined by the overflow frequency and SMOD value according to the following equations.

$$B = f_{TC1} \times \frac{1}{2} \times \frac{1}{16} \quad \text{(SMOD=0)}$$

$$B = f_{TC1} \times \frac{1}{16} \quad \text{(SMOD=1)}$$

where B is the baud rate and f_{TC1} is the timer/counter 1 overflow frequency.
When timer/counter 1 is used as a timer (internal clock) in auto reload mode (mode 2), the baud rate is determined by the following equations.

$$B = f_{OSC} \times \frac{1}{12} \times \frac{1}{256\text{-}D_{TH1}} \times \frac{1}{2} \times \frac{1}{16} \quad \text{(SMOD=0)}$$

$$B = f_{OSC} \times \frac{1}{12} \times \frac{1}{256\text{-}D_{TH1}} \times \frac{1}{16} \quad \text{(SMOD=1)}$$

where B is the baud rate, f_{OSC} the fundamental (XTAL1·2) frequency, and D_{TH1} the TH1 contents (expressed in decimal).
Where the timer/counter 2 overflow serves as the baud rate clock source, the baud rate is determined by the overflow frequency irrespective of the SMOD value.
When timer/counter 2 is used as a counter (external clock), the baud rate is determined by the following equation.

$$B = f_{T2} \times \frac{1}{65536\text{-}D_{RCAP2}} \times \frac{1}{16}$$

where B is the baud rate, f_{T2} the frequency of the clock applied to the T2 pin, and D_{RCAP2} the contents of RCAP2L and RCAP2H (expressed in decimal).
Or if timer/counter 2 is used as a timer, the baud rate is determined in the following way.

$$B = f_{OSC} \times \frac{1}{2} \times \frac{1}{65536\text{-}D_{RCAP2}} \times \frac{1}{16}$$

where B is the baud rate, f_{OSC} the fundamental frequency (XTAL1·2), and D_{RCAP2} the contents of RCAP2L and RCAP2H (expressed in decimal).

4.6.3.2.3 Mode 1 transmit operation

The transmit basic clock (TXCLOCK in Figure 4-31) is obtained from the overflow of a hexadecimal free-run counter where the timer/counter 1 or timer/counter 2 overflow is used as the clock.
Transmission is commenced when transmit data is written in SBUF.
The start bit, the eight SBUF data bits (with the LSB first), and the stop bit are transmitted sequentially from TXD synchronized with the basic clock.
As soon as output of the eight data bits has been completed, the transmit circuit is initialized, and the T1 flag is set at the first M1·S3 after the completion of that output.

4.6.3.2.4 Mode 1 receive operation

The receive circuit timing is generated by a hexadecimal counter which uses the timer/counter 1 or timer/counter 2 overflow as the clock, and the input data received from RXD is bit synchronized. That is, at the same time that reception is started following input of the start bit, the hexadecimal counter commences to count up, and with one complete round of the hexadecimal counter corresponding to one bit of received data, reception is continued by the receive circuit.
The RXD change from “1” to “0” is regarded as the beginning of the start bit for commencement of reception.
When this “1” to “0” RXD change is detected, the hexadecimal counter which had been stopped in reset status commences to count up. When the hexadecimal counter is in state 7, 8, and 9, the start bit is sampled, and is accepted as valid if at least two of the three sampled values are “0”, thereby enabling data reception to continue. If two or three of the sampled values are “1”, the start bit becomes invalid, and the receive circuit is initialized when the hexadecimal counter reaches state 10.
The reception data is sampled when the hexadecimal counter is in state 7, 8, and 9, and the more common value of the three sampled values is read sequentially as data into the input shift register.
If the hexadecimal counter is in state 10 during the period of the next bit (that is, the stop bit) after the eight bits of data have been received, and if the conditions stated below are satisfied, the input shift register data (the LSB being read first) is loaded into SBUF, and the sampled stop bit is read into RB8, thereby initializing the receive circuit. The RI flag is set at the first M1·S3 after that.
Conditions: (1) RI=“0”
(2) SM2=“0”, or SM2=“1” and sampled stop bit=“0”
If the above conditions are not satisfied, the received data is disregarded, and the receive circuit is initialized without change to the SBUF, RB8, and RI flags.
Since the receive circuit is double buffered (input shift register and SBUF), processing of the previous receive data may be completed within the interval up to the stop bit period of the next frame.

4.6.3.2.5 Mode 1 UART error detection

If the following two conditions are satisfied when the hexadecimal counter is in state 10 during reception of the stop bit, it is assumed that new data is received before processing of the previously received data has been completed. Hence, an overrun error is generated, and the new data is lost. The SERR flag is set at the first M1·S3 after the hexadecimal counter has reached state 10. Note that the previous SBUF (R) data is preserved.

Conditions: (1) RI="1"
(2) SM2="0", or SM2="1" and sampled stop bit="1"

And if the sampled stop bit is "0" when the hexadecimal counter is in state 10, it is assumed that correct frame synchronization has not been achieved. Hence, a framing error is detected, and the SERR flag is set at the first M1·S3 after that. Serial port reception is not effected by the UART error detector circuit detecting an overrun or framing error and only the status flag being set.

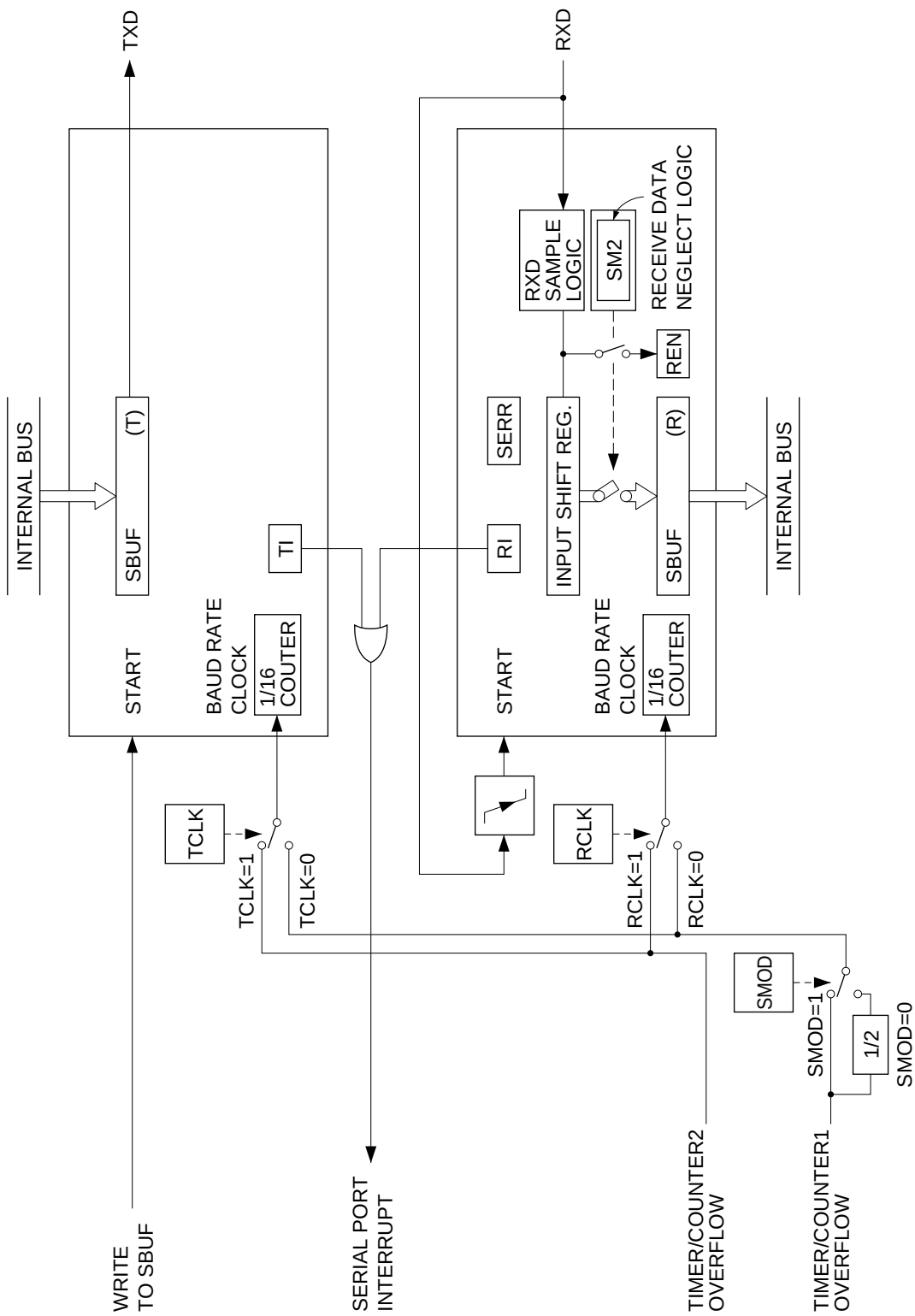


Figure 4-31 Serial port (mode 1)

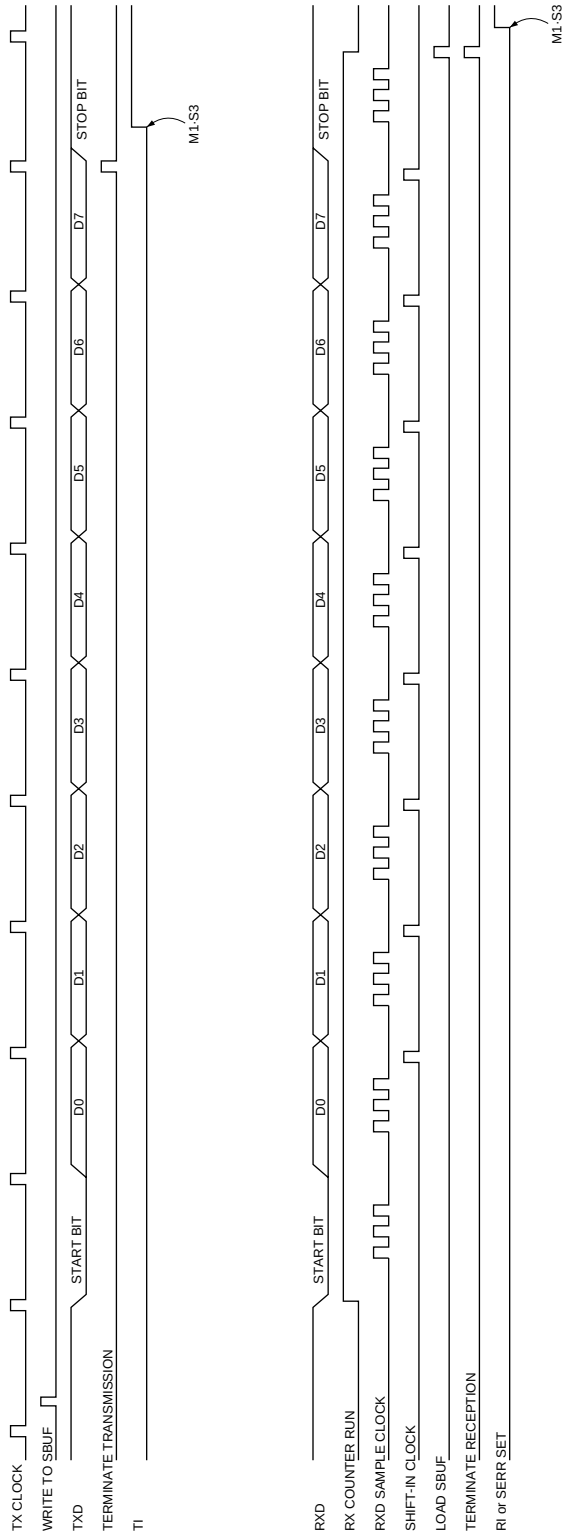


Figure 4-32 Serial port (mode 1) timing chart

4.6.3.3 Mode 2

4.6.3.3.1 Outline

Mode 2 is an 11-bit frame UART mode (with one start bit, eight data bits, one multipurpose data bit, and one stop bit) where the baud rate is 1/64th or 1/32nd of the fundamental oscillator (XTAL1·2) frequency.
A block diagram of the serial port in mode 2 is shown in Figure 4-33, and the operational timing chart is given in Figure 4-34.

4.6.3.3.2 Mode 2 baud rate

Since the fundamental oscillator frequency divided by two serves as the baud rate clock source in mode 2, the baud rate is determined by the SMOD value according to the following equations.

$$B = fosc \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{16} \quad (SMOD=0)$$

$$B = fosc \times \frac{1}{2} \times \frac{1}{16} \quad (SMOD=1)$$

where B is the baud rate and fosc the fundamental oscillator (XTAL1·2) frequency.

4.6.3.3.3 Mode 2 transmit operation

The transmit basic clock (TXCLOCK in Figure 4-34) is obtained from a hexadecimal free-run counter overflow where the frequency of 1/2XTAL1·2 (fundamental oscillator frequency divided by 2) divided by 2 (when SMOD=0) or the 1/2XTAL1·2 frequency (when SMOD=1) is used as the clock.
Transmission is commenced when transmit data is written in SBUF. The start bit, the eight SBUF data bits (with the LSB first), TB8, and the stop bit are transmitted sequentially from the TXD synchronized with the basic clock.
As soon as the TB8 output has been completed, the transmit circuit is initialized, and the T1 flag is set at the first M1·S3 after the completion of that output.

4.6.3.3.4 Mode 2 receive operation

The receive circuit timing is generated by a hexadecimal counter overflow where the frequency of 1/2XTAL1·2 (fundamental oscillator frequency divided by 2) divided by 2 (when SMOD=0) or the 1/2XTAL1·2 frequency (when SMOD=1) is used as the clock, and the input data received from the RXD is bit synchronized. That is, at the same time that reception is started following input of the start bit, the hexadecimal counter commences to count up, and with one complete round of the hexadecimal counter corresponding to one bit of received data, reception is continued by the receive circuit. Therefore, the reception data baud rate must be equal to the period of a single round of the hexadecimal counter.
The RXD change from “1” to “0” is regarded as the beginning of the start bit where reception is commenced.

When this “1” to “0” RXD change is detected, the hexadecimal counter which had been stopped in reset status commences to count up. When the hexadecimal counter is in state 7, 8, and 9, the start bit is sampled, and is accepted as valid if at least two of the three sampled values are “0”, thereby enabling data reception to continue. If two or three of the sampled values are “1”, the start bit becomes invalid, and the receive circuit is initialized when the hexadecimal counter reaches state 10.
The receive data is sampled when the hexadecimal counter is in state 7, 8, and 9, and the more common value of the three sampled values is read sequentially as data into the input shift register.
If the hexadecimal counter is in state 10 during the period of the next bit (that is, the multi-purpose data bit) after the eight bits of data have been received, and if the conditions stated below are satisfied, the input shift register data (the LSB being read first) is loaded into SBUF, and the sampled multi-purpose data bit is read into RB8. And when the hexadecimal counter is in state 10 during the period of the next after that (that is, the stop bit) the receive circuit is initialized.
The RI flag is set at the first M1·S3 after that.
Conditions: (1) R1=“0”
(2) SM2=“0”, or SM2=“1” and sampled multi-purpose data bit=“1”
If the above conditions are not satisfied when the hexadecimal counter is in state 10 during the multi-purpose data bit interval, the received data is disregarded, the SBUF, RB8, and RI flags remain unchanged, and the receive circuit is initialized when the hexadecimal counter is in state 10 during the stop bit interval.
Since the receive circuit is double buffered (input shift register and SBUF), processing of the previous receive data may by completed within the interval up to the multipurpose data bit period of the next frame.

4.6.3.3.5 Mode 2 UART error detection

If the following two conditions are satisfied when the hexadecimal counter is in state 1 0 during reception of a multi-purpose data bit, it is assumed that new data is received before processing of the previously received data has been completed. Hence, an overrun error is generated, and the new data is lost. The SERR flag is set at the first M1·S3 after the hexadecimal counter has reached state 10 during the stop bit interval. Note that the previous SBUF (R) data is preserved.
Conditions: (1) R1 =“1”
(2) SM2=“0”, or SM2=“1” and sampled multi-purpose data bit=“1”
And if the sampled stop bit is “0” when the hexadecimal counter is in state 10, it is assumed that correct frame synchronization has not been achieved. Hence, a framing error is detected, and the SERR flag is set at the first M1·S3 after that. Serial port reception is not effected by the UART error detector circuit detecting an overrun or framing error and only the status flag being set.

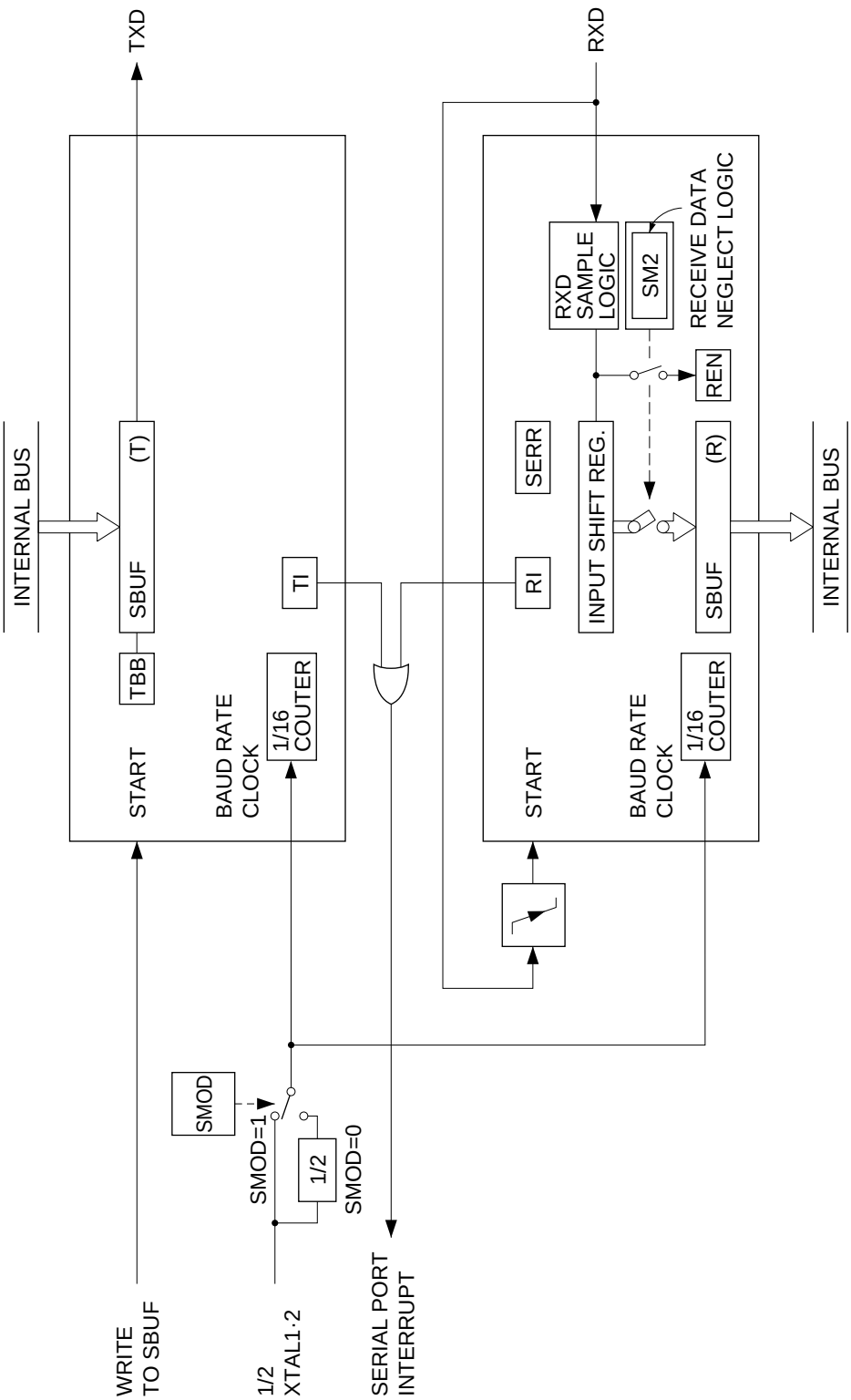


Figure 4-33 Serial port (mode 2)

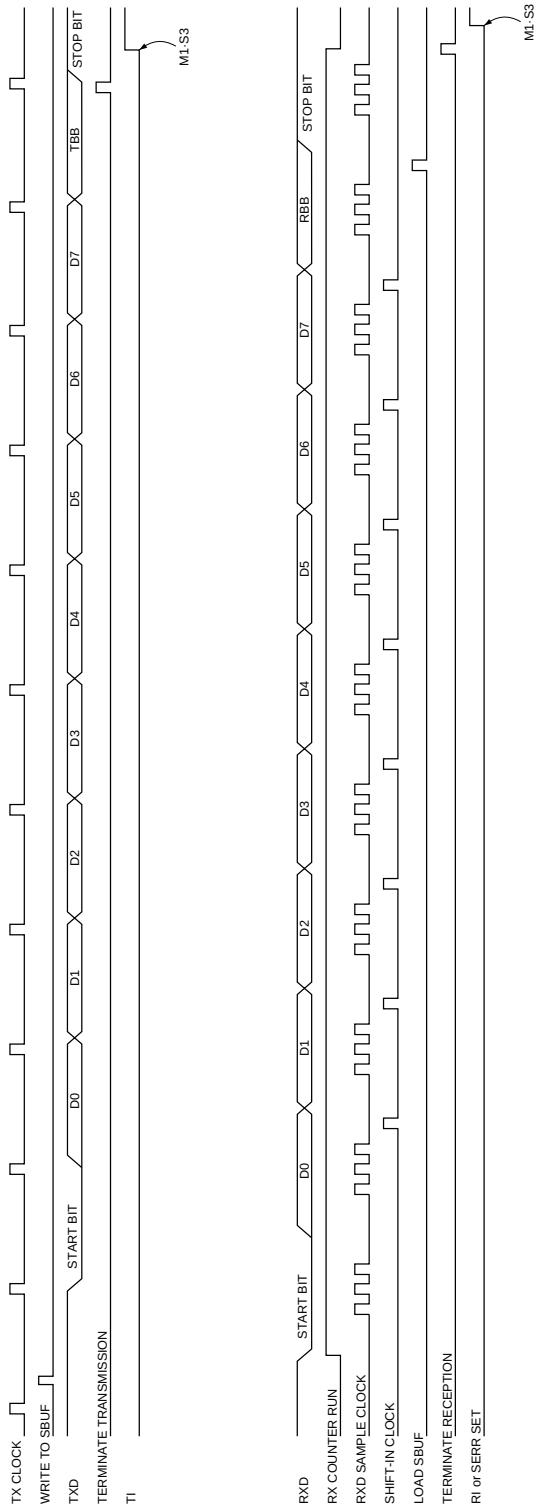


Figure 4-34 Serial port (mode 2) timing chart

4.6.3.4 Mode 3

4.6.3.4.1 Outline

Mode 3 is another 11-bit frame UART mode (with one start bit, eight data bits, one multi-purpose data bit, and one stop bit). Whereas the baud rate is 1/64th or 1/32nd of the fundamental oscillator frequency in mode 2, the mode 3 baud rate can be freely selected according to the timer/counter 1 or timer/counter 2 setting. Apart from the ability to vary the baud rate, mode 3 is identical to mode 2.
A block diagram of the serial port in mode 3 is shown in Figure 4-35, and the operational timing chart is given in Figure 4-36.

4.6.3.4.2 Mode 3 baud rate

As in mode 1, the timer/counter 1 or timer/counter 2 overflow can be set as the baud rate clock source in mode 3 by independent TCLK and RCLK setting for the transmit and receive circuits.
Where the baud rate is determined by the timer/counter 1 overflow, baud rate is determined by the overflow frequency and SMOD value according to the following equations.

$$B = f_{TC1} \times \frac{1}{2} \times \frac{1}{16} \quad \text{(SMOD=0)}$$

$$B = f_{TC1} \times \frac{1}{16} \quad \text{(SMOD=1)}$$

Where B is the baud rate and f_{TC1} is the timer/counter 1 overflow frequency.
When timer/counter 1 is used as a timer in auto reload mode (mode 2), the baud rate is determined by the following equations.

$$B = f_{OSC} \times \frac{1}{12} \times \frac{1}{256\text{-}D_{TH1}} \times \frac{1}{2} \times \frac{1}{16} \quad \text{(SMOD=0)}$$

$$B = f_{OSC} \times \frac{1}{12} \times \frac{1}{256\text{-}D_{TH1}} \times \frac{1}{16} \quad \text{(SMOD=1)}$$

where B is the baud rate, f_{OSC} the fundamental oscillator (XTAL1·2) frequency, and D_{TH1} the TH1 contents (expressed in decimal).
Where the timer/counter 2 overflow serves as the baud rate clock source, the baud rate is determined by the overflow frequency irrespective of the SMOD value.
When timer/counter 2 is used as a counter, the baud rate is determined by the following equation.

$$B = f_{T2} \times \frac{1}{65536\text{-}D_{RCAP2}} \times \frac{1}{16}$$

where B is the baud rate, f_{T2} the frequency of the clock applied to the T2 pin, and D_{RCAP2} the contents of R_{CAP2L} and R_{CAP2H} (expressed in decimal).
Or if timer/counter 2 is used as a timer, the baud rate is determined by the following way.

$$B = f_{OSC} \times \frac{1}{2} \times \frac{1}{65536\text{-}D_{RCAP2}} \times \frac{1}{16}$$

where B is the baud rate, f_{OSC} the fundamental oscillator (XTAL1·2) frequency, and D_{RCAP2} the contents of R_{CAP2L} and R_{CAP2H} (expressed in decimal).

4.6.3.4.3 Mode 3 transmit operation

The transmit basic clock (TXCLOCK in Figure 4-36) is obtained from a hexadecimal free-run counter overflow where timer/counter 1 or timer/counter 2 overflow is used as the clock.
Transmission is commenced when transmit data is written in SBUF.
The start bit, the eight SBUF data bits (with the LSB first), TB8, and the stop bit are transmitted sequentially from the TXD synchronized with the basic clock.
As soon as the TB8 output has been completed, the transmit circuit is initialized, and the T1 flag is set at the first M1·S3 after the completion of that output.

4.6.3.4.4 Mode 3 receive operation

The receive circuit timing is generated by a hexadecimal counter overflow where timer/counter 1 or timer/counter 2 overflow is used as the clock, and the input data received from the RXD is bit synchronized. That is, at the same time that reception is started following input of the start bit, the hexadecimal counter commences to count up, and with one complete round of the hexadecimal counter corresponding to one bit of received data, reception is continued by the receive circuit. Therefore, timer/counter 1 must be set so that the period of a single round of the hexadecimal counter is equal to the reception data baud rate.
The RXD change from “1” to “0” is regarded as the beginning of the start bit where reception is commenced.
When this “1” to “0” RXD change is detected, the hexadecimal counter which had been stopped in reset status commences to count up. When the hexadecimal counter is in state 7, 8, and 9, the start bit is sampled, and is accepted as valid if at least two of the three sampled values are “0”, thereby enabling data reception to continue. If two or three of the sampled values are “1”, the start bit becomes invalid, and the receive circuit is initialized when the hexadecimal counter reaches state 10.
The reception data is sampled when the hexadecimal counter is in state 7, 8, and 9, and the more common value of the three sampled values is read sequentially as data into the input shift register.
If the hexadecimal counter is in state 10 during the period of the next bit (that is, the multi-purpose data bit) after the eight bits of data have been received, and if the conditions stated below are satisfied, the input shift register data (the LSB being read first) is loaded into SBUF, and the sampled multi-purpose data bit is read into RB8. And when the hexadecimal counter is in state 10 during the period of the next after that (that is, the stop bit) the receive circuit is initialized.
The RI flag is set at the first M1·S3 after that.
Conditions: (1) RI=“0”
(2) SM2=“0”, or SM2=“1” and sampled multi-purpose data bit=“1”

If the above conditions are not satisfied when the hexadecimal counter is in state 10 during the multi-purpose data bit interval, the received data is disregarded, the SBUF, RB8, and RI flags remain unchanged, and the receive circuit is initialized when the hexadecimal counter is in state 10 during the stop bit interval.
Since the receive circuit is double buffered (input shift register and SBUF), processing of the previous receive data may be completed within the interval up to the multipurpose data bit period of the next frame.

4.6.3.4.5 Mode 3 UART error detection

Mode 3 UART error detection is identical to mode 2 UART error detection.
If the following two conditions are satisfied when the hexadecimal counter is in state 10 during reception of a multi-purpose data bit, it is assumed that new data is received before processing of the previously received data has been completed. Hence, an overrun error is generated, and the new data is lost. The SERR flag is set at the first M1·S3 after the hexadecimal counter has reached state 10 during the stop bit interval. Note that the previous SBUF (R) data is preserved.
Conditions: (1) RI =“1”
(2) SM2=“0”, or SM2=“1” and sampled multi-purpose data bit=“1”
And if the sampled stop bit is “0” when the hexadecimal counter is in state 10, it is assumed that correct frame synchronization has not been achieved. Hence, a framing error is detected, and the SERR flag is set at the first M1·S3 after that.
Serial port reception is not effected by the UART error detector circuit detecting an overrun or framing error and only the status flag being set.

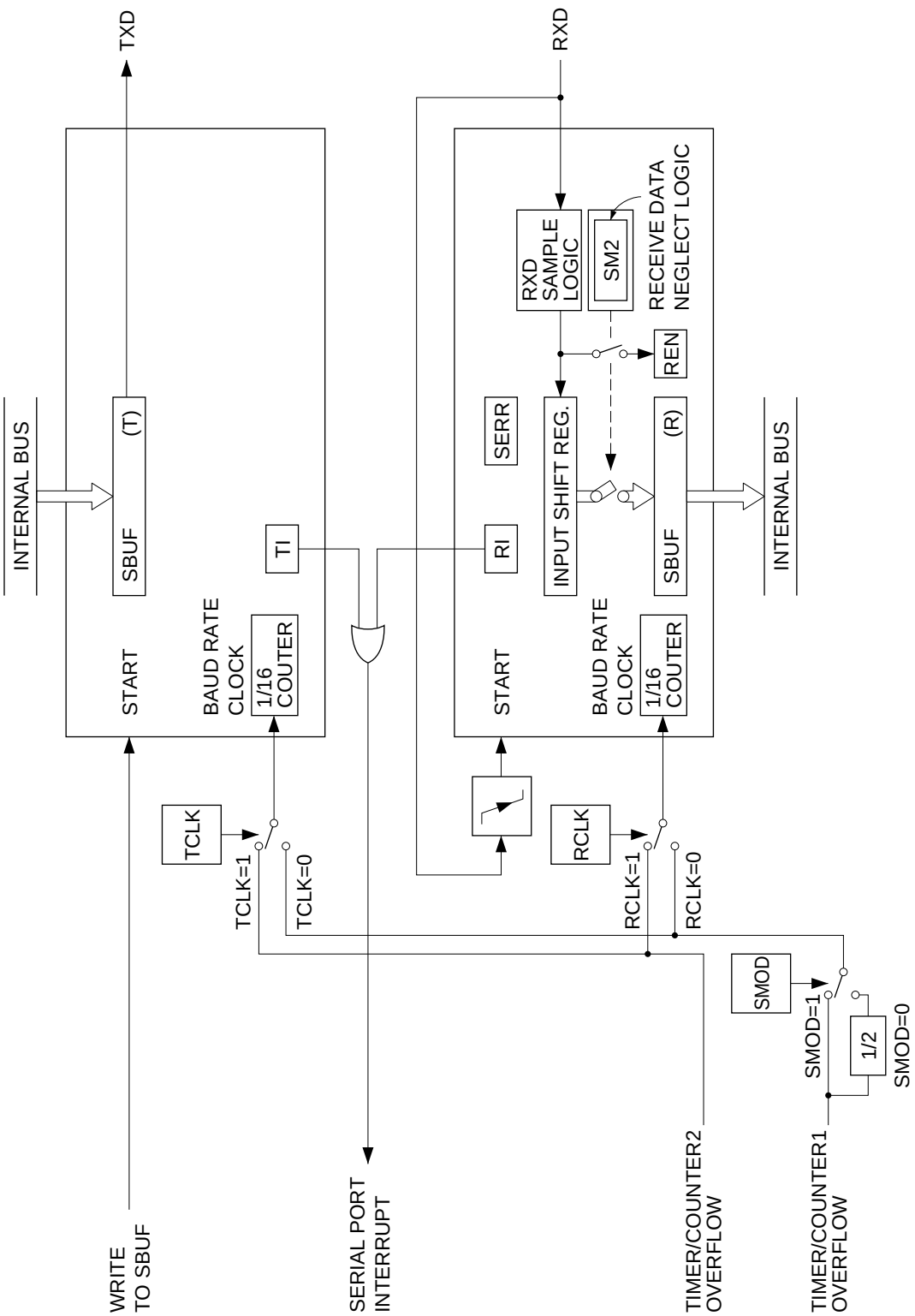


Figure 4-35 Serial port (mode 3)

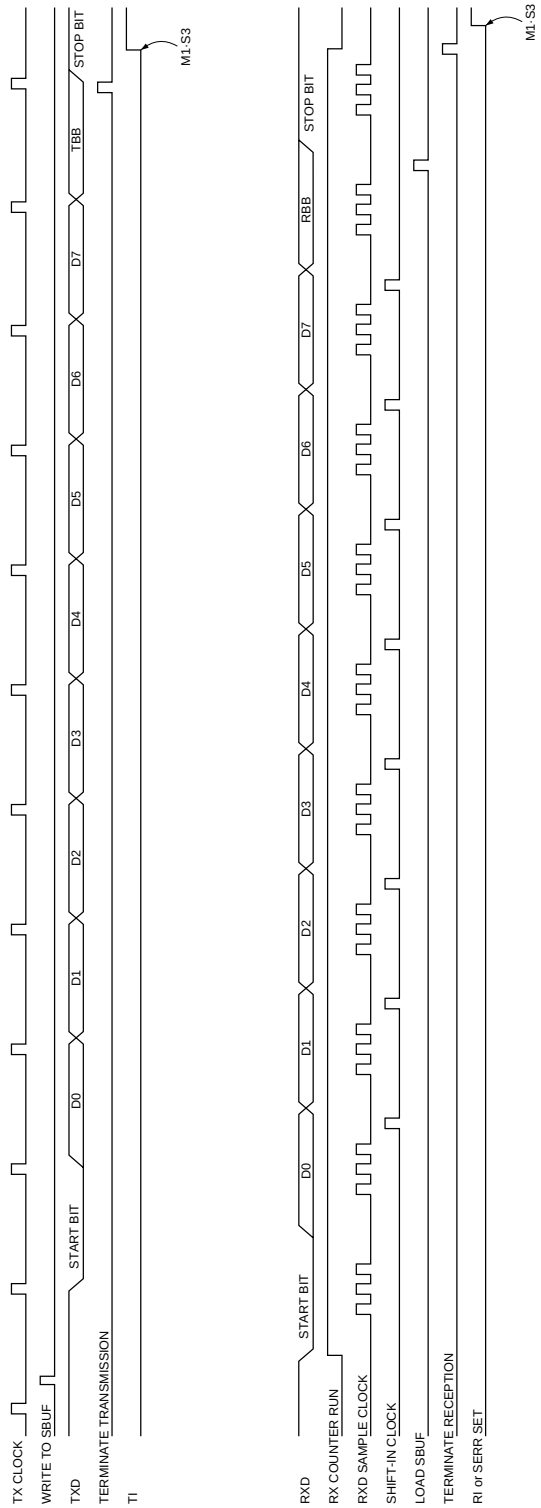


Figure 4-36 Serial port (mode 3) timing chart

4.6.4 Serial port application examples

4.6.4.1 I/O extension

I/O extension can be achieved by using the serial port in mode 0. An input extension example is shown in Figure 4-37 and the corresponding timing chart is shown in Figure 4-38. Following output of the latch pulse from PX.X, REN="1" and R1="0" are set for shift in of 74LS165 data.

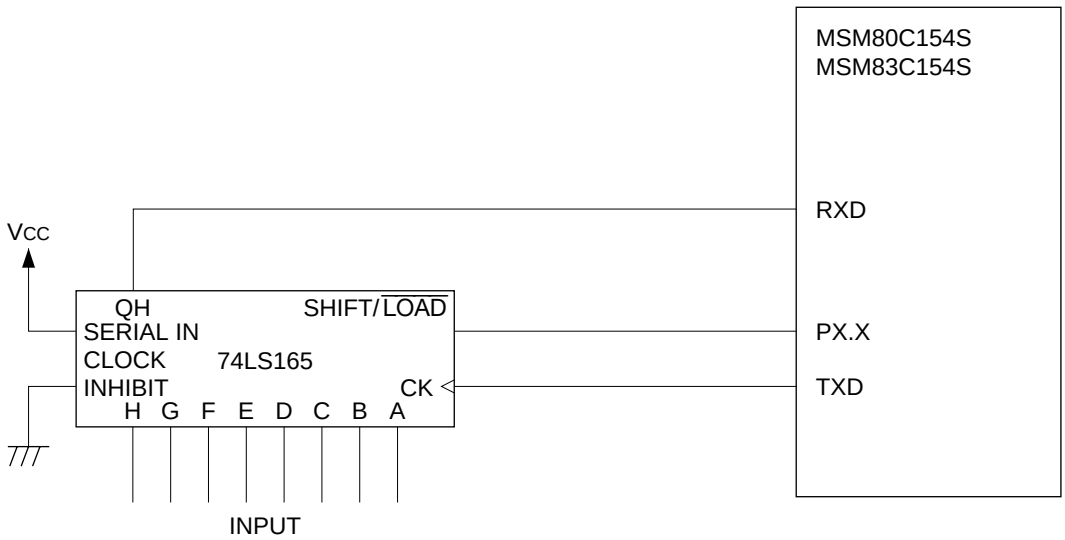


Figure 4-37 Input extension example

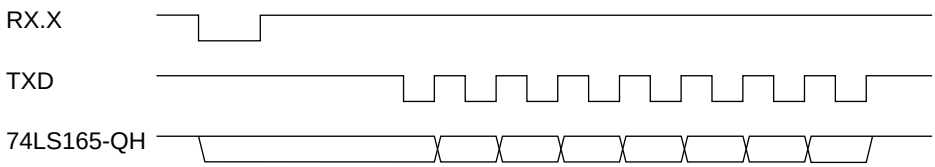


Figure 4-38 Input extension example timing chart

An output extension example is shown in Figure 4-39 and the corresponding timing chart is shown in Figure 4-40. After output data has been written into SBUF and the output sequence completed, the latch pulse output from PX.X is obtained and the 74LS164 data is shifted to 74LS373.

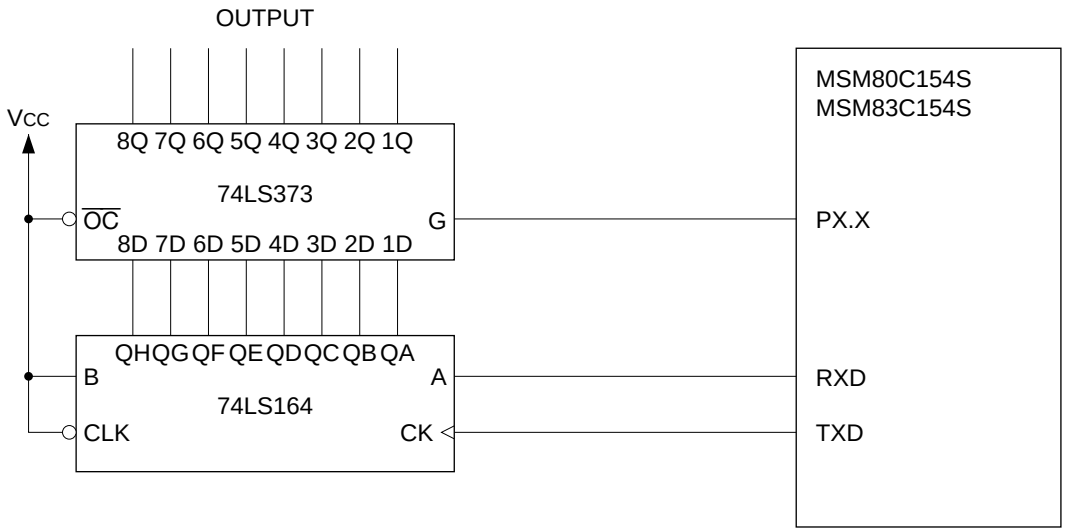


Figure 4-39 Output extension example

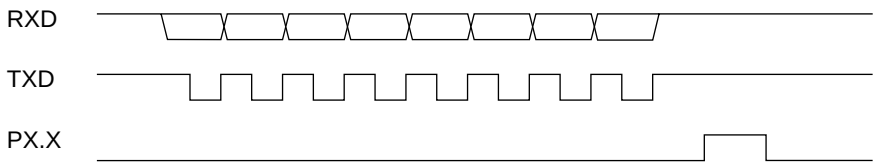


Figure 4-40 Output extension example timing chart

An input/output extension example is shown in Figure 4-41 and the corresponding timing chart is shown in Figure 4-42. When input data is applied, INPUT CONTROL is changed from “0” to “1”, and the parallel input is latched. This is then followed by REN=1 and RI=0 settings, and shift in of 74LS165 data. INPUT CONTROL is returned to “0” after the input has been completed. Since INPUT CONTROL is connected to the 74LS126 control pin, the MSM80C154S/MSM83C154S switches the 74LS126 output to high impedance when 74LS165 input data is not being applied, thereby preventing collision between the 74LS126 and MSM80C154S/MSM83C154S outputs.
When output data is generated, and the output is completed after writing output data into SBUF, an output latch pulse is generated from OUTPUT CONTROL, and the 74LS164 data is transferred to 74LS373. Although the 74LS164 data is changed to parallel input data when 74LS165 data is passed to MSM80C154S/MSM83C154S, an output latch pulse is generated only when output data is obtained from MSM80C154S/MSM83C154S, thereby preserving the correct data in 74LS373.

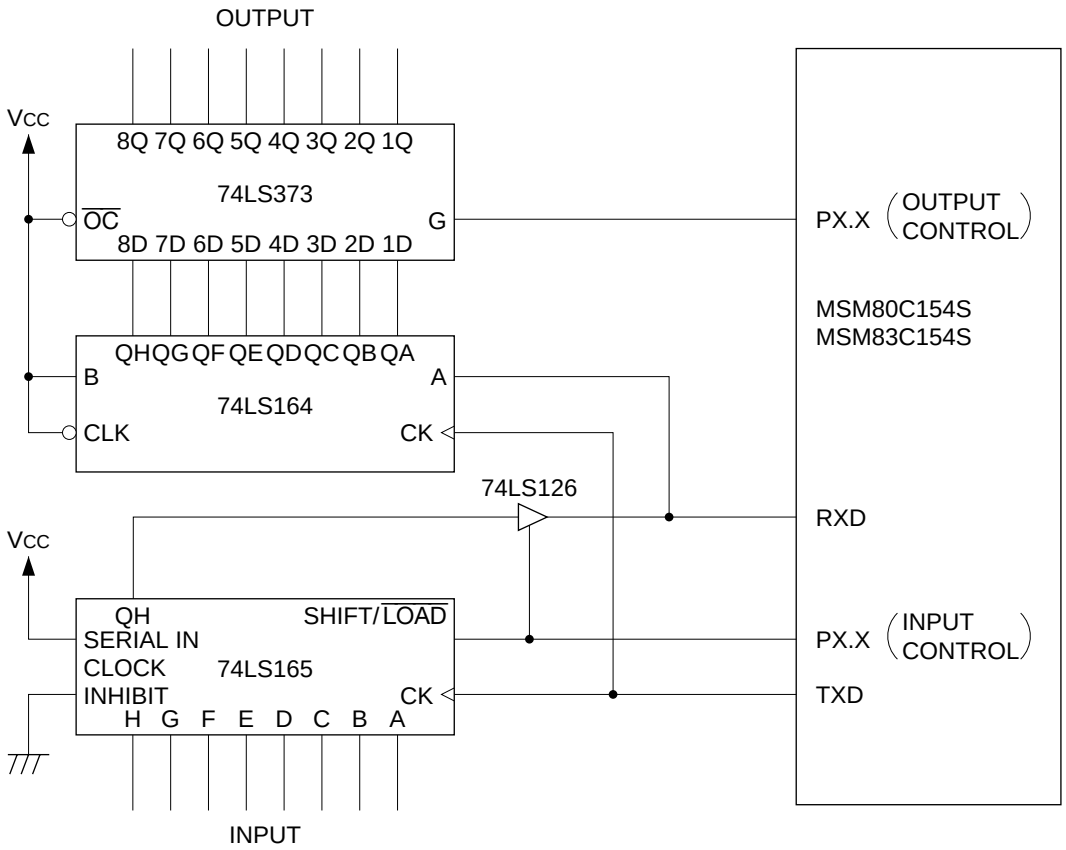
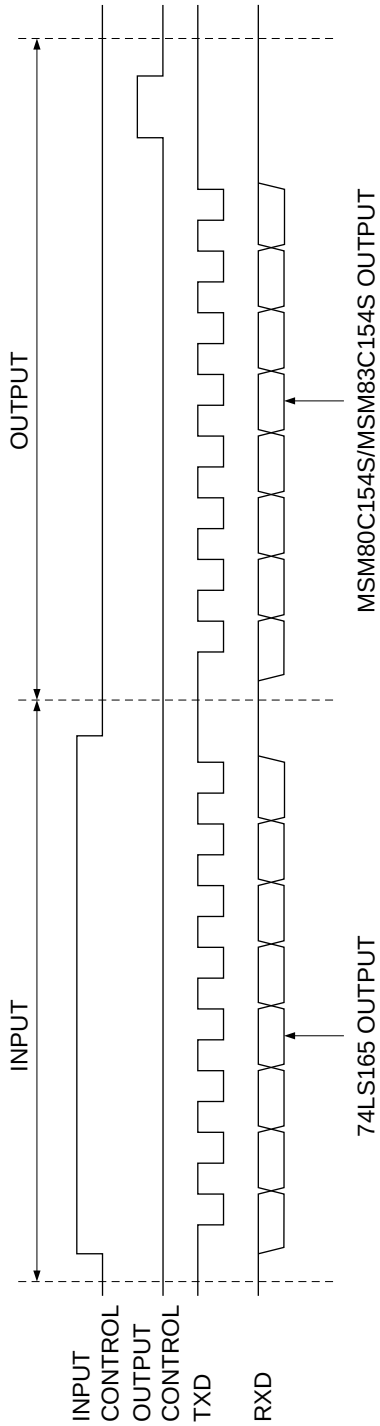


Figure 4-41 Input/output extension example



In all examples, additional multiple bit I/O extension is made possible by multiple cascade connections of 74LS164 or 74LS165.

Figure 4-42 Input/output extension example timing chart

4.6.4.2 Multi-processor systems

Multi-processor systems can be formed with MSM80C154S/MSM83C154S by using the serial port in mode 2 or mode 3 for inter-processor communications.
If reception data bit 9 (multi-purpose data bit) is “1” when SM2 is set in mode 2 or 3, reception data is received and an interrupt is generated. If the data bit is “0”, however, the reception data is disregarded and no interrupt is generated. This function is used in forming a multi-processor system when more than one MSM80C154S/MSM83C154S device is coupled by serial bus. An example of a multi-processor system with one master processor and a number of slave processors is shown in Figure 4-43. In this example, data is transmitted only from master to slave processors. Operation proceeds in accordance with the following protocol.

(1) Set SM2=“1”. All slave processors wait in standby for address data from the master processor specifying which slave is to be selected.
(2) With TB8 set to “1” to distinguish address data from other data, the master processor generates address data which ensures that data bit 9 (the multi-purpose data bit) is “1”.
(3) At this stage, all slave processors generate interrupts and check whether the received address data has specified itself or not.
(4) The specified slave processor sets SM2 “0” to prepare for reception of the subsequent data to be sent by the master processor.
Slave processors which are not specified remain at SM2=“1”
(5) With TB8=“0”, the master processor next sends data which ensures that data bit 9 (the multi-purpose data bit) is “0” following the address data.
(6) Since the specified slave processor is changed to SM2=“0”, all data following the address data is received and processed.
(7) The slave processors which are not specified (that is, where SM2=“1”) disregard all data after the address data and wait in standby for the next address data.
(8) After transmitting all of the intended data the master processor transmits a final special code (predetermined in advance).
(9) When this special code is received by the specified slave processor, SM2 is set to “1” and that slave processor is again put into standby waiting for address data.

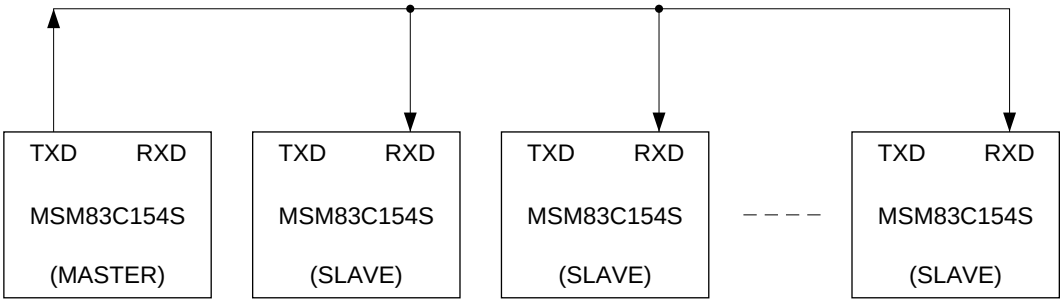


Figure 4-43 Multi-processor system example

4.7 Interrupt

4.7.1 Outline

MSM80C154S/MSM83C154S is equipped with six interrupts.

1. $\overline{\text{INT}}$0 External interrupt 0
2. TM0 Timer interrupt 0
3. $\overline{\text{INT}}$1 External interrupt 1
4. TM1 Timer interrupt 1
5. SI/O Serial port interrupt
6. TM2 Timer interrupt 2

These six interrupts are controlled by interrupt enable register (IE) and interrupt priority register (IP). When the relevant interrupt conditions are met, the respective interrupt address is called and the interrupt routine is commenced.

The interrupt addresses are listed in Table 4-18, and the interrupt control equivalent circuit is shown in Figure 4-44.

Table 4-18 Interrupt addresses

	Interrupt source	Interruput address
1	External interruput 0	3[0003hH]
2	Timer interruput 0	11[000BhH]
3	External interruput 1	19[0013hH]
4	Timer interruput 1	27[001BhH]
5	Serial port interruput	35[0023hH]
6	Timer interruput 2	43[002BhH]

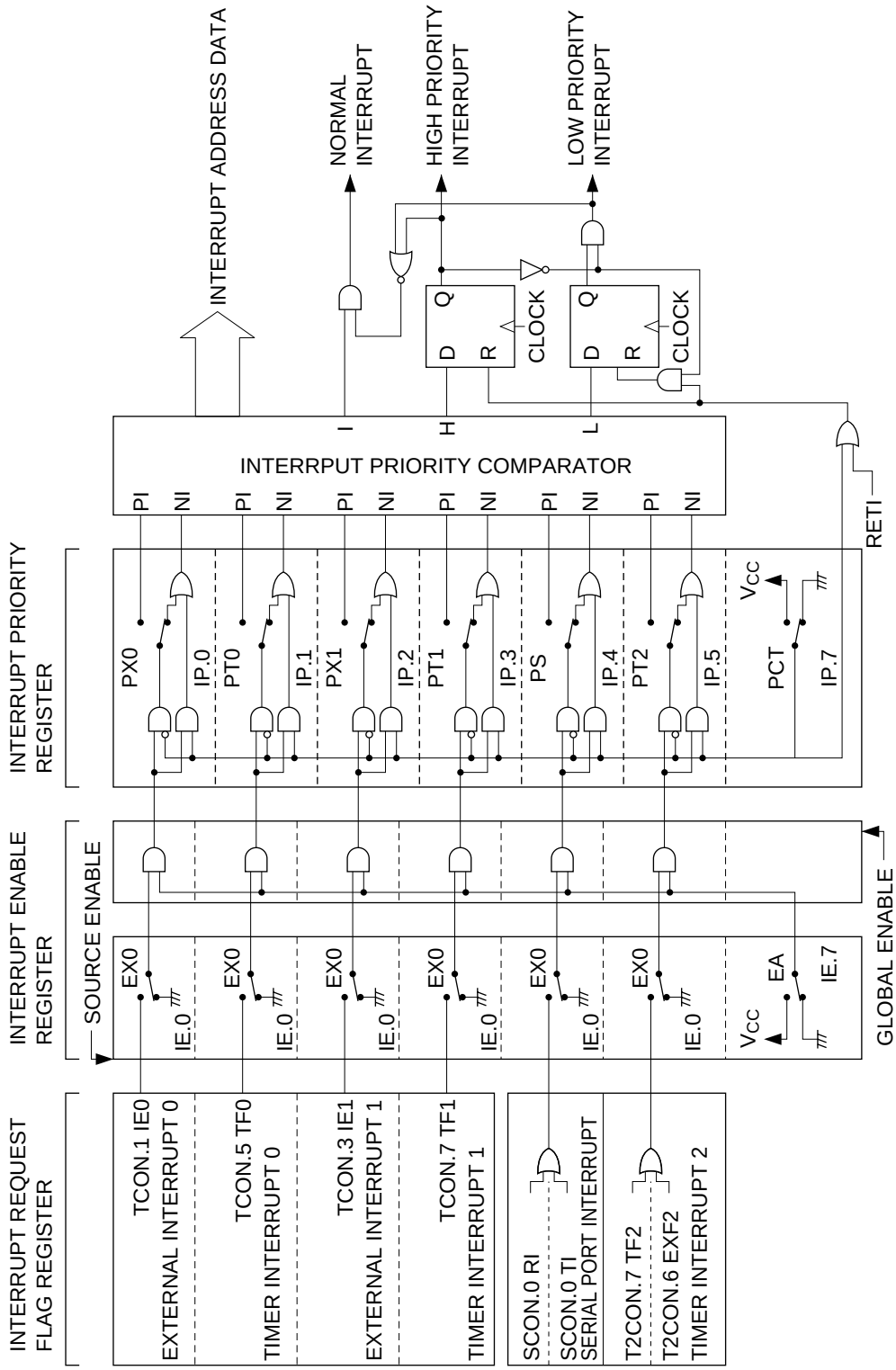


Figure 4-44 Interrupt control equivalent circuit

4.7.2 Interrupt enable register (IE)

The function of the interrupt enable register (IE, 0A8H) is to enable or disable interrupt processes when an interrupt is requested.
To execute the intended interrupt routine, the interrupt is first enabled by setting “1” in the corresponding interrupt bit in the interrupt enable register, and the routine then is executed when the interrupt is requested.
Requested interrupts are disabled if the corresponding interrupt bit is “0”, and no interrupt routines are executed.
The contents of the interrupt enable register (IE) are shown in Table 4-19.

Table 4-19 Interrupt enable register (IE, 0A8H)

Bit	7	6	5	4	3	2	1	0
Flag	EA	—	ET2	ES	ET1	EX1	ET0	EX0

EX0 : External interrupt 0 control bit
Interrupt enabled when “1”, disabled when “0”.
ET0 : Timer interrupt 0 control bit
Interrupt enabled when “1”, disabled when “0”.
EX1 : External interrupt 1 control bit
Interrupt enabled when “1”, disabled when “0”.
ET1 : Timer interrupt 1 control bit
Interrupt enabled when “1”, disabled when “0”.
ES : Serial port interrupt control bit
Interrupt enabled when “1”, disabled when “0”.
ET2 : Timer interrupt 2 control bit
Interrupt enabled when “1”, disabled when “0”.
— : Reserve bit for output of “1” when read.
EA : Interrupt control bit for all six interrupts (EX0, ET0, EX1, ET1, ES, and ET2) When EA is “1”, an interrupt routine is commenced if interrupt conditions are met for any one of the six interrupts.
When EA is “0”, no interrupt routine is commenced even if interrupt conditions are met for one of the six interrupts.

4.7.3 Interrupt priority register (IP)

The function of the interrupt priority register (IP, 0B8H) is to allocate rights to commence interrupt routines on a priority basis when an interrupt is requested.
Interrupt priority can be programmed by setting the bit corresponding to the interrupt request in the interrupt priority register (IP) to “1”. If the interrupt conditions have been satisfied for an interrupt where “1” data has been set, processing of that interrupt is commenced. If another interrupt (with “0” priority bit) is already being processed, that routine is suspended, and processing of the higher priority interrupt is commenced. Note that once a priority interrupt routine has been commenced, processing of the next interrupt cannot start until processing of the current interrupt has been completed.
This priority circuit function can be stopped by setting “1” in bit 7 (PCT) of the priority register. The functions of the priority interrupt control circuit are suspended, and interrupt control is handled only by the interrupt enable register (IE 0A8H). After this mode has been set, the interrupt disable instruction (CLR EA) must be placed at the beginning of interrupt routines to disable the generation of other interrupts.
If another interrupt routine have to be generated during the processing of an interrupt routine, set the desired interrupt enable bit in the interrupt enable register (IE 0A8H). The desired interrupt routine is processed when the conditions for that routine are met. Multi-level interrupt processing can thus be achieved by software control of the interrupt enable register.
The contents of the interrupt priority register are given in Table 4-20, and a priority interrupt routine flow chart is shown in Figure 4-45. The flow chart for an interrupt routine when the priority circuit is stopped (PCT=“1”) is shown in Figure 4-46.

Table 4-20 nterrupt priority register (IP, 0B8H)

Bit	7	6	5	4	3	2	1	0
Flag	PCT	—	PT2	PS	PT1	PX1	PT0	PX0

PX0 : External interrupt 0 priority bit.
Priority is allocated when this bit is “1”.
PT0 : Timer interrupt 0 priority bit.
Priority is allocated when this bit is “1”.
PX1 : External interrupt 1 priority bit. Priority is allocated when this bit is “1”.
PT1 Timer interrupt 1 priority bit.
Priority is allocated when this bit is “1”.
PS : Serial port interrupt priority bit
Priority is allocated when this bit is “1”.
PT2 : Timer interrupt 2 priority bit.
Priority is allocated when this bit is “1”.
— : Reserve bit for output of “1” when read.
PCT : Priority interrupt circuit control bit.
The priority interrupt control circuit is activated when this bit is “0”, and an interrupt is processed on the priority basis (2 level interrupt processing).
The priority interrupt control circuit is stopped when this bit is “1”. In this case, all interrupts are controlled by the interrupt enable register (IE) where multi-level interrupt processing is possible by software management.

4.7.3.1 Priority interrupt routine flow

The flow of interrupt processing when a priority interrupt is generated and processed after a routine has been commenced by a non-priority interrupt generated during execution of a main routine program is outlined in Figure 4-45 below. This diagram shows the flow chart up to the point of return to the main routine.

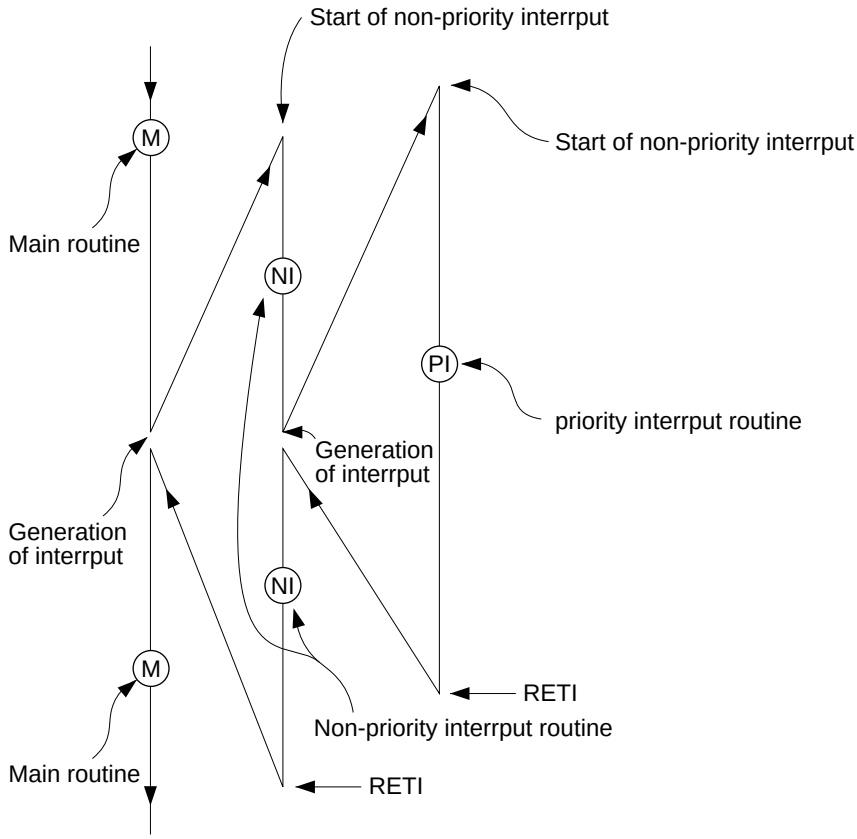


Figure 4-45 Interrupt processing flow chart when priority circuit is activated

4.7.3.2 Interrupt routine flow when priority circuit is stopped

When bit 7 (PCT) of the priority register (IP 0B8H) is set to "1", all interrupt control is transferred to the interrupt enable register (IE 0A8H). When this mode is set, the interrupt disable instruction (CLR EA) must always be placed at the beginning of the interrupt routine to prevent any other interrupt from being generated. If another interrupt routine have to be generated during the processing of an interrupt routine, set the desired interrupt enable bit in the interrupt enable register (IE 0A8H) to commence the new interrupt routine. Multi-level interrupt processing can thus be achieved by control of the interrupt enable register. The flow of this interrupt routine is shown in Figure 4-46.

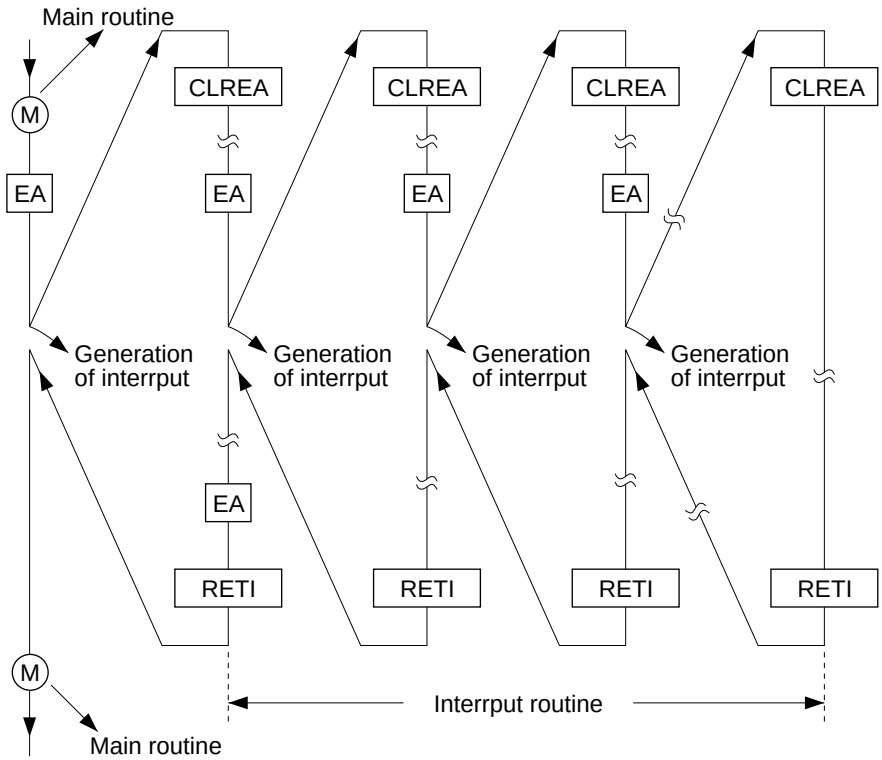


Figure 4-46 Interrupt routine flow chart when priority circuit is stopped

4.7.3.3 Interrupt priority when priority register (IP) contents are all “0”

The interrupt priority when the priority register (IP, 0B8H) contents are all “0” indicates the priority in which a certain interrupt is processed in preference to other interrupts when interrupt requests are generated simultaneously.
As can be seen from Table 4-21, the interrupt to be processed in preference to all other interrupts is external interrupt 0, and the interrupt routine with lowest priority is timer interrupt 2.
The interrupt level when all priority bits are “0” is 1 level, and even if the interrupt conditions for an external interrupt 0 (highest priority) are satisfied while timer interrupt 2 (lowest priority) is being processed, the external interrupt cannot be processed.
The same operational preferences as described above also exist when all priority bits are “ 1”.

Table 4-21 Non-priority interrupt order of preference

Order of preference	Interrupt source
1	External interruput 0
2	Timer interruput 0
3	External interruput 1
4	Timer interruput 1
5	Serial port interruput
6	Timer interruput 2

4.7.4 Detection of external interrupt signals $\overline{INT}$0 and $\overline{INT}$1

4.7.4.1 Outline of $\overline{INT}$ signal detection

Detect modes of the external interrupt signals 0 and 1 can be set to level-detect or trigger-detect mode by the IT0 and IT1 data values in the timer control register (TCON 88H) as indicated in Table 4-22.

Table 4-22 TCON[88H] register

	Timer				$\overline{INT}$1		$\overline{INT}$0	
Bit	7	6	5	4	3	2	1	0
Flag	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Set						•		•

4.7.4.2 External interrupt signal 0 and 1 level detection

When bit 0 (IT0) in the timer control register (TCON 88H) is “0”, external interrupt 0 is level-activated. And when bit 2 (IT1) is “0”, external interrupt 1 is also level-activated. With the external interrupt signals in level-detect mode, external interrupts 0 and 1 are level-detected by the equivalent circuit shown in Figure 4-47.
When the level of the external interrupt pin is “0” at S5 timing, the level is latched and the Q output becomes “1”. The latched external interrupt signal is set as the external interrupt flag in the timer control register (TCON) at S3 timing. The interrupt flag set by external interrupt signal is always reset at S6 timing of the end of the machine cycle, thereby executing the equivalent of a “level sense” operation. The cycle width of the respective “0” and “1” levels of the external interrupt signal applied to the external interrupt pin in this case must be at least 12 times (12T) the XTAL1·2 oscillator clock cycle time T.
And the external interrupt signal should be held at “0” level until the corresponding interrupt is actually generated.

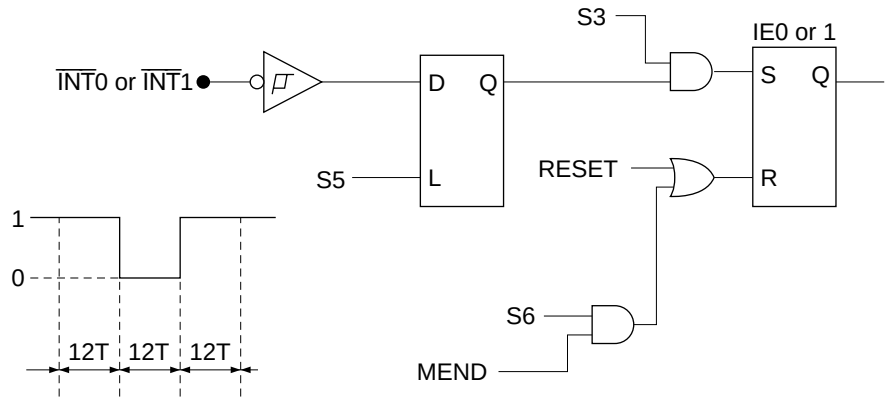


Figure 4-47 Interrupt level detect equivalent circuit for IT bit “0”

4.7.4.3 External interrupt signal 0 and 1 trigger detection

When bit 0 (IT0) in the timer Control register (TCON 88H) is “1”, external interrupt 0 is edge-activated. And when bit 2 (IT1) is “1”, external interrupt 1 is also edge-activated. With the external interrupt signals in trigger-detect mode, external interrupts 0 and 1 are trigger-detected by the equivalent circuit shown in Figure 4-48. When the level of the external interrupt pin is “0” at S5 timing, the level is latched at the first stage and the latched Q output becomes “1”. The external interrupt signal stored in the first stage latch is transferred to the second stage latch and is subject to digital differentiation until the S3 timing signal. The RS-F/F in the next stage is set by the differentiated output signal.
The external interrupt signal applied to the RS-F/F is synchronized with the $\overline{M2}\cdot S3$ timing signal to be applied as a trigger for the external interrupt flag in the timer control register (TCON). The RS-F/F is subsequently reset at $\overline{M2}\cdot S4$ and waits for the next interrupt. Note that the next interrupt signal is invalid until the first stage latch detects level “1” after detecting level “0”.
The cycle width of the respective “0” and “1” levels of the external interrupt signal applied to the external interrupt pin in this case must be at least 12 times (12T) the XTAL1·2 oscillator clock cycle time T.

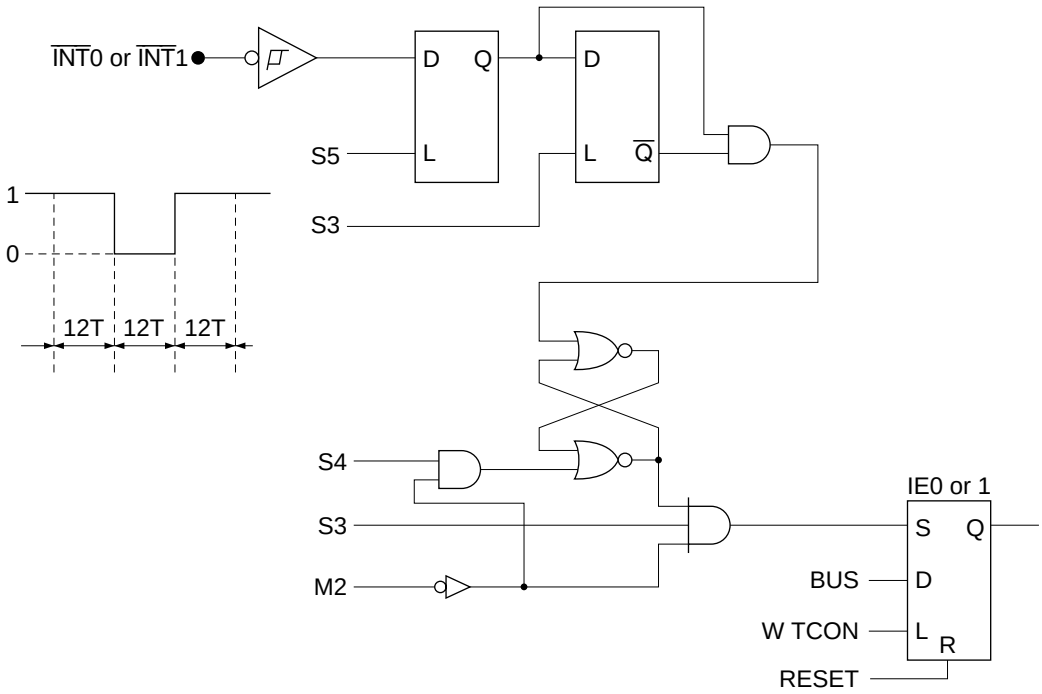


Figure 4-48 Interrupt edge detect equivalent circuit for IT bit “1”

4.7.5 MSM80C154S/MSM83C154S interrupt response time charts

4.7.5.1 Interrupt response time chart when interrupt conditions are satisfied during execution of ordinary instructions in main routine

If interrupt conditions are satisfied during execution of an ordinary instruction (which does not manipulate IE or IP) in the main routine, the MSM80C154S/MSM83C154S calls the interrupt address in the next cycle following completion of the ordinary instruction. The time chart is given in Figure 4-49.

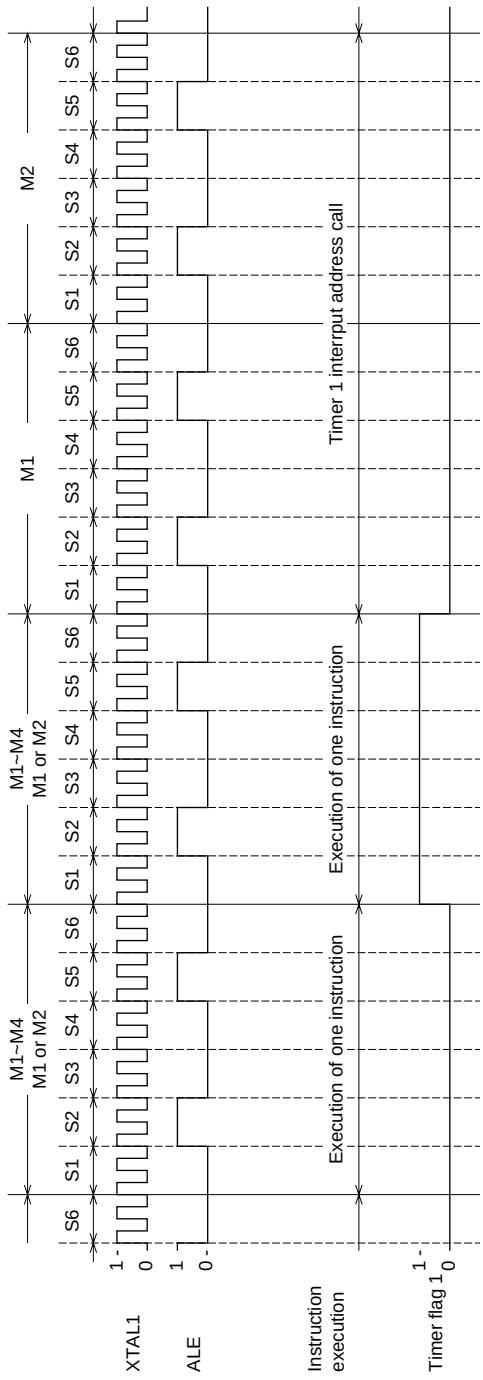


Figure 4-49 Interrupt response time chart when interrupt conditions are satisfied during execution of ordinary instruction in main routine

4.7.5.2 Interrupt response time chart when interrupt conditions are satisfied during execution of IE or IP register operation instruction in main routine

If interrupt conditions are satisfied during execution of an instruction used to manipulate the interrupt enable register (IE) or the interrupt priority register (IP) in the main routine, the MSM80C154S/MSM83C154S reactivates the interrupt mask circuit in the next cycle following completion of the register manipulation instruction. If interrupt conditions were met as a result of the re-interrupt mask, the interrupt address is called in the next cycle. That is, if the interrupt conditions are satisfied during execution of the IE or the IP manipulating instruction, the interrupt address is called after the next instruction is executed following execution of the register manipulating instruction. The time chart is given in Figure 4-50.

* In the MOV data address 1, data address 2 instructions, transfer of data to another register from IE or IP is an exception. (example: MOV ACC, IE)

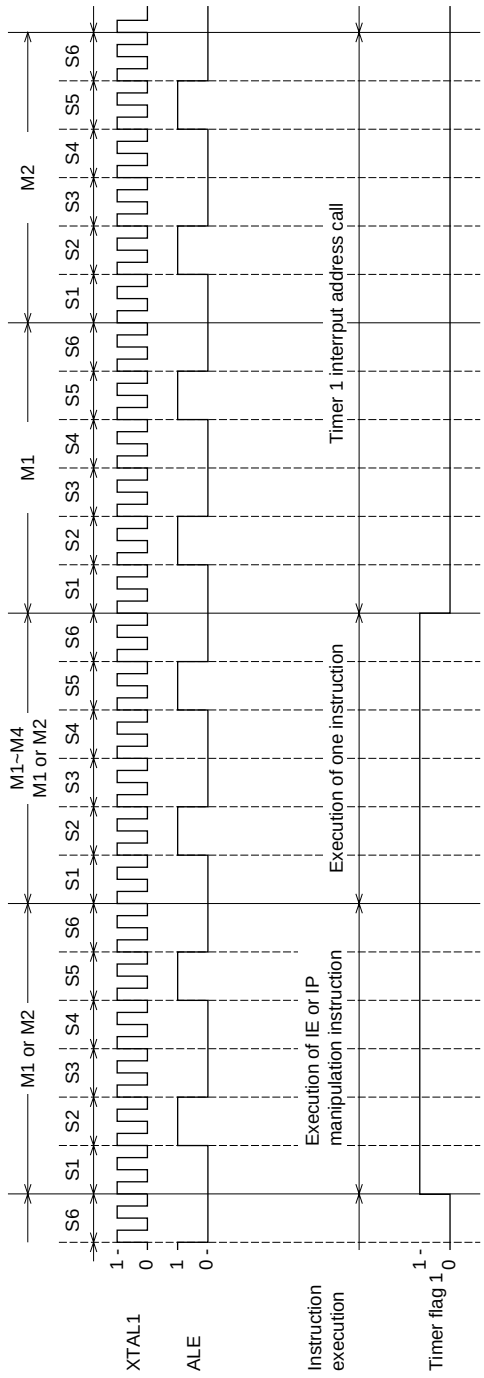


Figure 4-50 Interrupt response time chart when interrupt conditions are satisfied during execution of IE or IP register manipulating instruction in main routine

4.7.5.3 Interrupt response time chart when an ordinary instruction is executed after temporarily returning to the main routine from continuous interrupt processing

If an ordinary instruction (which does not manipulate IE or IP) is executed after returning to the main routine following execution of the interrupt routine end instruction RETI, and if the next interrupt conditions have been met during execution of a previous interrupt routine, the MSM80C154S/MSM83C154S calls the interrupt address in the next cycle following execution of one main routine instruction. The same occurs when interrupt conditions are satisfied during execution of the first main routine instruction after returning to the main routine from the interrupt routine. The time chart is shown in Figure 4-51.

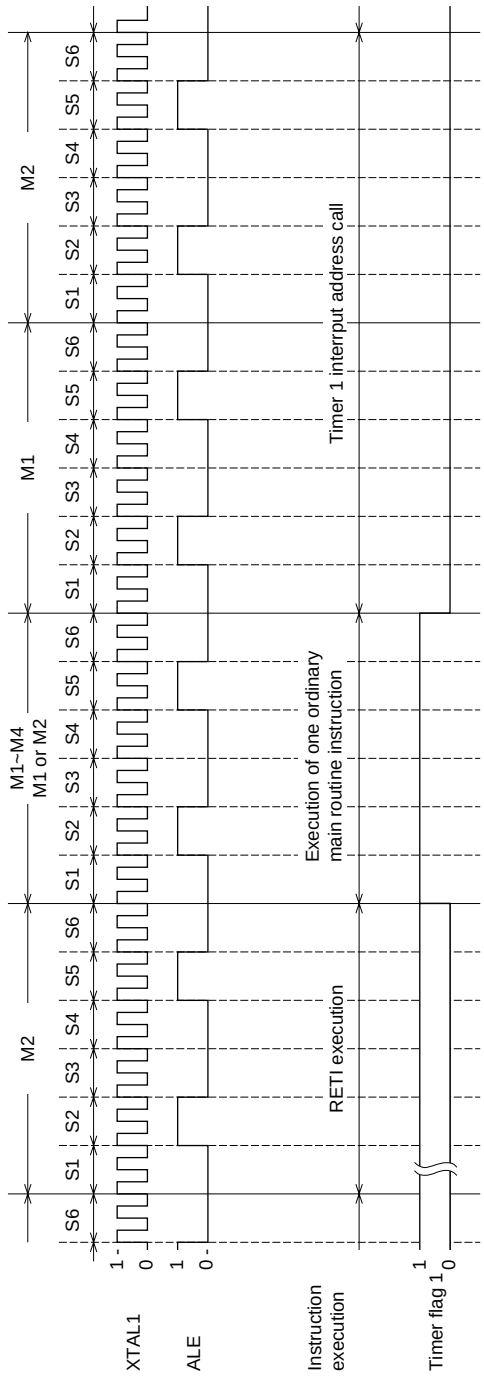


Figure 4-51 Interrupt response time chart when ordinary instruction is executed after returning to main routine during continuous interrupt processing

4.7.5.4 Interrupt response time chart when an IE or IP manipulating instruction is executed after temporarily returning to the main routine from continuous interrupt processing

If the next interrupt conditions are satisfied during execution of an interrupt processing routine and the interrupt terminating instruction RETI is then executed and followed by a return to the main routine where an instruction which manipulates the interrupt enable register (IE) or interrupt priority register (IP) is executed, the MSM80C154S/MSM83C154S activates the interrupt mask circuit in the next cycle following execution of the register manipulating instruction. And if interrupt conditions are met as a result of the re-interrupt mask, the interrupt address is called in the next cycle. That is, if the instruction executed in the main routine manipulates either IE or IP, the interrupt address is called after two instructions are executed. The time chart is shown in Figure 4-52.

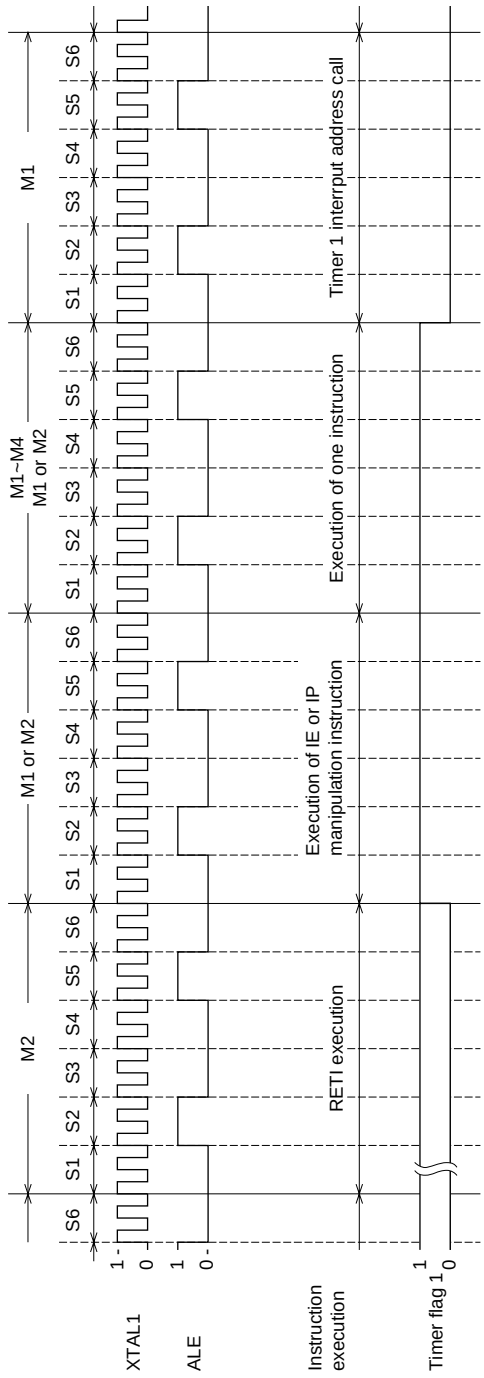


Figure 4-52 Interrupt response time chart when IE or IP manipulating instruction is executed after returning to main routine during continuous interrupt processing

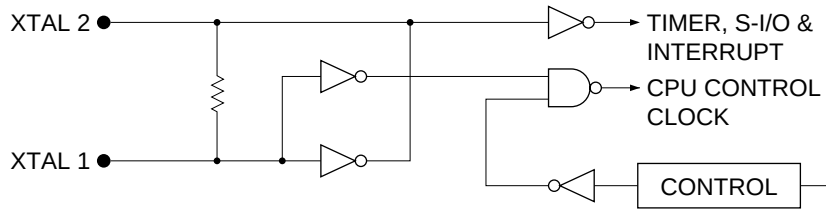
4.8 CPU "Power Down"

4.8.1 Outline

Since the internal MSM80C154S/MSM83C154S circuits have been designed as completely static circuits, all internal information (register data) is preserved if XTAL1·2 oscillation is stopped.
This feature is utilized to incorporate a fuller range of power down modes.
In idle mode (IDLE) where "1" is set in bit 0 (IDL) of the power control register (PCON), XTAL1·2 operation is continued but CPU operations are stopped. In soft power down mode where "1" is set in bit 1 (PD) of the power control register (PCON), XTAL1·2 operation and CPU operations are both stopped.
And in hard power down mode where "1" is set in advance in bit 6 (HPD) of the power control register (PCON), XTAL1·2 and CPU operations are stopped when the level of the power failure detect signal applied to the HPDI pin (P3.5) is changed from "1" to "0".
If "1" is set in bit 0 (ALF) of the I/O control register (IOCON 0F8H) prior to activation of soft and hard power down modes where CPU and XTAL1·2 operations are stopped, the port 0, 1, 2, and 3 outputs can be floated.
CPU power down modes can be released (CPU start-up) by CPU resetting, interrupt generation, and interrupt source signal generation.
Execution can be recommenced from address 0, resumed from the interrupt address or from the next address after the power down setting instruction.

4.8.2 Idle mode (IDLE) setting

Idle mode is set when "1" is set in bit 0 (IDL) of the power control register (PCON 87H). The circuit connections involved in this setting are shown in Figure 4-53.
The idle mode cancellation conditions can be set through manipulation of bit 5 (RPD) of the power control register. When "0" is set in RPD, idle mode cannot be cancelled by the interrupt signal if the corresponding interrupt enable bit has not been set. And if "1" is set in RPD, idle mode is cancelled by setting the interrupt flag and the program is executed from the next address of the idle mode setting instruction, even when the corresponding interrupt enable bit is not set.
In idle mode, the supply of clocks to the CPU control section is stopped and CPU operations are halted. But since XTAL1·2 operations are maintained, the serial port, interrupt circuits, and timer/counters 0, 1, and 2 remain operative.
The CPU pin status during idle mode is outlined in Table 4-23, and the corresponding time charts for starting idle mode are shown in Figures 4-54 and 4-55.



PCON, 87H

	SMOD	HPD	RPD	—	GF1	GF0	PD	IDL
Bit	7	6	5	4	3	2	1	0
Set			*					•

Figure 4-53 Idle mode equivalent circuit

Table 4-23 CPU pin details in idle mode

Name	Internal ROM	External ROM
P1.0/T2	Port data output	Port data output
P1.1/T2EX	Port data output	Port data output
P1.2	Port data output	Port data output
P1.3	Port data output	Port data output
P1.4	Port data output	Port data output
P1.5	Port data output	Port data output
P1.6	Port data output	Port data output
P1.7	Port data output	Port data output
RESET	“0” level input	“0” level input
P3.0/RXD	Port data output	Port data output
P3.1/TXD	Port data output	Port data output
P3.2/$\overline{\text{INT}}$0	Port data output	Port data output
P3.3/$\overline{\text{INT}}$1	Port data output	Port data output
P3.4/ T0	Port data output	Port data output
P3.5/ T1/HPDI	Port data output	Port data output
P3.6/$\overline{\text{WR}}$	Port data output	Port data output
P3.7/$\overline{\text{RD}}$	Port data output	Port data output
XTAL 2	Oscillator operative	Oscillator operative
XTAL 1	Oscillator operative	Oscillator operative
V_{SS}	0 [V]	0 [V]
P2.0	Port data output	Address 8 output
P2.1	Port data output	Address 9 output
P2.2	Port data output	Address 10 output
P2.3	Port data output	Address 11 output
P2.4	Port data output	Address 12 output
P2.5	Port data output	Address 13 output
P2.6	Port data output	Address 14 output
P2.7	Port data output	Address 15 output
$\overline{\text{PSEN}}$	“1” level output	“1” level output
ALE	“1” level output	“1” level output
$\overline{\text{EA}}$	“1” level input	“0” level input
P0.7	Port data output	Floating
P0.6	Port data output	Floating
P0.5	Port data output	Floating
P0.4	Port data output	Floating
P0.3	Port data output	Floating
P0.2	Port data output	Floating
P0.1	Port data output	Floating
P0.0	Port data output	Floating
V_{CC}	+2.2~+6 [V]	+2.2~+6 [V]

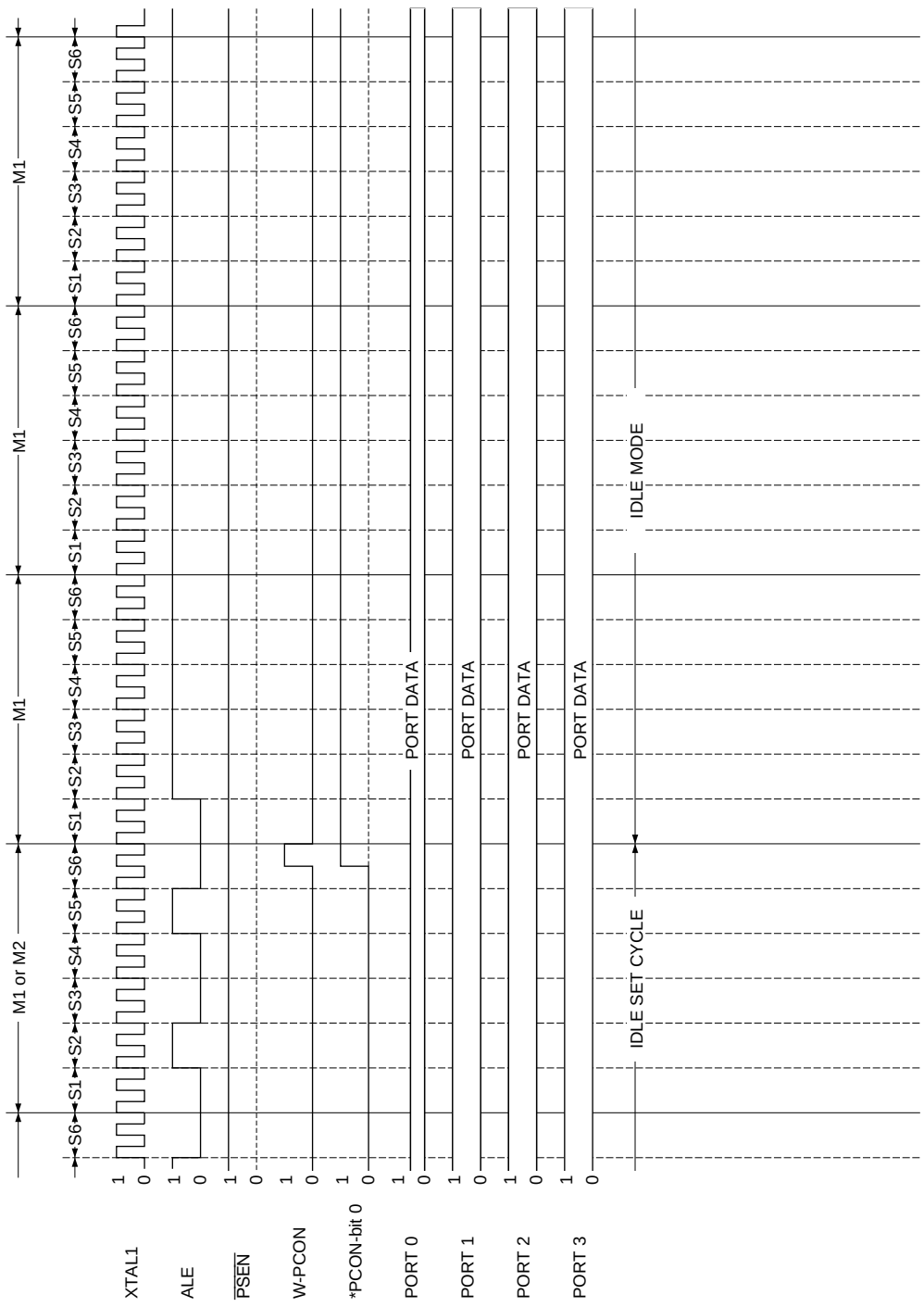


Figure 4-54 Idle mode setting time chart (internal ROM mode)

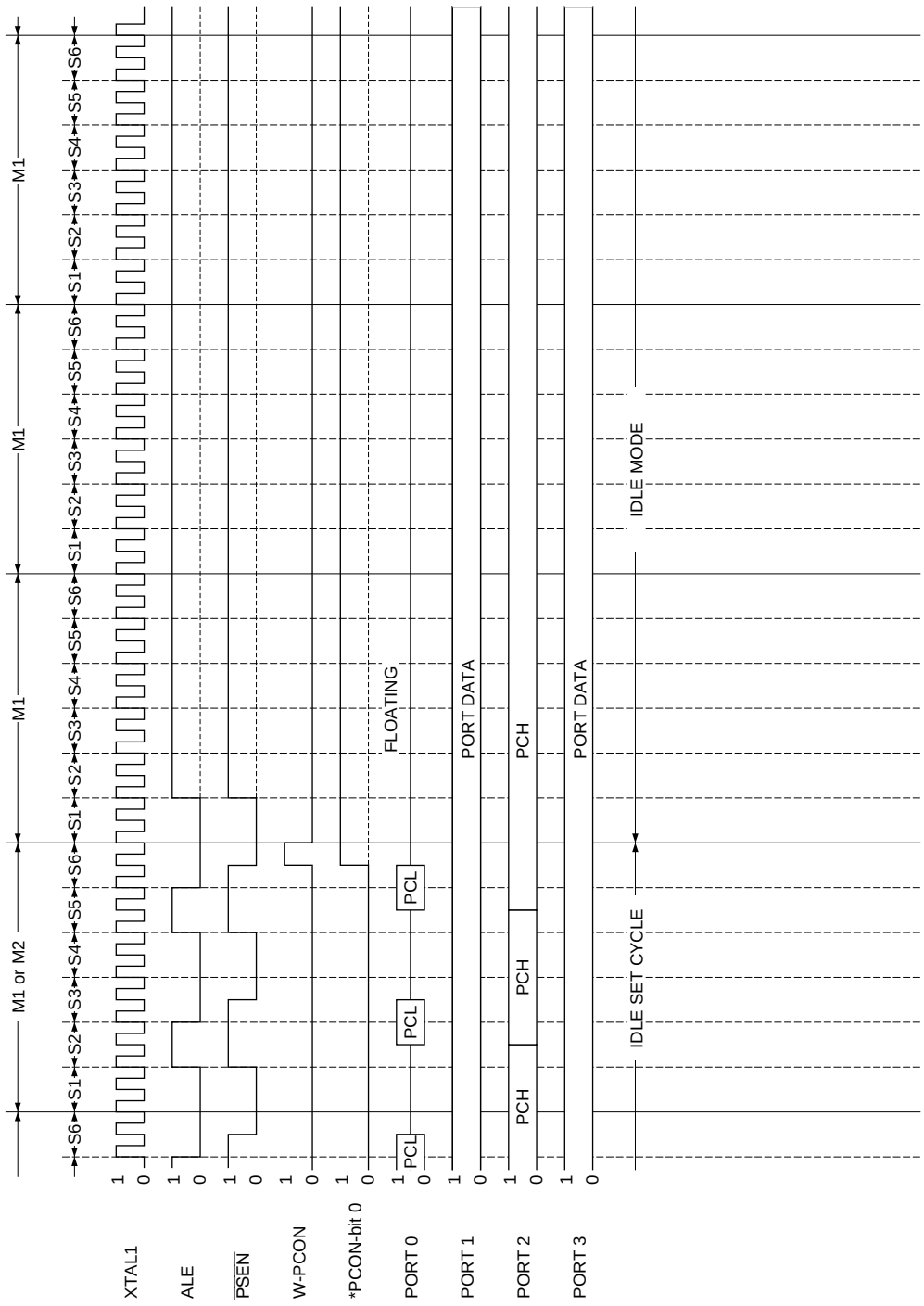


Figure 4-55 Idle mode setting time chart (external ROM mode)

4.8.3 Soft power down mode (PD) setting

Soft power down mode (PD) is set when “1” is set in bit 1 (PD) of the power control register (PCON 87H). The circuit connection involved in this setting is shown in Figure 4-56.
Soft power down mode cancellation conditions can be set through manipulation of bit 5 (RPD) of the power control register.
When “0” is set in RPD, soft power down mode cannot be cancelled by the interrupt signal if the corresponding interrupt enable bit has not been set. And if “1” is set in RPD, the power down mode is cancelled by setting the interrupt flag and the program is executed from the next address of the soft power down mode setting instruction, even when the corresponding interrupt enable bit is not set. In soft power down mode, XTAL1·2 operations are halted. Then with all internal data preserved, all CPU operations are stopped apart from timer/counters 0 and 1.
(Timer/counters 0 and 1 operate in external clock mode.)

Note, however, that the soft power down mode can not be set under the following conditions.

4.8.3.1 Caution about software power down mode setting

If the software power down mode can be cancelled by interruption and the following conditions are established, the software power down mode cannot be set.

(1) If trying to set the software power down mode under the conditions that the mode can be cancelled by external interrupt 0 or 1 and the $\overline{INT0}$ or $\overline{INT1}$ pin is set to "0" (either level input or edge input).

(2) If trying to set the software power down mode under the conditions that the mode can be cancelled by timer 0 or 1 (external clock mode is set) and the T0 or T1 pin is set to "1" when the value of the counter is "FF".

Figures 4-57, 4-58, and 4-59 show power down cancellation circuits by external interrupt or timer interrupt.Note, however, that the soft power down mode can not be set under the following conditions.

The pin output status of ports 0 thru 3 in soft power down mode can be left in port data output status, or set to port output floating status.
The ports are set to data output status by setting bit 0 (ALF) of the I/O control register (IOCON) to “0” when soft power down mode is activated, and to floating status by setting ALF to “1” before activating power down mode. In floating status, the port pins are disconnected electrically from the external circuitry. Apart from pins 2,3, 4, and 5 of port 3, all floating status input port pins may be open, or undefined within the –0.5 to V_{CC}+0.5V range.
The CPU pin status during soft power down mode (PD) with “0” on the ALF bit is /outlined in Table 424, and the corresponding time charts for starting soft power down mode are shown in Figures 4-60 and 4-61.
The CPU pin status during soft power down mode with “1” on the ALF bit is outlined in Table 4-25, and the corresponding time charts for starting soft power down mode are shown in Figures 4-62 and 4-63.

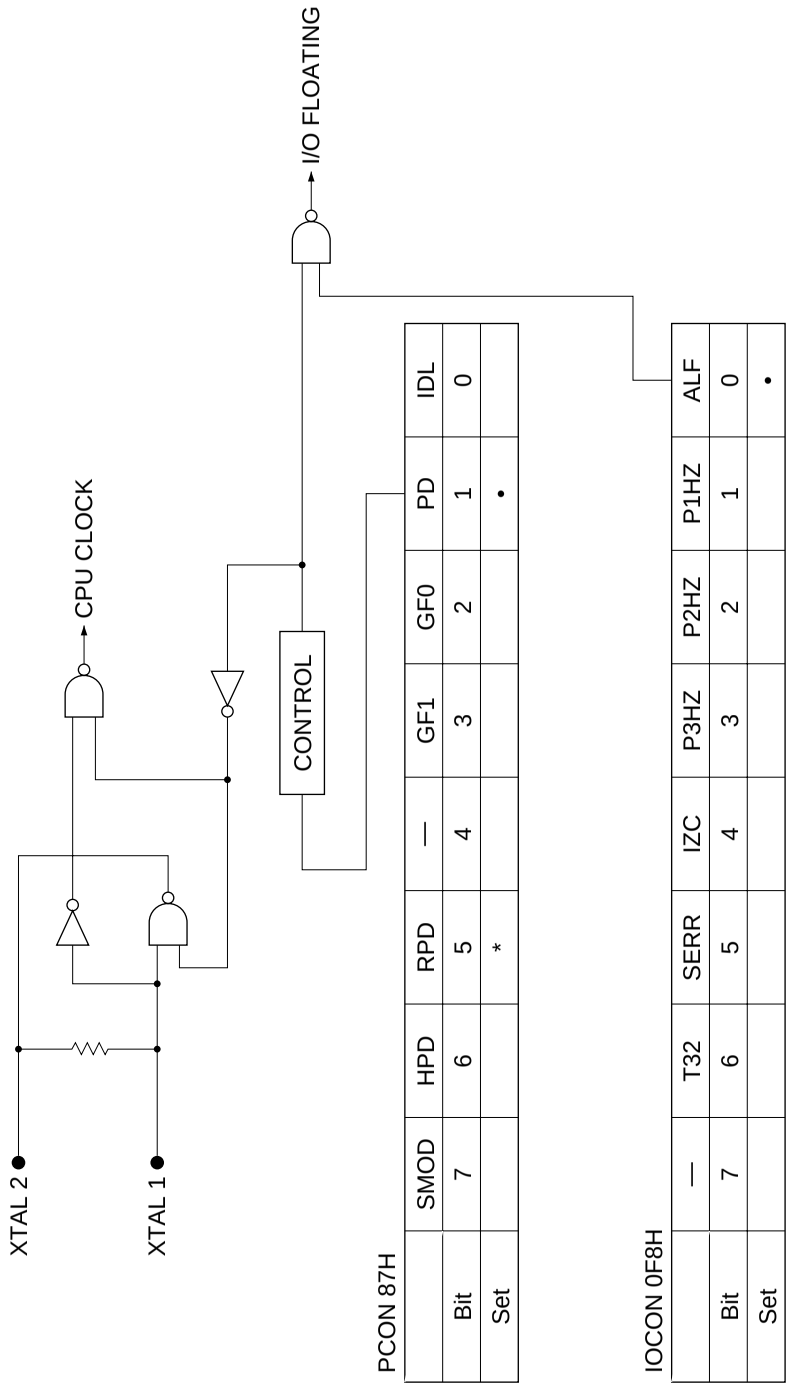


Figure 4-56 Soft power down mode equivalent circuit

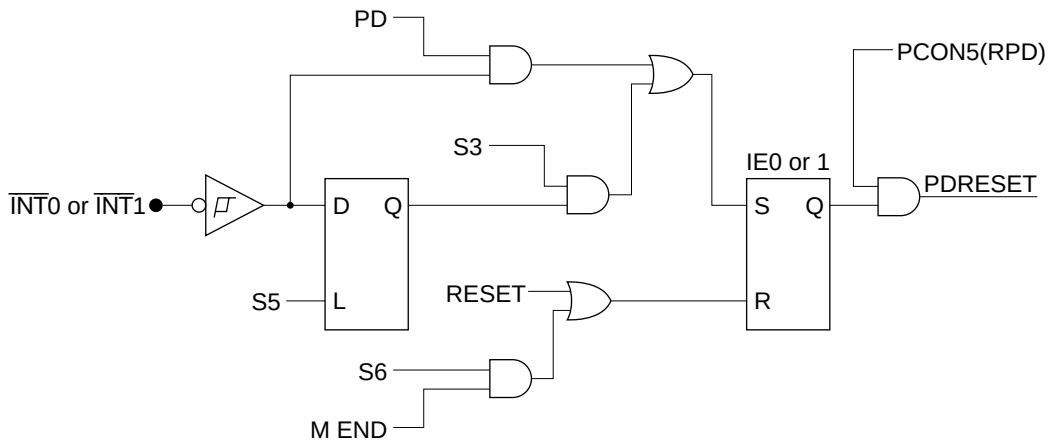


Figure 4-57 Power down cancellation circuit at INTERRUPT level input

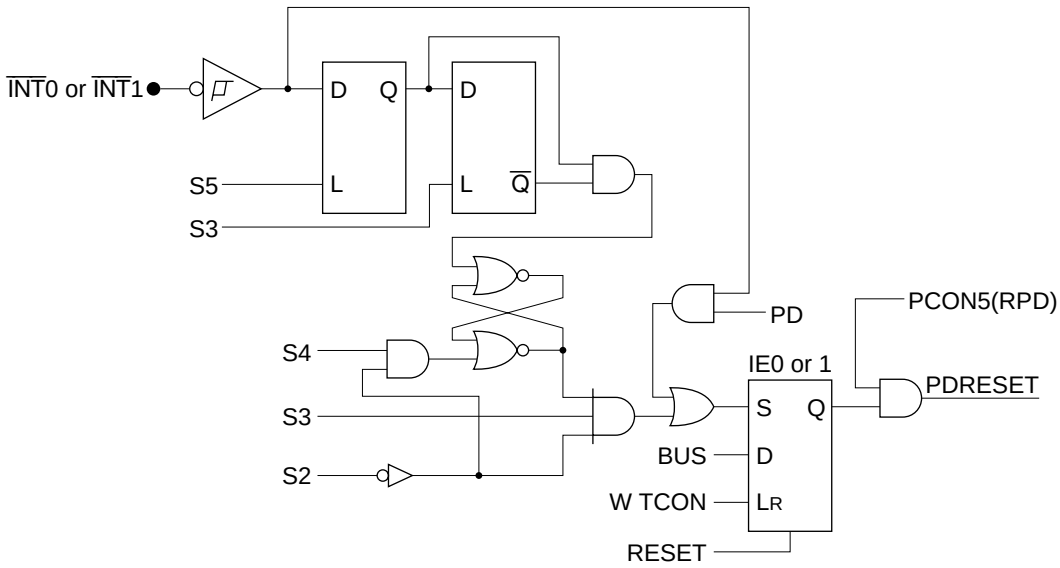


Figure 4-58 Power down cancellation circuit at INTERRUPT edge input

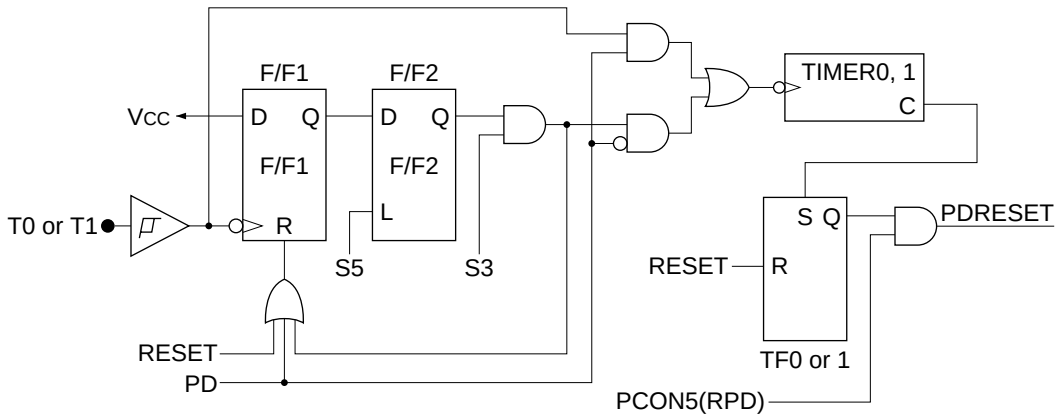


Figure 4-59 TIMER0, 1 power down cancellation circuit

Table 4-24 CPU pin details (ALF=0) in soft power down mode (PD)

Name	Internal ROM	External ROM
P1.0/T2	Port data output	Port data output
P1.1/T2EX	Port data output	Port data output
P1.2	Port data output	Port data output
P1.3	Port data output	Port data output
P1.4	Port data output	Port data output
P1.5	Port data output	Port data output
P1.6	Port data output	Port data output
P1.7	Port data output	Port data output
RESET	“0” level input	“0” level input
P3.0/RXD	Port data output	Port data output
P3.1/TXD	Port data output	Port data output
P3.2/$\overline{\text{INT}}$0	Port data output	Port data output
P3.3/$\overline{\text{INT}}$1	Port data output	Port data output
P3.4/ T0	Port data output	Port data output
P3.5/ T1/HPDI	Port data output	Port data output
P3.6/$\overline{\text{WR}}$	Port data output	Port data output
P3.7/$\overline{\text{RD}}$	Port data output	Port data output
XTAL 2	Oscillator operative	Oscillator operative
XTAL 1	Oscillator operative	Oscillator operative
V_{SS}	0 [V]	0 [V]
P2.0	Port data output	Port data output
P2.1	Port data output	Port data output
P2.2	Port data output	Port data output
P2.3	Port data output	Port data output
P2.4	Port data output	Port data output
P2.5	Port data output	Port data output
P2.6	Port data output	Port data output
P2.7	Port data output	Port data output
$\overline{\text{PSEN}}$	“0” level output	“0” level output
ALE	“0” level output	“0” level output
$\overline{\text{EA}}$	“1” level input	“0” level input
P0.7	Port data output	Floating
P0.6	Port data output	Floating
P0.5	Port data output	Floating
P0.4	Port data output	Floating
P0.3	Port data output	Floating
P0.2	Port data output	Floating
P0.1	Port data output	Floating
P0.0	Port data output	Floating
V_{CC}	*+2.0~+6 [V]	*+2.0~+6 [V]

* V_{CC}=+2.0~+6V when internal CPU data is held.

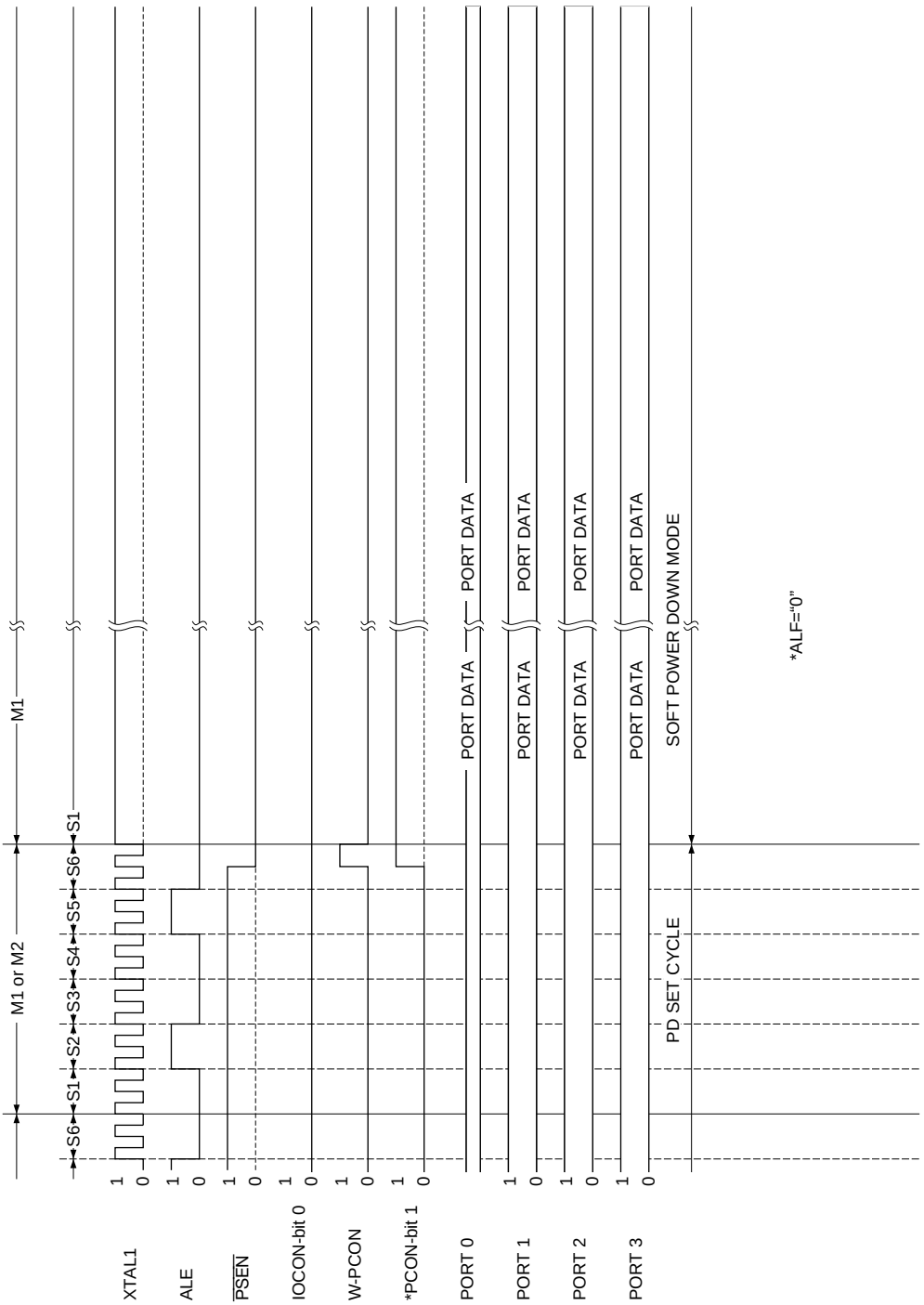


Figure 4-60 Soft power down mode setting time chart (internal ROM mode)

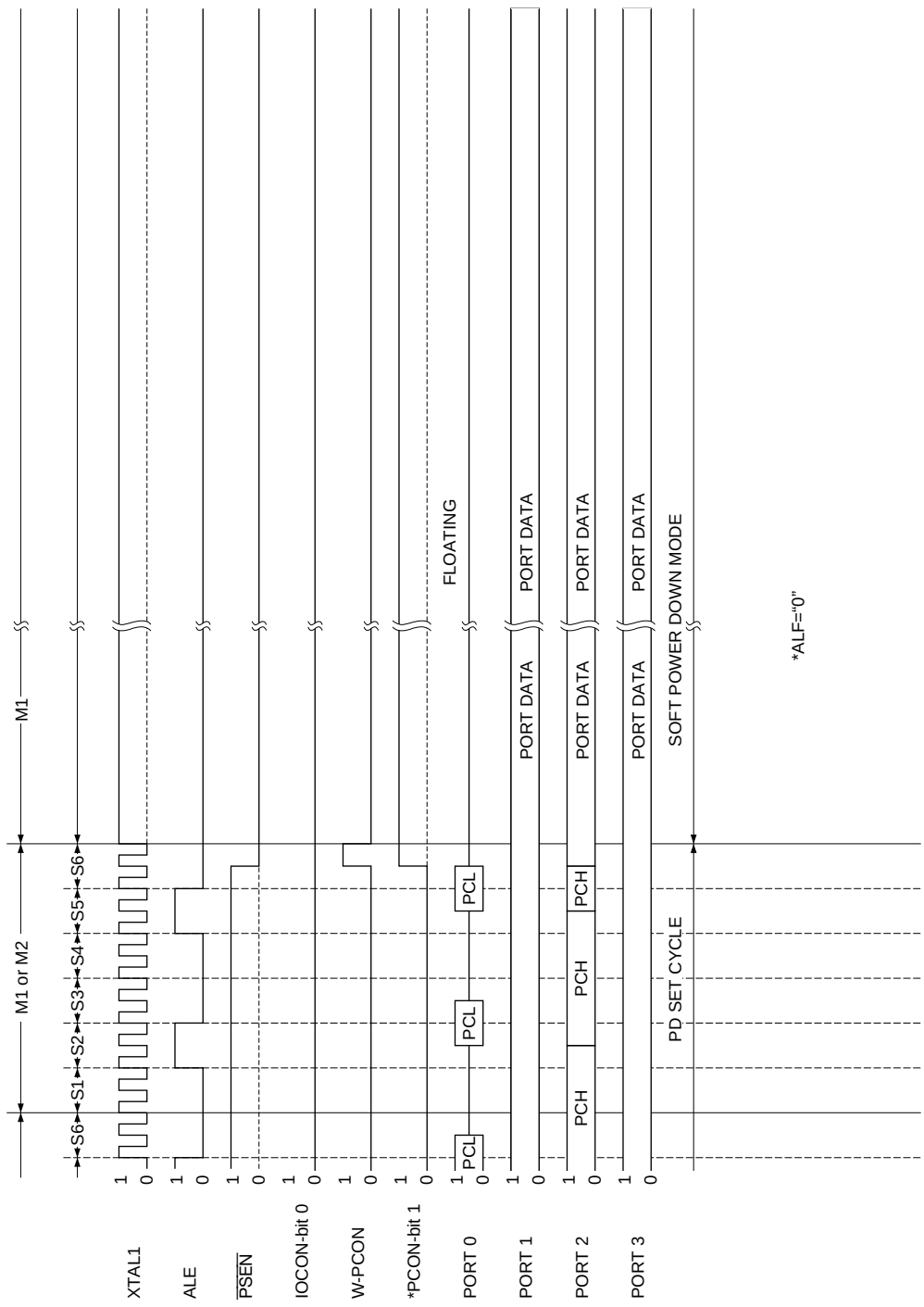


Figure 4-61 Soft power down mode setting time chart (external ROM mode)

Table 4-25 CPU pin details (ALF=1) in soft power down mode (PD)

Name	Internal ROM	External ROM
P1.0/T2	Floating	Floating
P1.1/T2EX	Floating	Floating
P1.2	Floating	Floating
P1.3	Floating	Floating
P1.4	Floating	Floating
P1.5	Floating	Floating
P1.6	Floating	Floating
P1.7	Floating	Floating
RESET	"0" level input	"0" level input
P3.0/RXD	Floating	Floating
P3.1/TXD	Floating	Floating
P3.2/$\overline{\text{INT}}$0	External data input	External data input
P3.3/$\overline{\text{INT}}$1	External data input	External data input
P3.4/ T0	External data input	External data input
P3.5/ T1/HPDI	External data input	External data input
P3.6/$\overline{\text{WR}}$	Floating	Floating
P3.7/$\overline{\text{RD}}$	Floating	Floating
XTAL 2	Oscillator operative	Oscillator operative
XTAL 1	Oscillator operative	Oscillator operative
V_{SS}	0 [V]	0 [V]
P2.0	Floating	Floating
P2.1	Floating	Floating
P2.2	Floating	Floating
P2.3	Floating	Floating
P2.4	Floating	Floating
P2.5	Floating	Floating
P2.6	Floating	Floating
P2.7	Floating	Floating
$\overline{\text{PSEN}}$	"0" level output	"0" level output
ALE	"0" level output	"0" level output
$\overline{\text{EA}}$	"1" level input	"0" level input
P0.7	Floating	Floating
P0.6	Floating	Floating
P0.5	Floating	Floating
P0.4	Floating	Floating
P0.3	Floating	Floating
P0.2	Floating	Floating
P0.1	Floating	Floating
P0.0	Floating	Floating
V_{CC}	*+2.0~+6 [V]	*+2.0~+6 [V]

* V_{CC}=+2.0~+6V when internal CPU data is held.

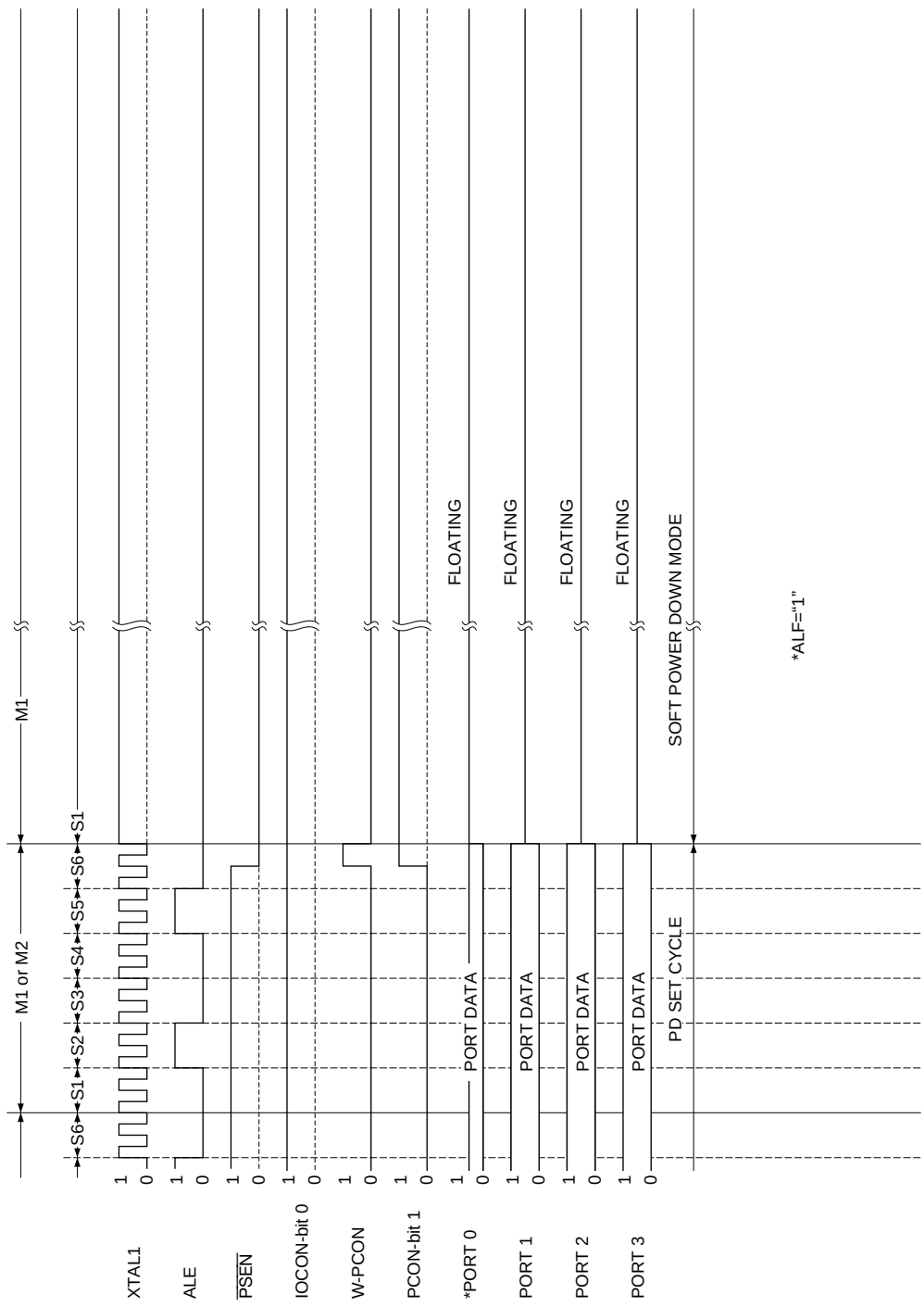


Figure 4-62 Soft power down mode setting and I/O floating time chart (internal ROM mode)

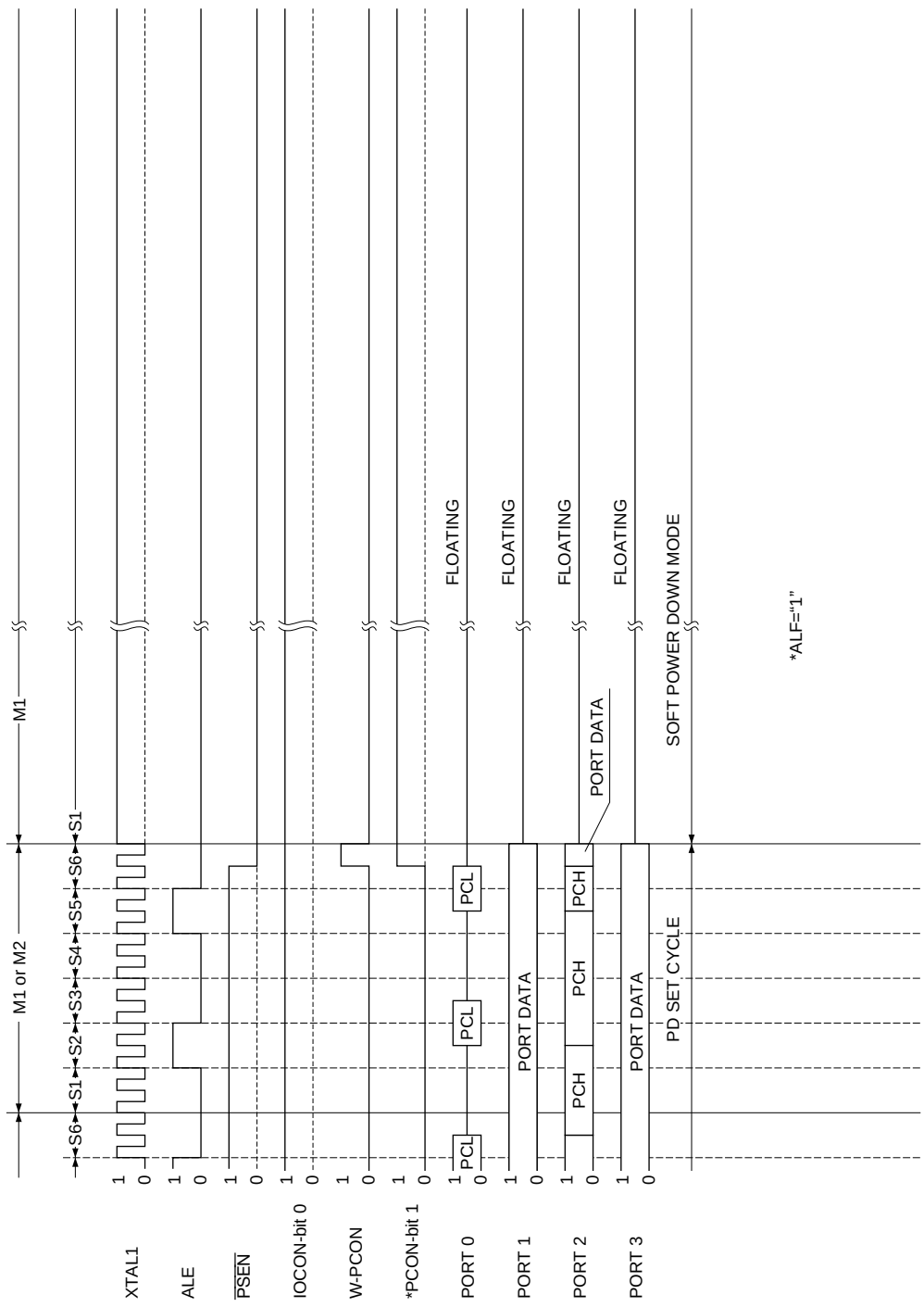


Figure 4-63 Soft power down mode setting and I/O floating time chart (external ROM mode)

4.8.4 Hard power down mode (HPD) setting

To set hard power down mode (HPD), “1” is set in bit 6 (HPD) of the power control register (PCON 87H) in advance to attain the circuit connections shown in Figure 4-61. Hard power down mode is set when the level of the power failure detect signal applied to the HPDI pin (bit 5 of port 3) is changed from level “1” to level “0”. XTAL1·2 operations are stopped in this mode. And while all internal data is retained, the CPU operations also are stopped apart from timer/counter 0 and 1. (Timer/counters 0 and 1 operate in external clock mode.)
The pin output status of ports 0 thru 3 in hard power down mode can be left in port data output status, or set to port output floating status.
The ports are set to data output status by setting bit 0 (ALF) of the I/O control register (IOCON 0F8H) to “0” when hard power down mode is activated, and to floating status by setting ALF to “1” before activating power down mode. In floating status, the port pins are disconnected electrically from the external circuitry.
Apart from pins 2, 3, 4, and 5 of port 3, all floating status input port pins may be open, or undefined within the –0.5 to V_{CC}+0.5 V range.
The CPU pin status during hard power down mode (HPD) with “0” on the ALF bit is outlined in Table 4-26, and the corresponding time charts for starting hard power down mode are shown in Figures 4-65 and 4-66.
And the CPU pin status during hard power down mode (HPD) with “1” on the ALF bit is outlined in Table 4-27, and the corresponding time charts for starting hard power down mode are shown in Figures 4-67 and 4-68.

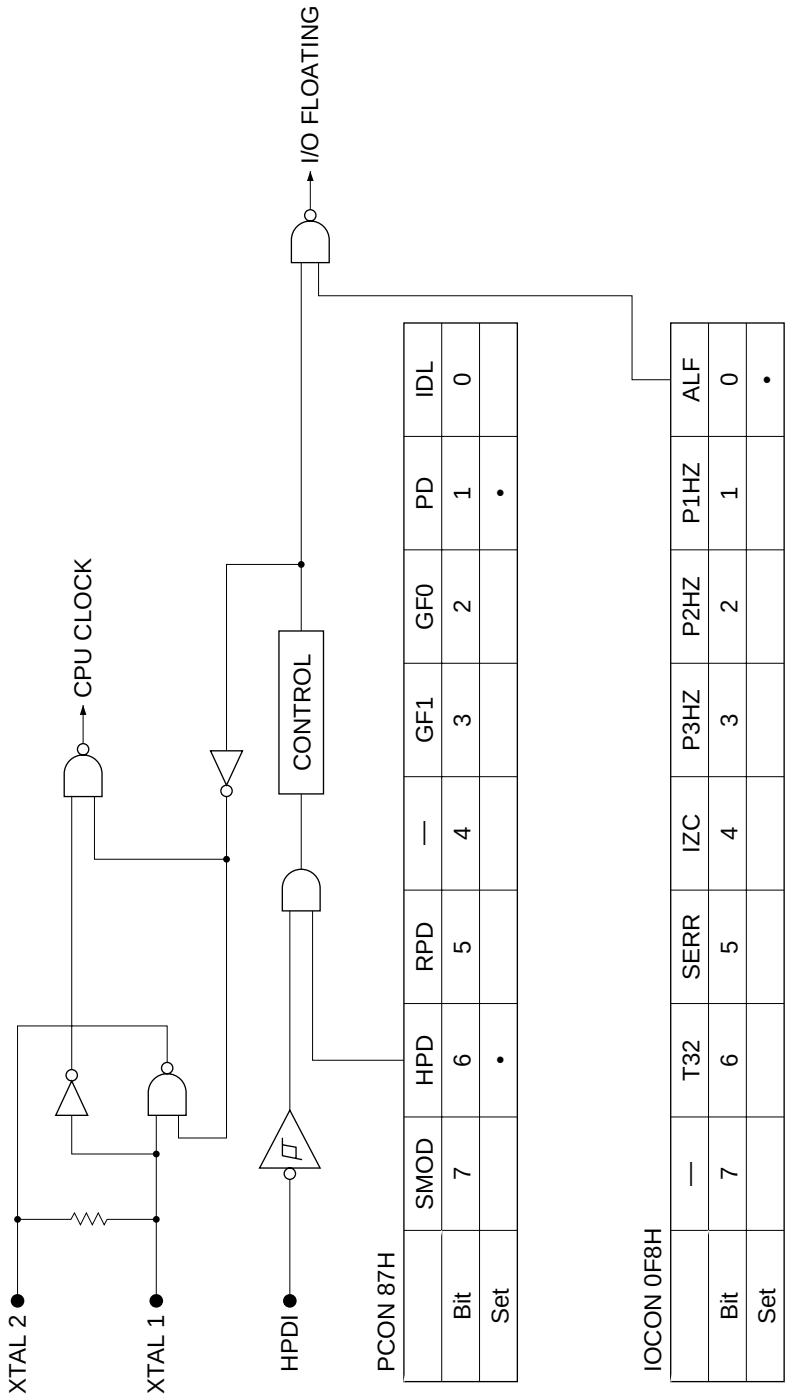


Figure 4-64 Hard power down mode equivalent circuit

Table 4-26 CPU pin details (ALF=0) in hard power down mode (HPD)

Name	Internal ROM	External ROM
P1.0/T2	Port data output	Port data output
P1.1/T2EX	Port data output	Port data output
P1.2	Port data output	Port data output
P1.3	Port data output	Port data output
P1.4	Port data output	Port data output
P1.5	Port data output	Port data output
P1.6	Port data output	Port data output
P1.7	Port data output	Port data output
RESET	“0” level input	“0” level input
P3.0/RXD	Port data output	Port data output
P3.1/TXD	Port data output	Port data output
P3.2/$\overline{\text{INT}}$0	Port data output	Port data output
P3.3/$\overline{\text{INT}}$1	Port data output	Port data output
P3.4/ T0	Port data output	Port data output
P3.5/ T1/HPDI	“0” level input	“0” level input
P3.6/$\overline{\text{WR}}$	Port data output	Port data output
P3.7/$\overline{\text{RD}}$	Port data output	Port data output
XTAL 2	Oscillator operative	Oscillator operative
XTAL 1	Oscillator operative	Oscillator operative
V_{SS}	0 [V]	0 [V]
P2.0	Port data output	Port data output
P2.1	Port data output	Port data output
P2.2	Port data output	Port data output
P2.3	Port data output	Port data output
P2.4	Port data output	Port data output
P2.5	Port data output	Port data output
P2.6	Port data output	Port data output
P2.7	Port data output	Port data output
$\overline{\text{PSEN}}$	“0” level output	“0” level output
ALE	“0” level output	“0” level output
$\overline{\text{EA}}$	“1” level input	“0” level input
P0.7	Port data output	Floating
P0.6	Port data output	Floating
P0.5	Port data output	Floating
P0.4	Port data output	Floating
P0.3	Port data output	Floating
P0.2	Port data output	Floating
P0.1	Port data output	Floating
P0.0	Port data output	Floating
V_{CC}	*+2.0~+6 [V]	*+2.0~+6 [V]

* V_{CC}=+2.0~+6V when internal CPU data is held.

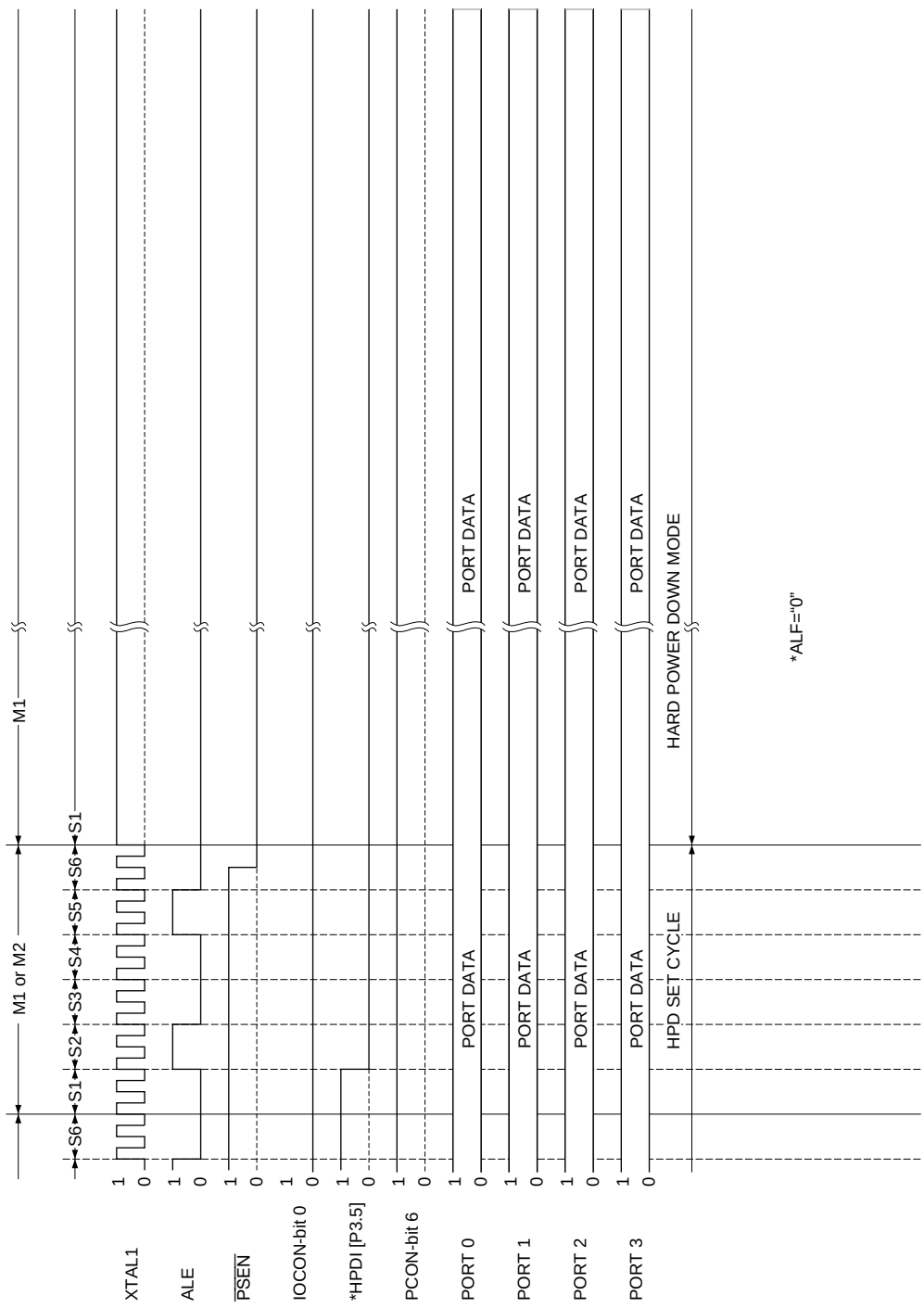


Figure 4-65 Hard power down mode setting time chart (internal ROM mode)

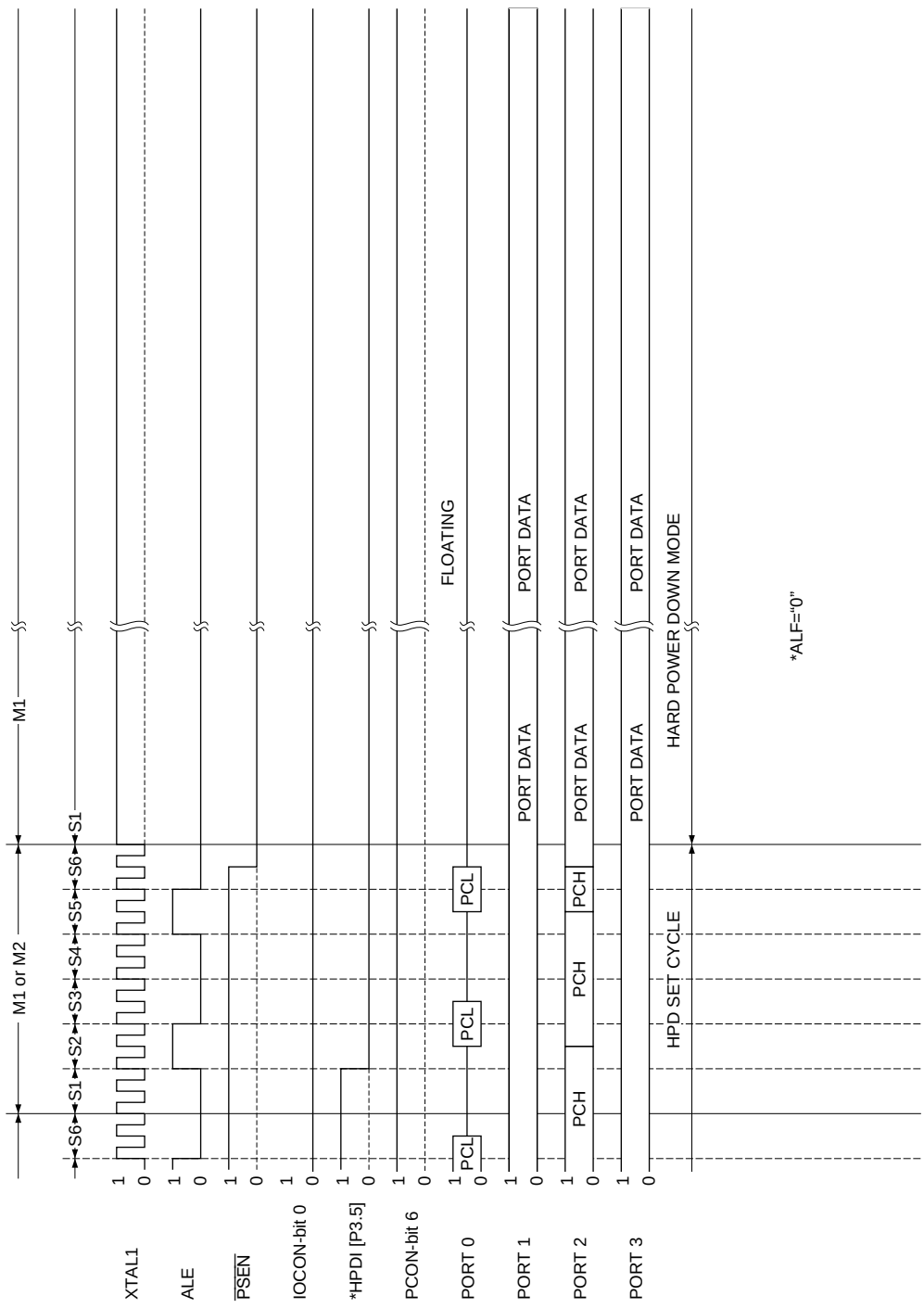


Figure 4-66 Hard power down mode setting time chart (external ROM mode)

Table 4-27 CPU pin details (ALF=1) in hard power down mode (HPD)

Name	Internal ROM	External ROM
P1.0/T2	Floating	Floating
P1.1/T2EX	Floating	Floating
P1.2	Floating	Floating
P1.3	Floating	Floating
P1.4	Floating	Floating
P1.5	Floating	Floating
P1.6	Floating	Floating
P1.7	Floating	Floating
RESET	“0” level input	“0” level input
P3.0/RXD	Floating	Floating
P3.1/TXD	Floating	Floating
P3.2/$\overline{\text{INT}}$0	External data input	External data input
P3.3/$\overline{\text{INT}}$1	External data input	External data input
P3.4/ T0	External data input	External data input
P3.5/ T1/HPDI	“0” level input	“0” level input
P3.6/$\overline{\text{WR}}$	Floating	Floating
P3.7/$\overline{\text{RD}}$	Floating	Floating
XTAL 2	Oscillator operative	Oscillator operative
XTAL 1	Oscillator operative	Oscillator operative
V_{SS}	0 [V]	0 [V]
P2.0	Floating	Floating
P2.1	Floating	Floating
P2.2	Floating	Floating
P2.3	Floating	Floating
P2.4	Floating	Floating
P2.5	Floating	Floating
P2.6	Floating	Floating
P2.7	Floating	Floating
$\overline{\text{PSEN}}$	“0” level output	“0” level output
ALE	“0” level output	“0” level output
$\overline{\text{EA}}$	“1” level input	“0” level input
P0.7	Floating	Floating
P0.6	Floating	Floating
P0.5	Floating	Floating
P0.4	Floating	Floating
P0.3	Floating	Floating
P0.2	Floating	Floating
P0.1	Floating	Floating
P0.0	Floating	Floating
V_{CC}	*+2.0~+6 [V]	*+2.0~+6 [V]

* V_{CC}=+2.0~+6V when internal CPU data is held.

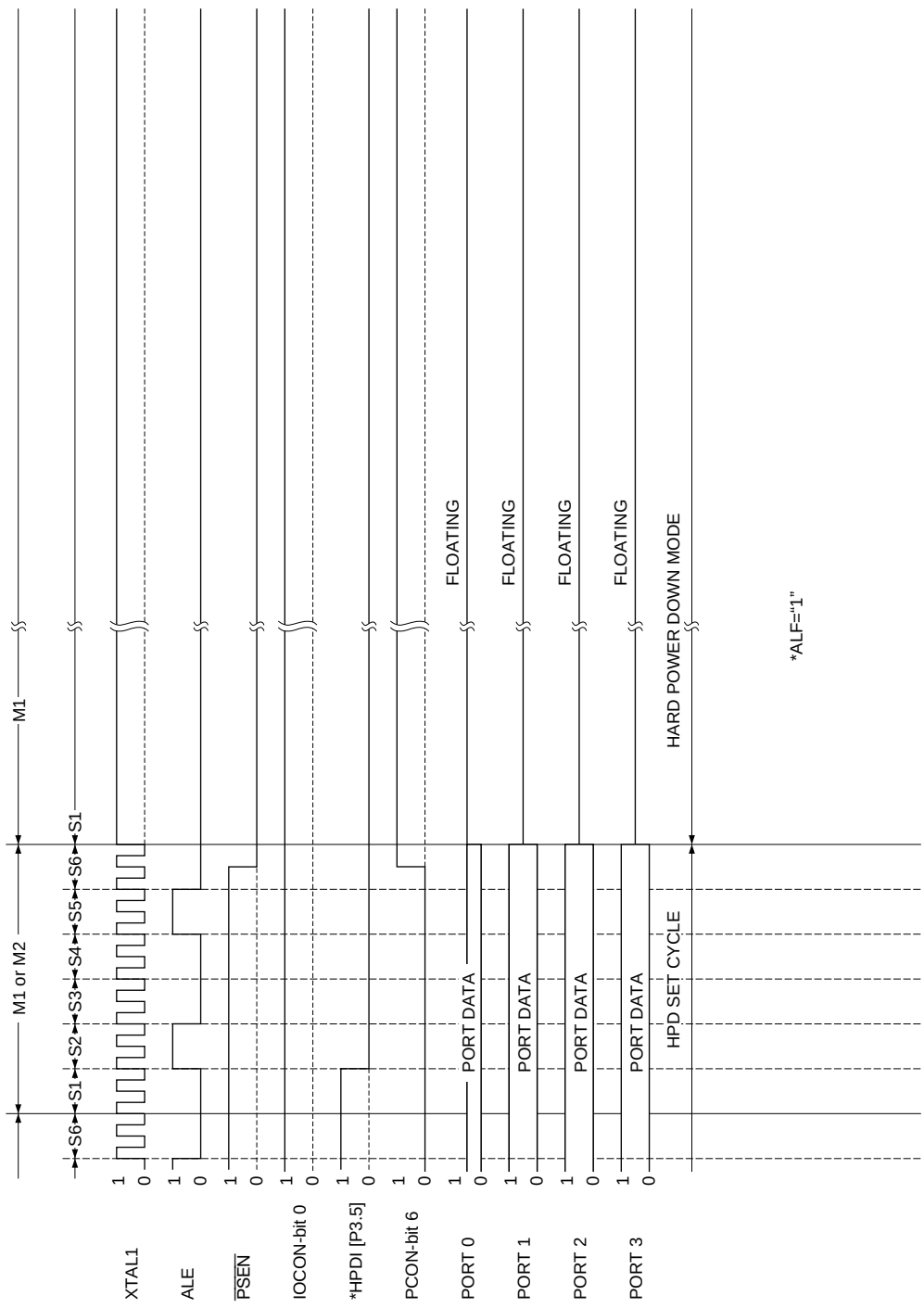


Figure 4-67 Hard power down mode setting and I/O floating time chart (internal ROM mode)

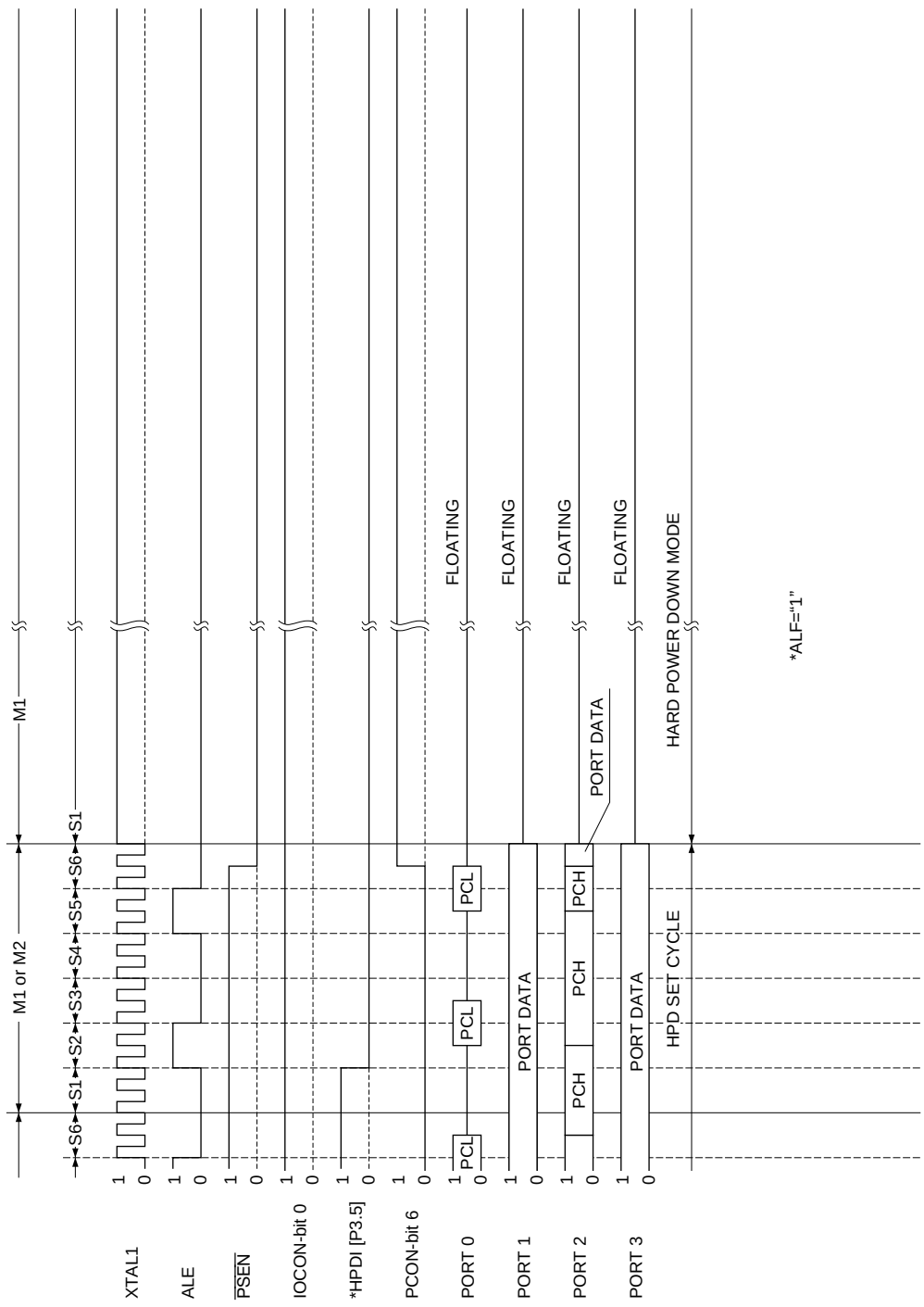


Figure 4-68 Hard power down mode setting andl/Of loating time chart (external ROM mode)

4.9 CPU Power Down Mode (IDLE, PD, and HPD) Cancellation (CPU Activation)

4.9.1 Outline

CPU power down mode (IDLE, PD, and HPD) can be cancelled (CPU activation) in the following two ways.
The CPU is reset when a “1” reset signal is applied to the CPU RESET pin, and the program is executed from address 0. This method can be used in IDLE, PD, and HPD modes.
By generating the respective interrupt source signals, the program can be executed from the interrupt address, and can also be continued from the next address after the stop address. This method can be used in IDLE and PD modes.

4.9.2 Cancellation by CPU resetting (RESET pin)

The CPU is reset when a “1” level signal is applied (for at least 1μAsec.) to the CPU RESET pin, and the CPU power down mode (IDLE, PD, or HPD) is cancelled. Programs are subsequently executed by the CPU from address 0. The reset cancellation time charts are outlined in Figures 4-69 thru 4-74.

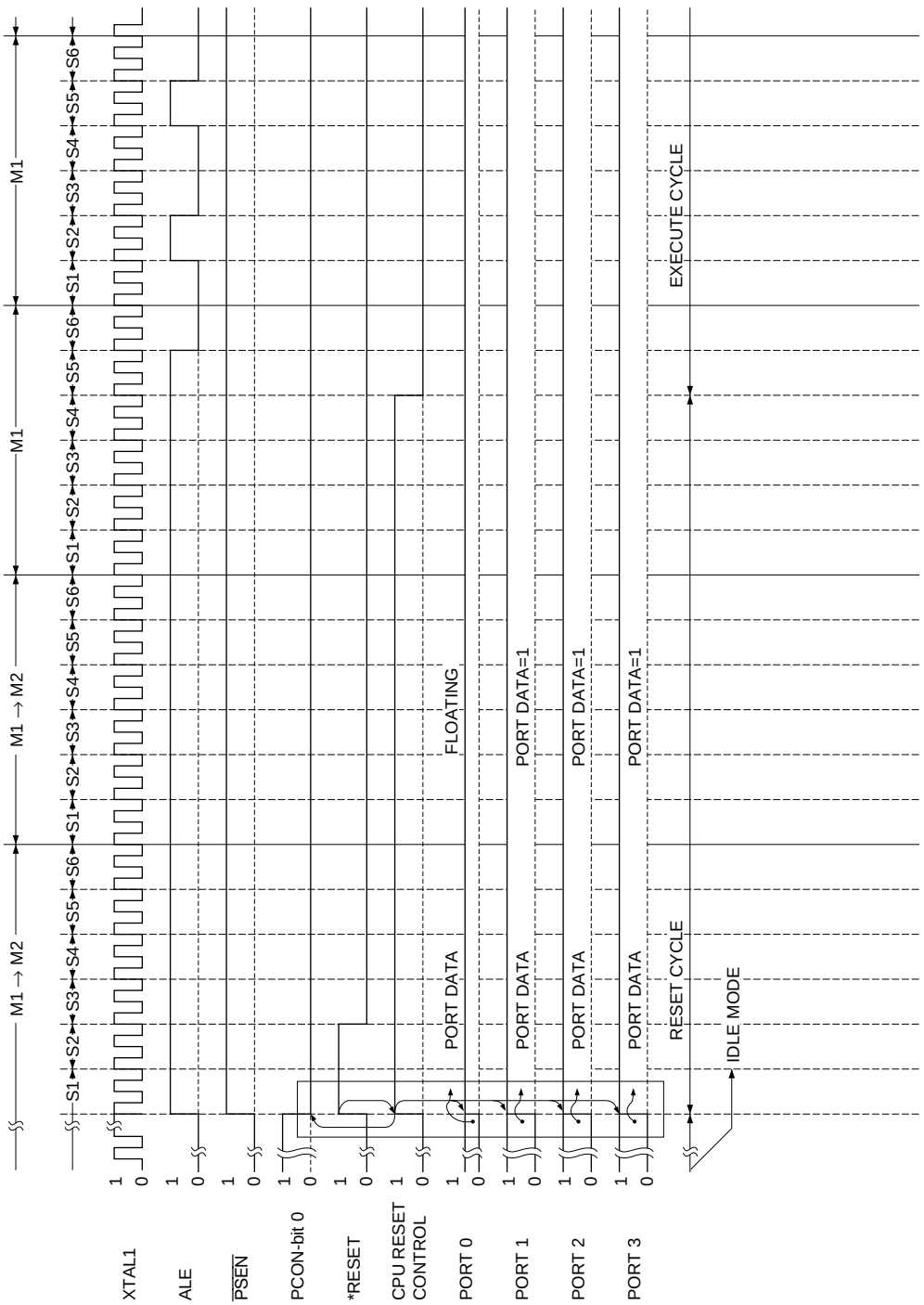


Figure 4-69 Restart from idle mode by reset (internal ROM mode)

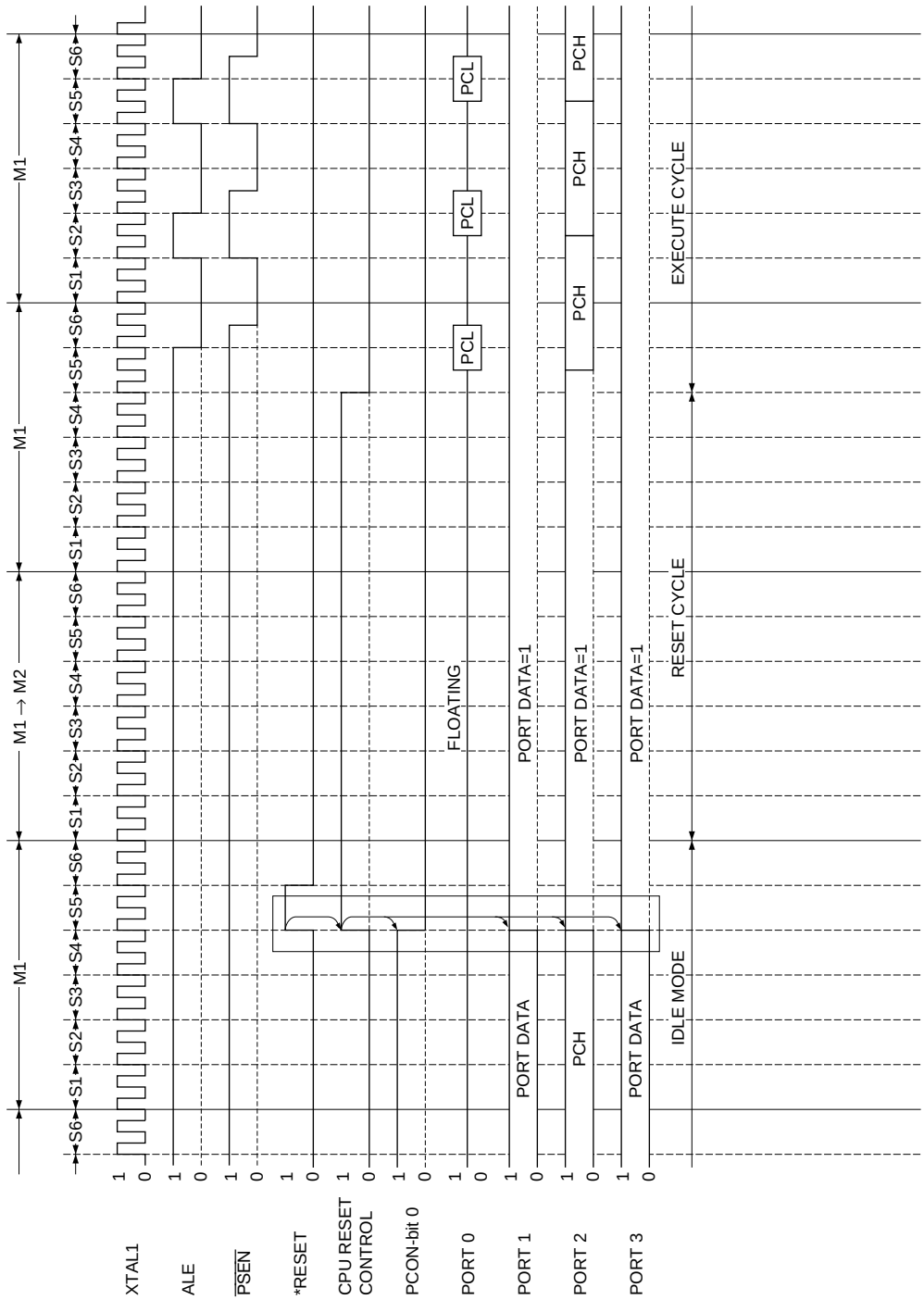


Figure 4-70 Restart from idle mode by reset (external ROM mode)

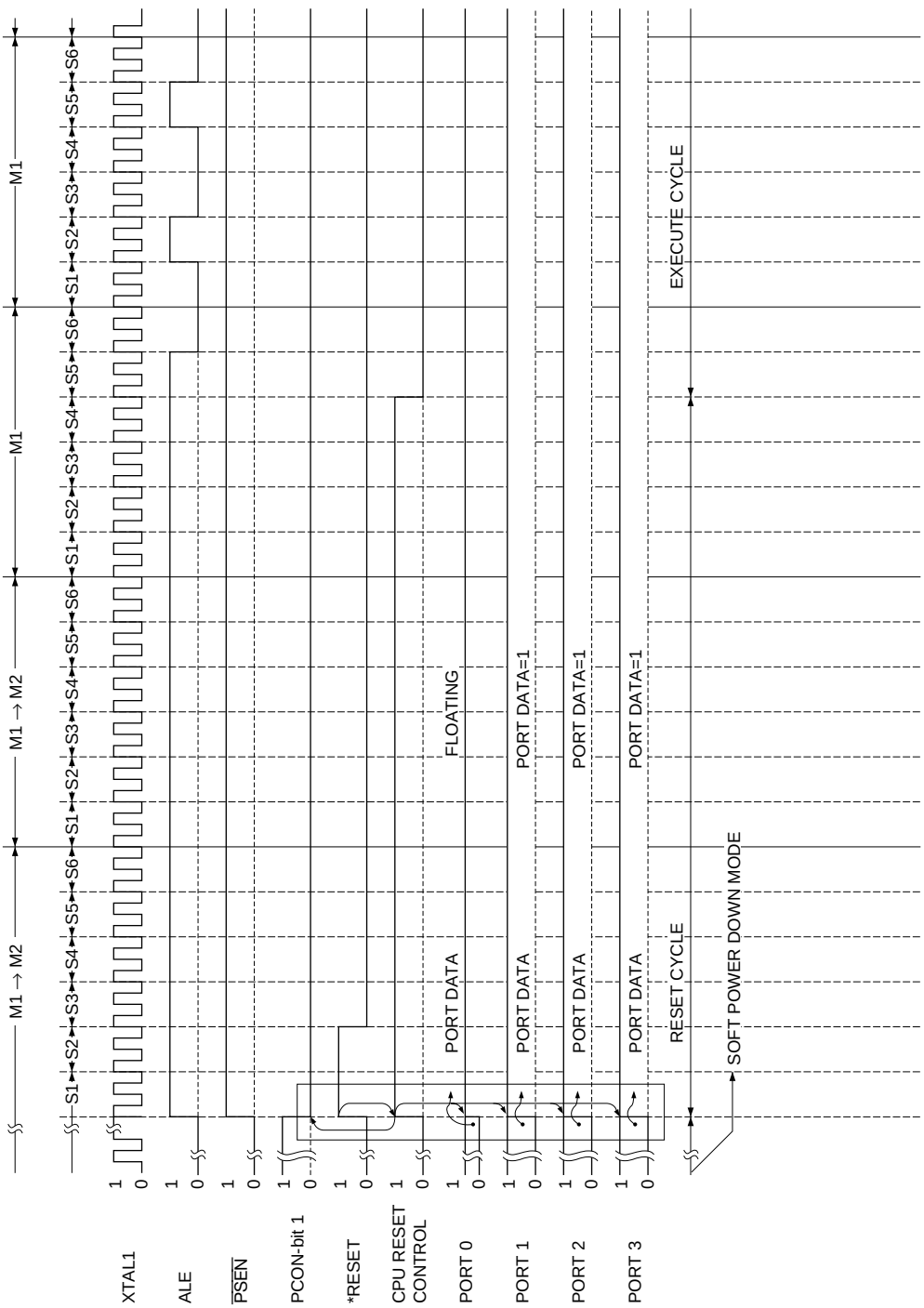


Figure 4-71 Restart from soft power mode by reset (internal ROM mode)

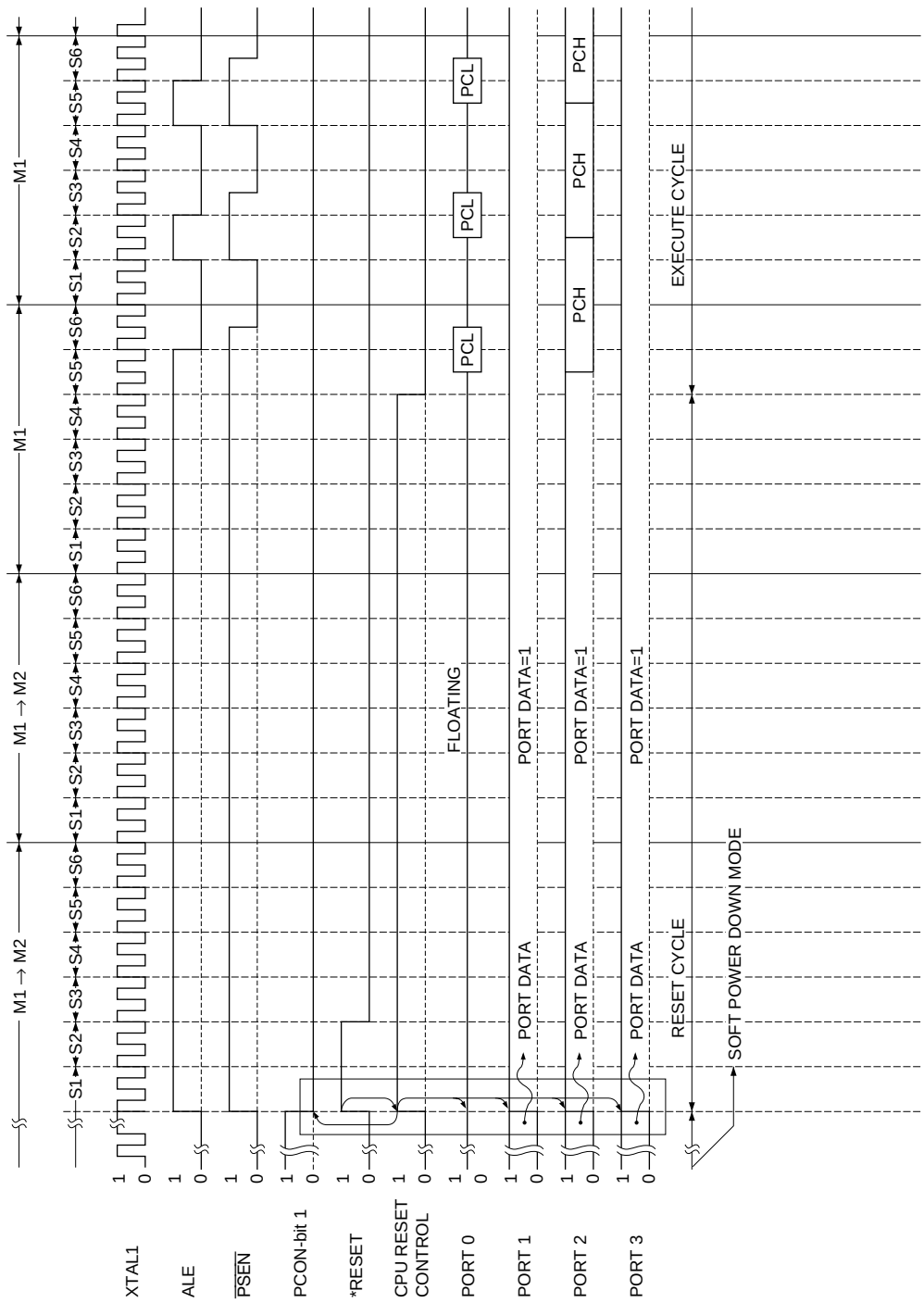


Figure 4-72 Restart from soft power mode by reset (external ROM mode)

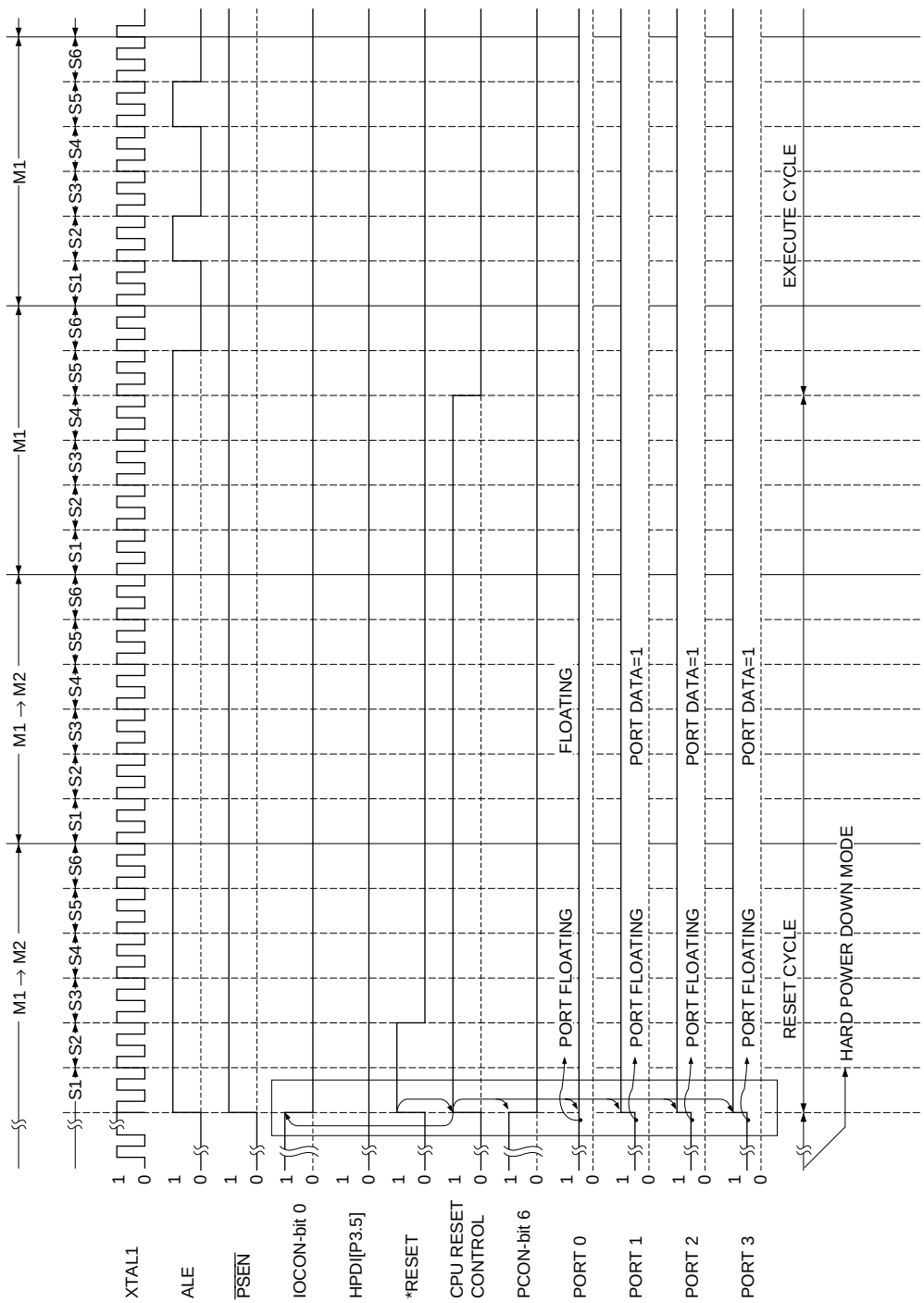


Figure 4-73 Restart from hard power down mode by reset (internal ROM mode)

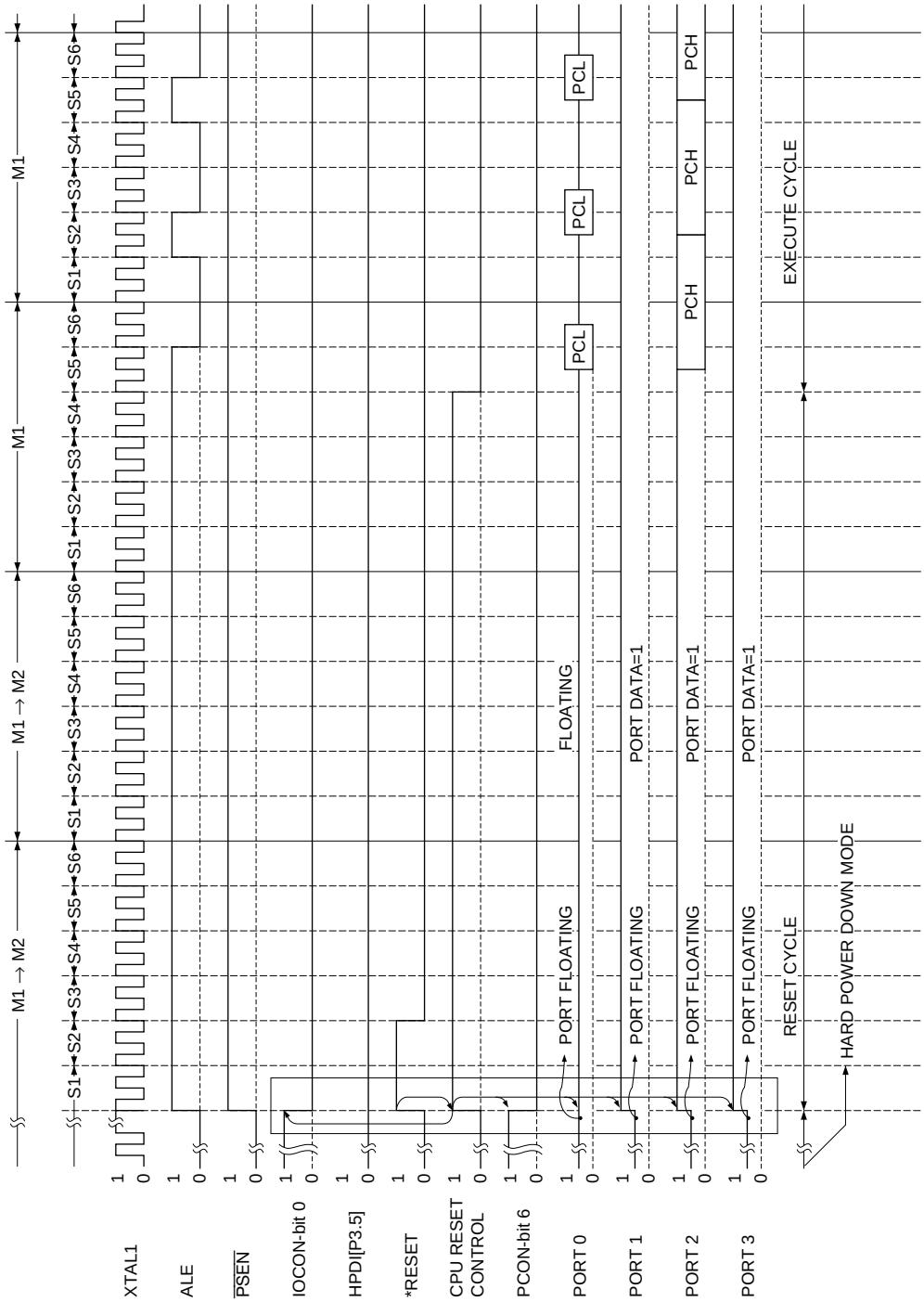


Figure 4-74 Restart from hard power down mode by reset (external ROM mode)

4.9.3 Cancellation of CPU power down mode (IDLE, PD) by interrupt signal

When idle mode (IDLE) and soft power down mode (PD) are cancelled by interrupt signal, power down mode cancellation condition is determined by bit 5 (RPD) of the power control register (PCON 87H) shown in Table 4-29.
When RPD is “0”, power down mode can be cancelled by interrupt signal and CPU executes program from the interrupt address only when the CPU has been set to interrupt enable status.
And when RPD is “1”, power down mode can be cancelled and resumes execution from the next address after the stop address if “1” is set in the interrupt flag by interrupt signal even when the CPU is in interrupt disable mode.
The conditions for cancellation of power down mode by interrupt signal can thus be specified by the RPD content.

Table 4-29 Power control register (PCON 87H)

	SMOD	HPD	RPD	—	GF1	GF0	PD	IDL
Bit	7	6	5	4	3	2	1	0
Set			•					

4.9.3.1 Cancellation of CPU power down mode (IDLE, PD) from interrupt address

To cancel idle mode (IDLE) or soft power down mode (PD) and resume execution from the interrupt address, an interrupt is specified in the interrupt enable register (IE 0A8H) prior to setting CPU power down mode and “0” is set in bit 5 (RPD) of the power control register (PCON 87H).
All six interrupts can be used to cancel idle mode. The interrupt conditions are satisfied when “1” is set in the specified interrupt flag in TCON, T2CON, or SCON. Clock signals are then passed to the CPU, and execution is commenced from the interrupt address.
Soft power down mode (PD) can be cancelled by four different interrupts - external interrupts 0 and 1, and timer interrupts 0 and 1. (Timer/counters 0 and 1 are operated in external clock mode.)
The external interrupts are generated by “0” level being applied to either the $\overline{INT}0$ or $\overline{INT}1$ pin. When the specified interrupt flag in TCON is set to “1” to satisfy the interrupt conditions, XTAL1·2 operation is commenced, and the program is executed from the interrupt address. When the interrupt routine is completed, the program returns to the next address after the stop address.
If all interrupts have been disabled, however, CPU power down mode cannot be cancelled from the interrupt address by this method. A “1” reset signal must be applied to the RESET pin and execution commenced from address 0 in this case. The equivalent circuit involved in CPU power down mode cancellation by interrupt is shown in Figure 4-75, and the CPU power down mode (PD, HPD) cancellation time charts are shown in Figures 4-76 thru 4-79.

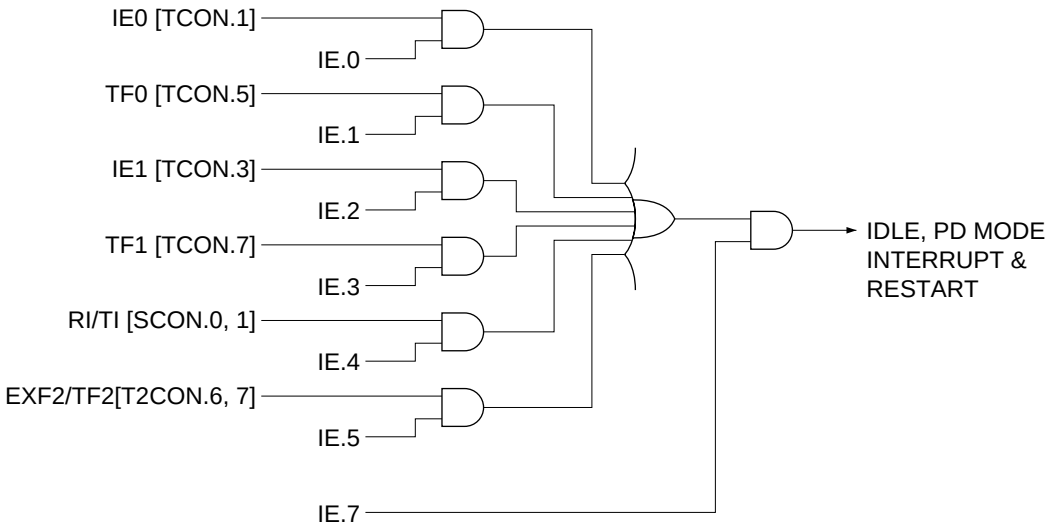


Figure 4-75 Equivalent circuit for, DLE and PD mode rancellation by interrupt signal

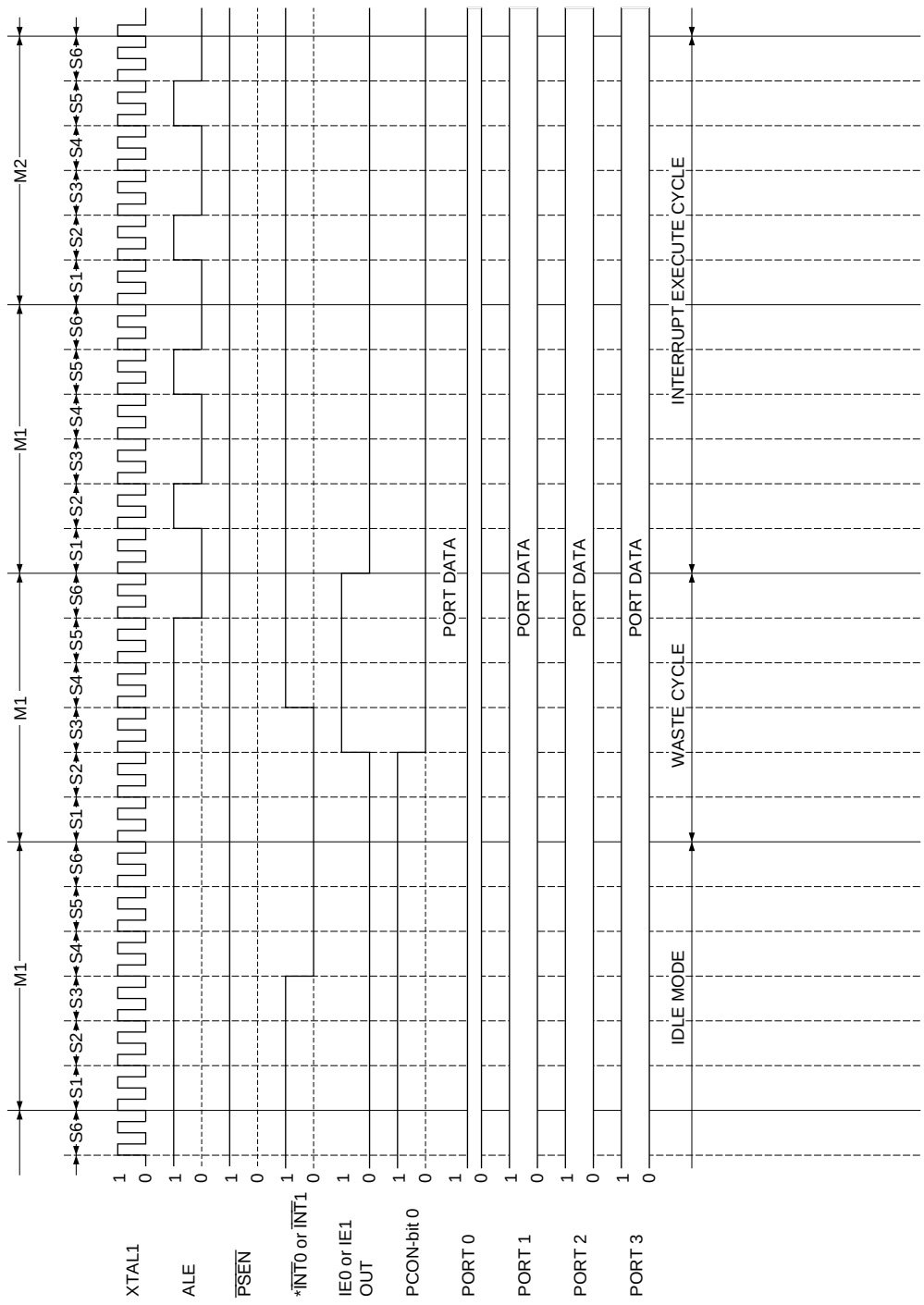


Figure 4-76 Restart from idle mode by interrupt $\overline{\text{INT}}$0 or 1 (internal ROM mode)

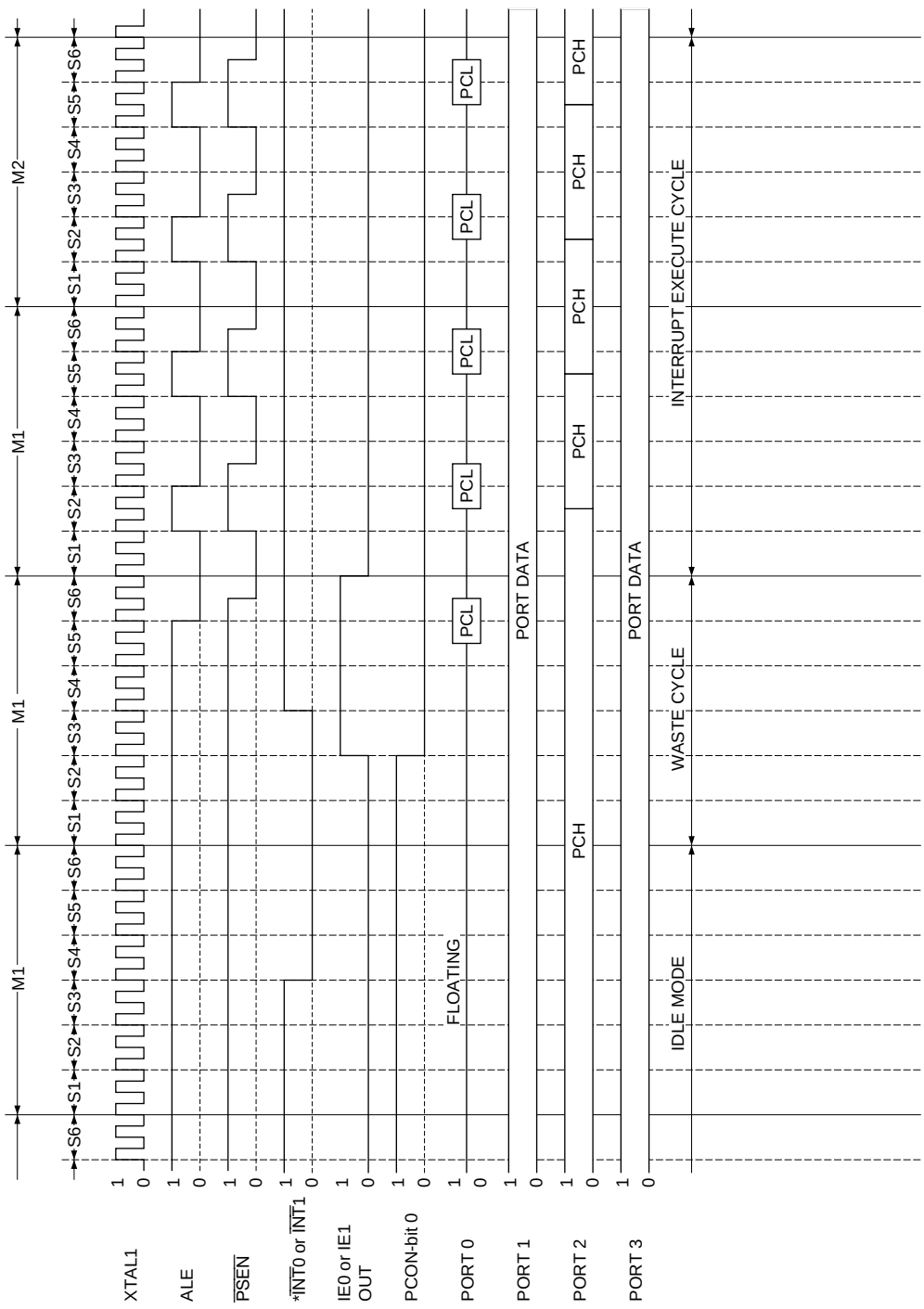


Figure 4-77 Restart from idle mode by interrupt INT0 or 1 (external ROM mode)

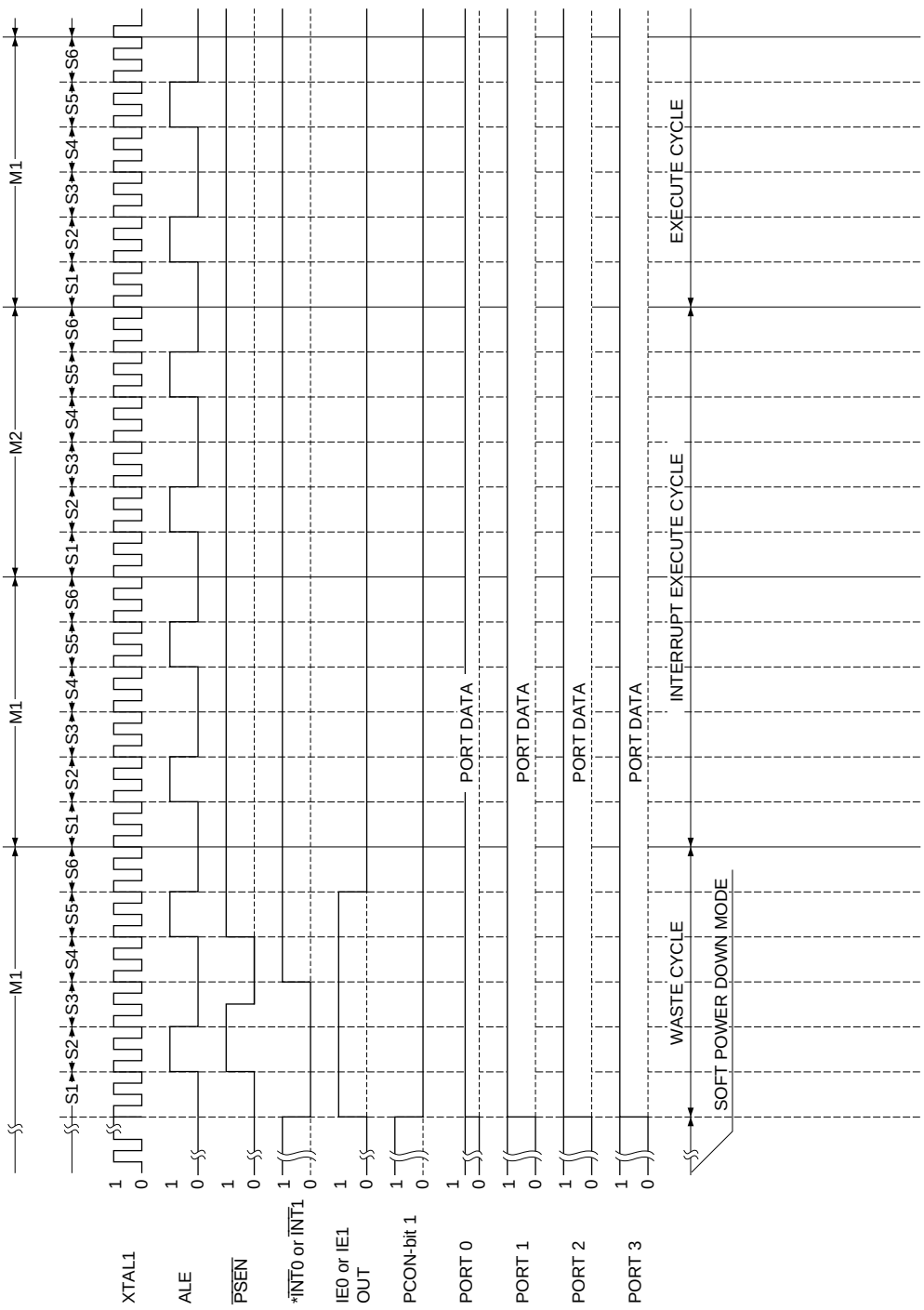


Figure 4-78 Restart from soft power down mode by Interrupt $\overline{\text{INT}}$0 or 1 (internal ROM mode)

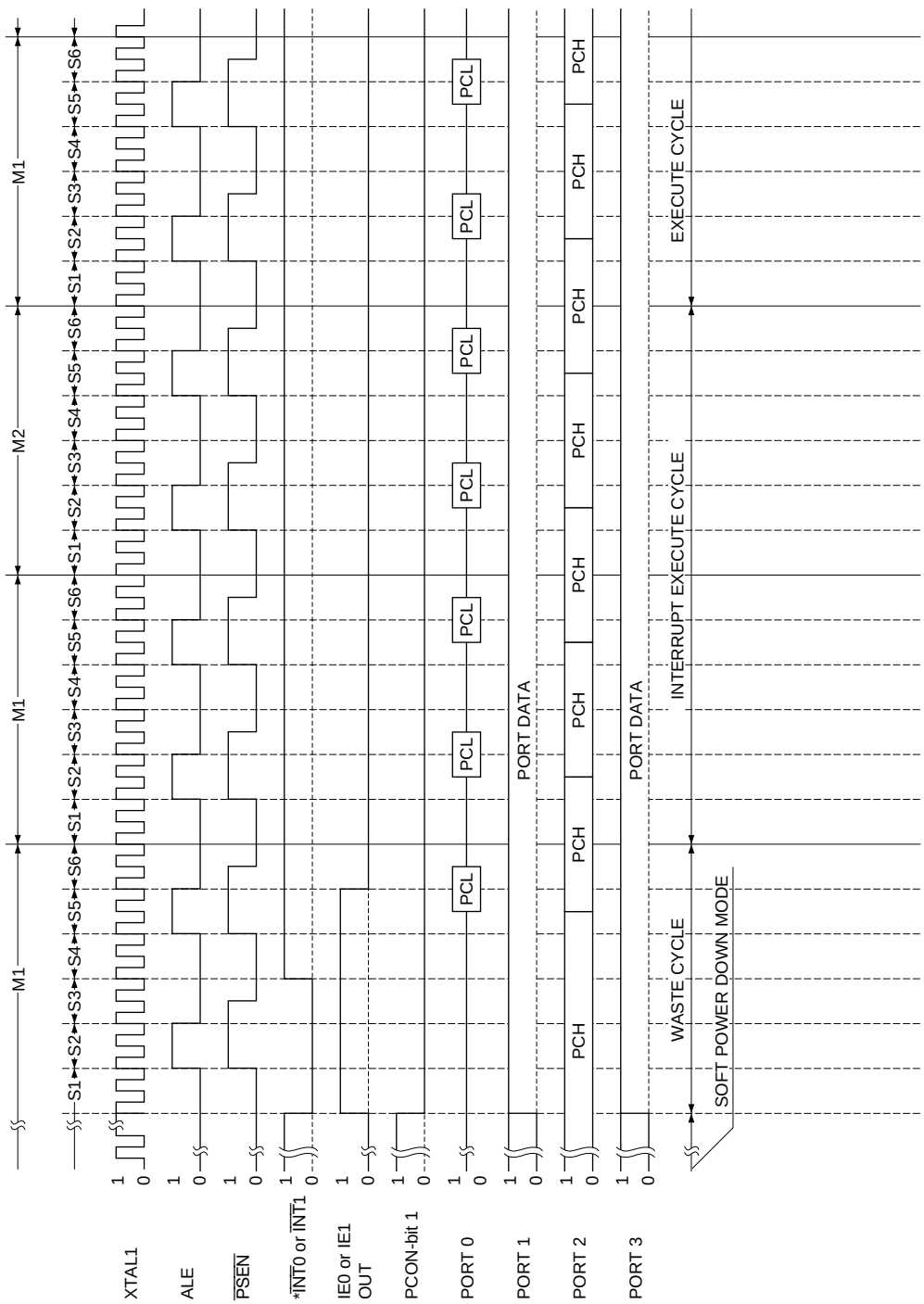
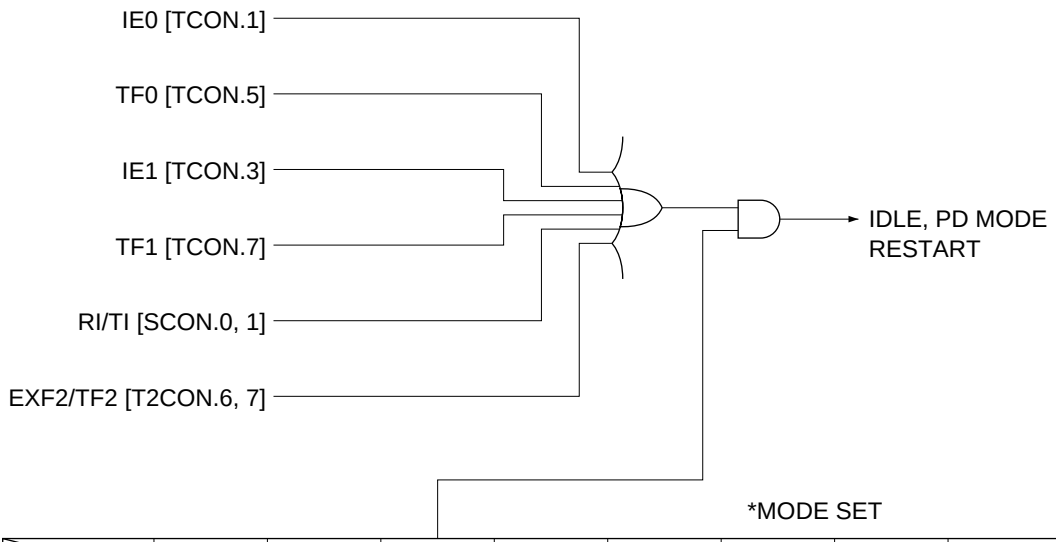


Figure 4-79 Restart from soft power down mode by interrupt INT0 or 1 (external ROM mode)

4.9.3.2 Cancellation of CPU power down mode (IDLE, PD) by interrupt request signal and restart from next address of stop address

To cancel idle mode (IDLE) or soft power down mode (PD) by interrupt request signal and then resume execution from the next address after the stop address, "1" is set in bit 5 (RPD) of the power control register. When "1" is set in this bit, the circuit connections shown in Figure 4-80 are made, and the CPU power down mode is cancelled when the interrupt flag has been set to "1", even if the entire contents of the interrupt enable register (IE 0A8H) have been put into interrupt disable status.
All six interrupt sources can be used to cancel idle mode (IDLE). If an interrupt source is generated and "1" is set in one of the interrupt flags in TCON, T2CON, or SCON, clock signals are passed to the CPU control stage, and execution is resumed from the next address after the stop address.
Soft power down mode (PD) can be cancelled by four different interrupt sources - external interrupts 0 and 1 , and timer interrupts 0 and 1. The external interrupt flag is set by "0" level being applied to either the $\overline{INT}0$ or $\overline{INT}1$ pin. And timer/counters 0 and 1 are used in external clock mode. When one of the interrupt flags in TCON is set to "1", XTAL1·2 operation is commenced, and the program is executed from the next address after the stop address.
Note, however, that the interrupt flags are reset by software. The cancellation time charts are shown in Figures 4-81 thru 4-84.



	SMOD	HPD	RPD	—	GF1	GF0	PD	IDL
Bit	7	6	5	4	3	2	1	0
Set			•				*	*

Figure 4-80 Equivalent circuit for power down mode cancellation and restart by interrupt source signal

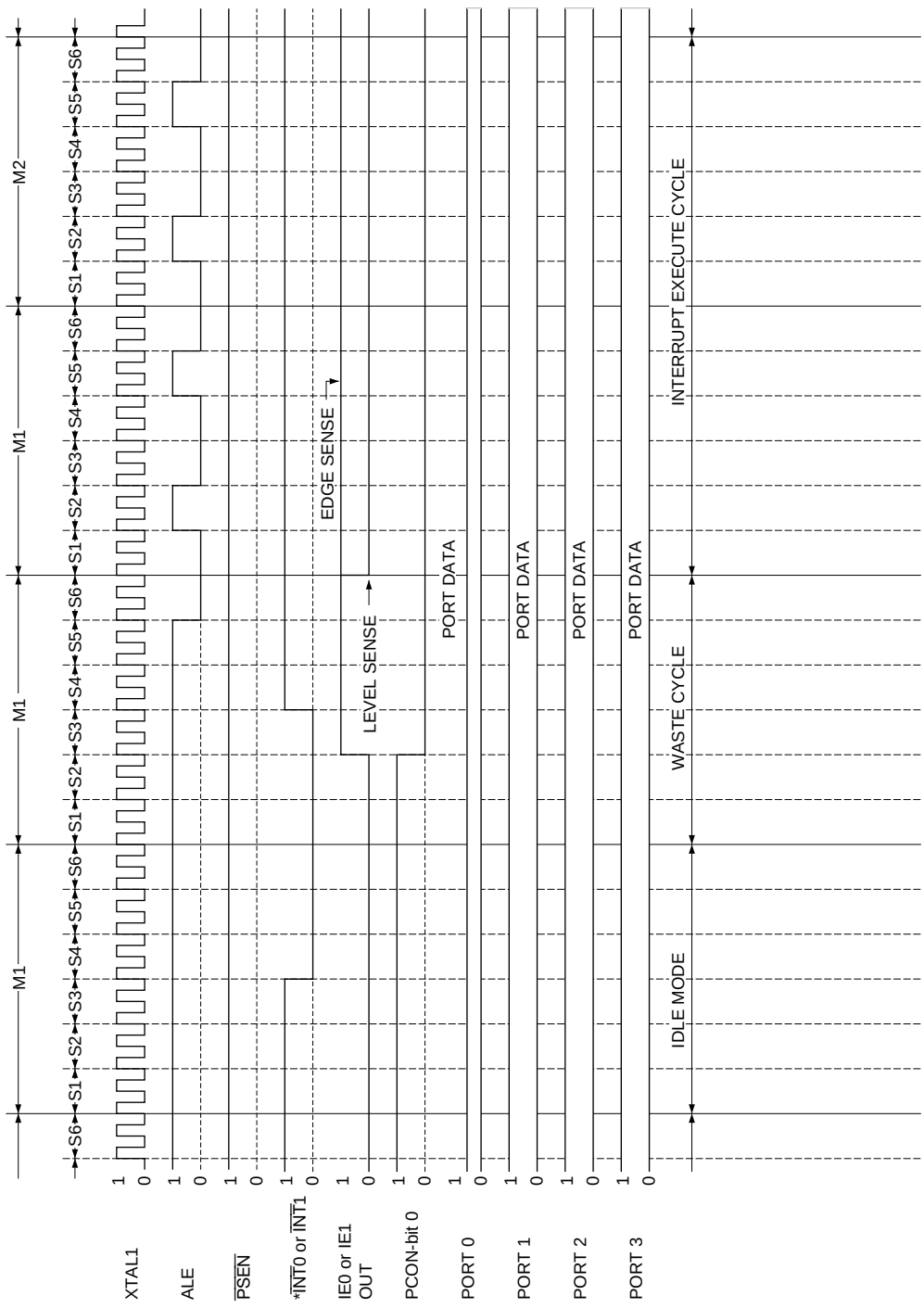


Figure 4-81 Restart from idle mode by INT0 or 1 (internal ROM mode)

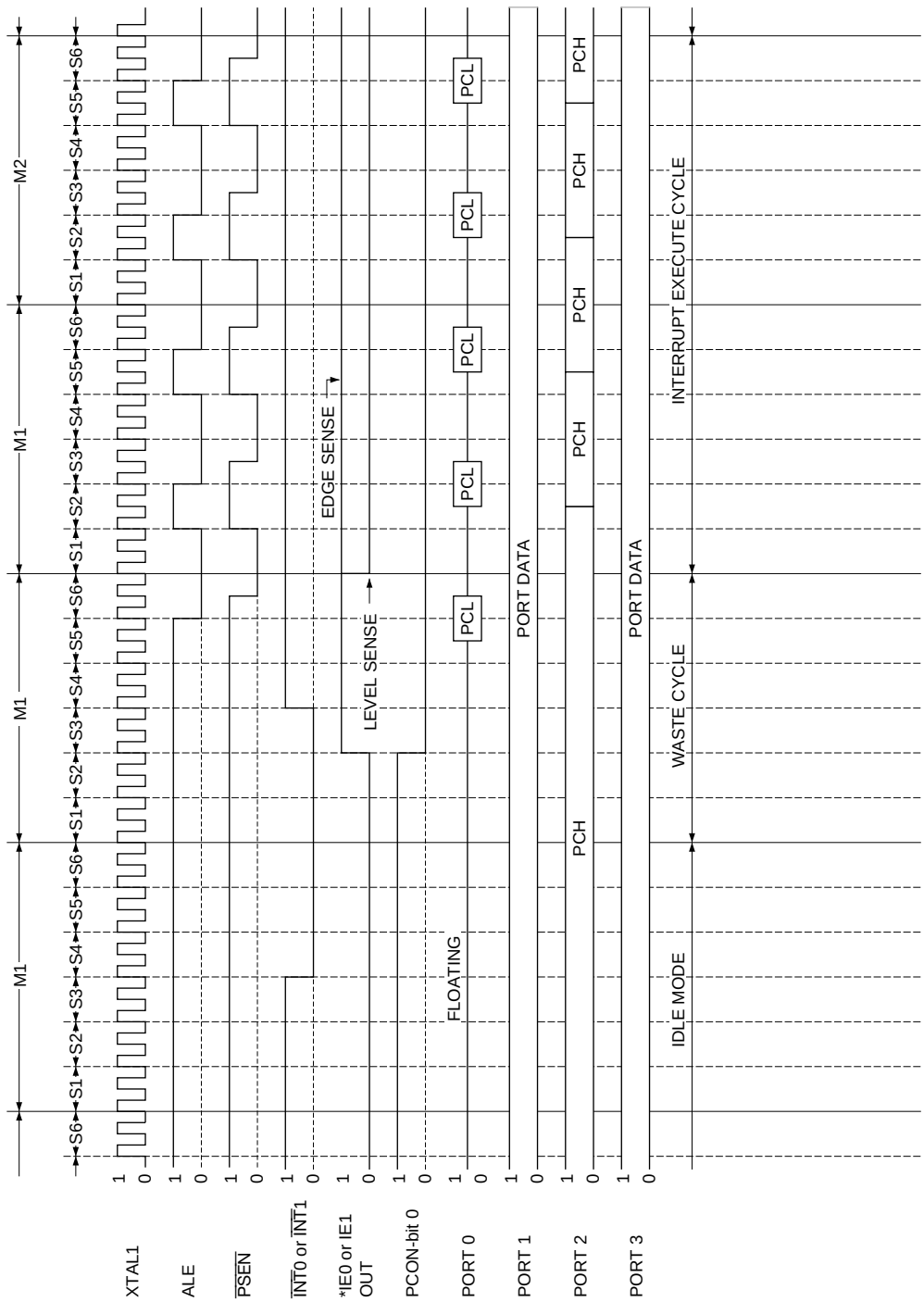


Figure 4-82 Restart from idle mode by $\overline{\text{INT}}$0 or 1 (external ROM mode)

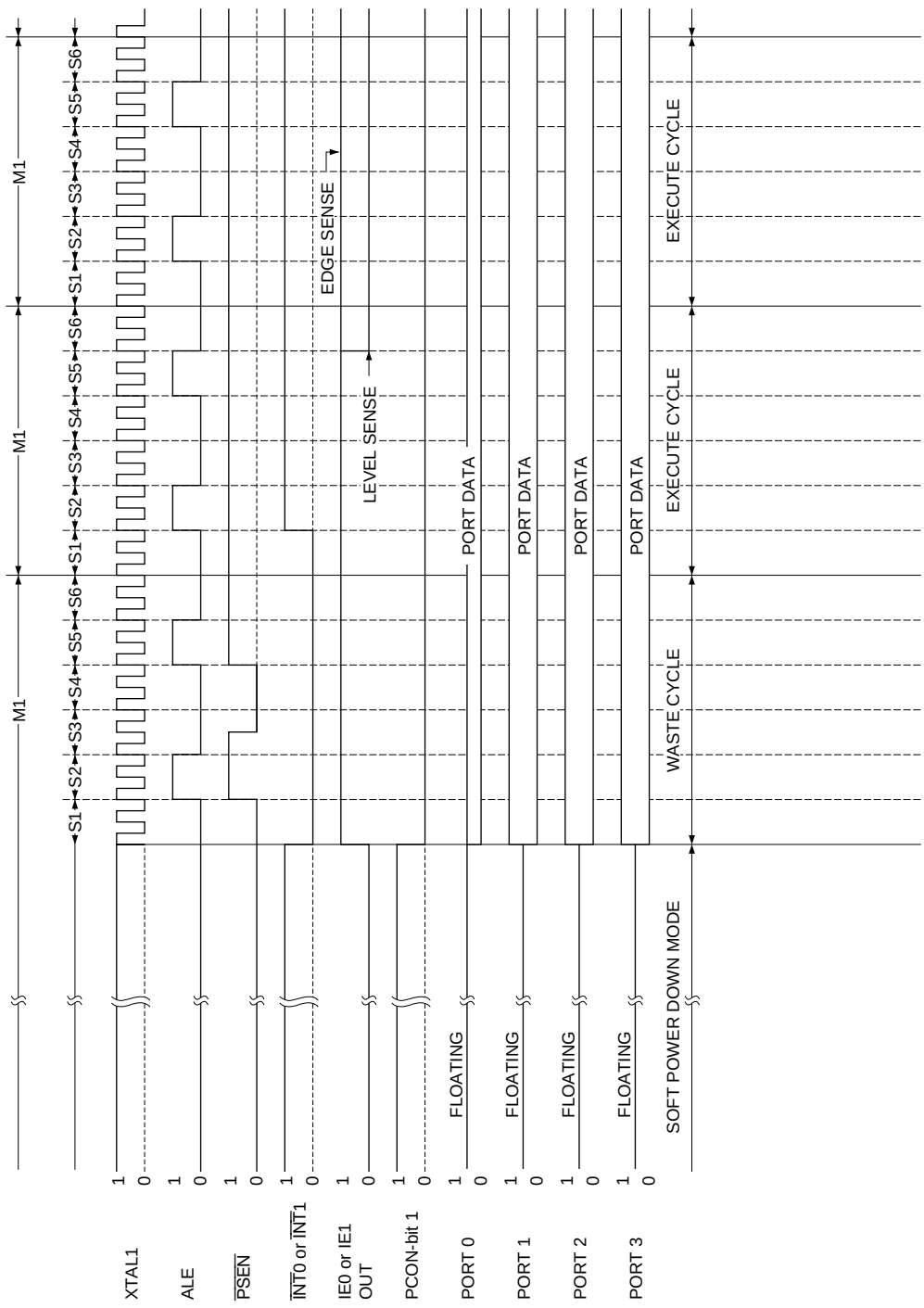


Figure 4-83 Restart from soft power down mode by $\overline{\text{INT}}$0 or 1 (internal ROM mode)

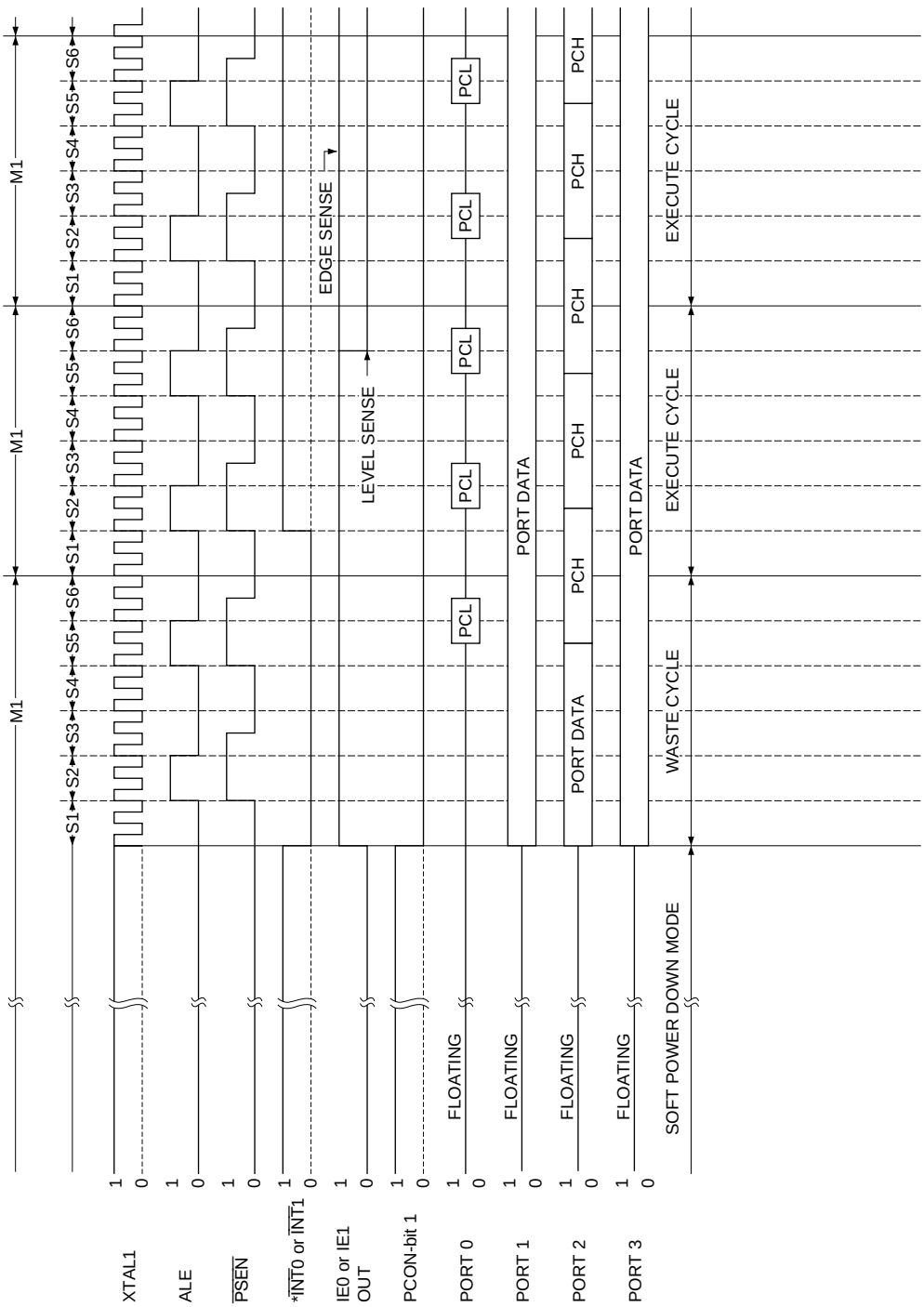


Figure 4-84 Restart from soft power down mode by $\overline{\text{INT}}$0 or 1 (external ROM mode)

4.10 MSM80C154S/83C154S Battery Backup with Hard Power Down Mode

Figures 4-85-1/2 and 2/2 show the examples of the MSM80C154S/83C154S battery backup circuits with hard power down mode. The hard power down mode serves to retain data stored in the CPU and external RAM if the AC 100V power failure occurs. Figure 4-85-1/2 shows the CPU, power failure detector, and external RAM control unit. Figure 4-85-2/2 shows the external RAM. The power failure detection voltage is set up by VR of the circuit A of Figure 4-85-1/2.

If the AC 100V power failure occurs when the power failure detection voltage is 4.5V, the circuit works as described below.

When the power failure occurs, the internal power supply voltage VCA goes down from 5V to 0V. When the VCA goes down less than 4.5V, a power failure detection signal is output from the A circuit to the B circuit.

If data is being transferred between the CPU and external RAM during the detection of power failure, information on power failure is stored in RS-F/F of the B circuit, when data transfer ends. When information on on power failure is stored in RS-F/F, the I/O control signal goes from “1” level to “0” level, which separates the external RAM and the peripheral circuit electrically to retain data in the external RAM. At the same time, a hard power down signal is output, the T1 pin of the CPU goes from “1” level to “0” level, and the CPU enters the hard power down mode.

If the I/O port is ready to output data during hard power down mode, electric current flows to the external via a 100KW pull-up resistance of the T1 pin.

The current flow to the external can be prevented by setting “1” into bit 0 (ALF) of IOCON (0F8H) when setting the hard power down mode. If the hard power down mode is set when ALF is at “1” level, electric current does not flow from the T1 pin to the external because I/O becomes a floating state.

When AC 100V power supply is restored and the internal VCA goes from 0V to 5V, the hard power down mode is cancelled.

When VCA exceeds 4.5V, the A circuit stops outputting a power failure signal for the B circuit. When a power failure signal is not output, the power failure memory RS-F/F of the B circuit is reset after a time constant of the internal 200W and 10mF, and the external RAM I/O control signal and hard power down signal turn from “0” level to “1” level.

When RS-F/F is reset, a CPU reset signal is output and the CPU’s power down mode is cancelled. The CPU starts the operation of XTAL1, 2 and executes a command starting from address 0.

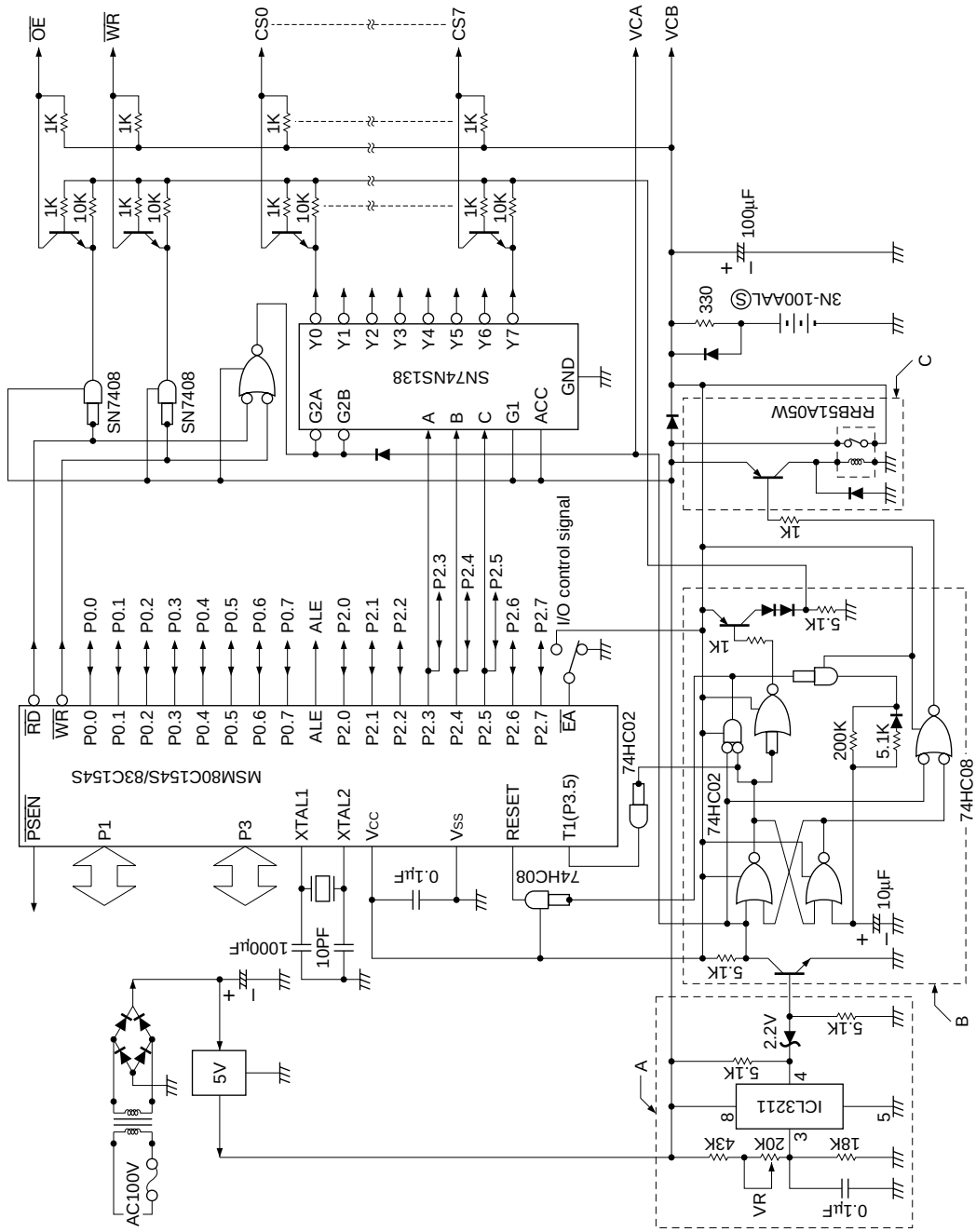


Figure 4-85-1/2 MSM80C154S/83C154S battery back up with hard power down mode

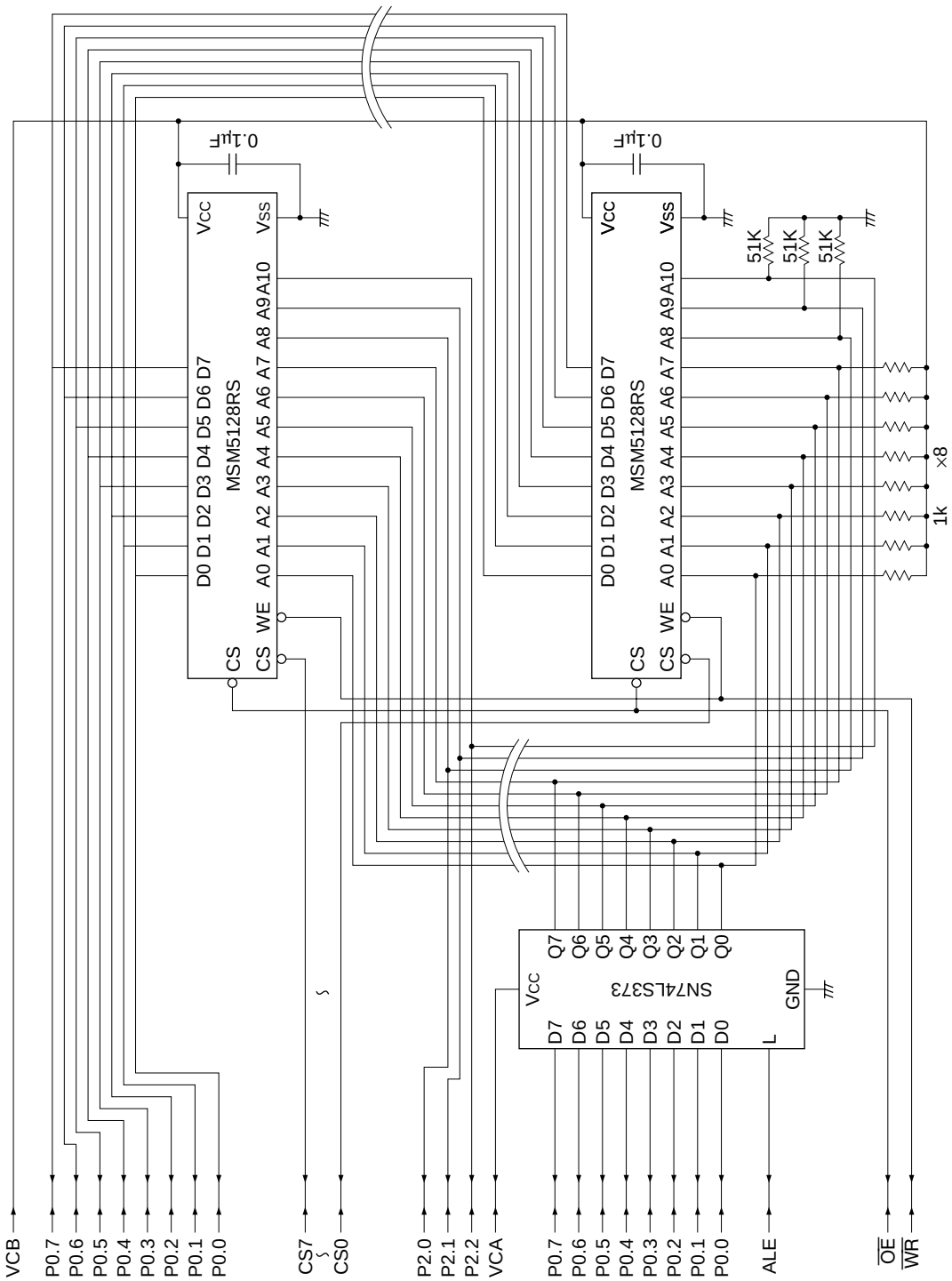


Figure 4-85-2/2 MSM80C154S/83C154S battery back up with hard power down mode

5. INPUT/OUTPUT PORTS

5. INPUT/OUTPUT PORTS

5.1 Outline

MSM80C154S/MSM83C154S is equipped with four 8-bit input/output ports. The functions of these four ports (port 0, 1, 2, and 3) are listed below.

1) Port 0: Input/output bus port, address output port, and data input/output port.
2) Port 1: Quasi-bidirectional input/output port and control input pin.
3) Port 2: Quasi-bidirectional input/output port and address output port.
4) Port 3: Quasi-bidirectional input/output port and control input/output pin.

5.2 Port 0

Port 0 is an 8-bit input/output port. The circuit configuration is shown in Figure 5-1. When port 0 is used as an input/output port in internal ROM mode (MSM83C154S), the equivalent circuit is indicated in Figure 5-2. When operated as an output port, port 0 becomes an open drain output port, and when operated as an input port, “1” should be set in the port 0 latch to put the port 0 pin into floating status prior to using the port for input purposes.
When port 0 is used in external ROM mode (MSM80C154S) and external RAM mode, the equivalent circuit is shown in Figure 5-3 where addresses and data outputs are obtained as “1” and “0” by totem pole output driver. When data from external ROM or external RAM is applied as input data, port 0 automatically becomes a tri-state input port. When the CPU is reset or when an external ROM or external RAM is accessed, “1” data is set automatically in the port 0 latch. The port 0 pin table is shown in Table 5-1.

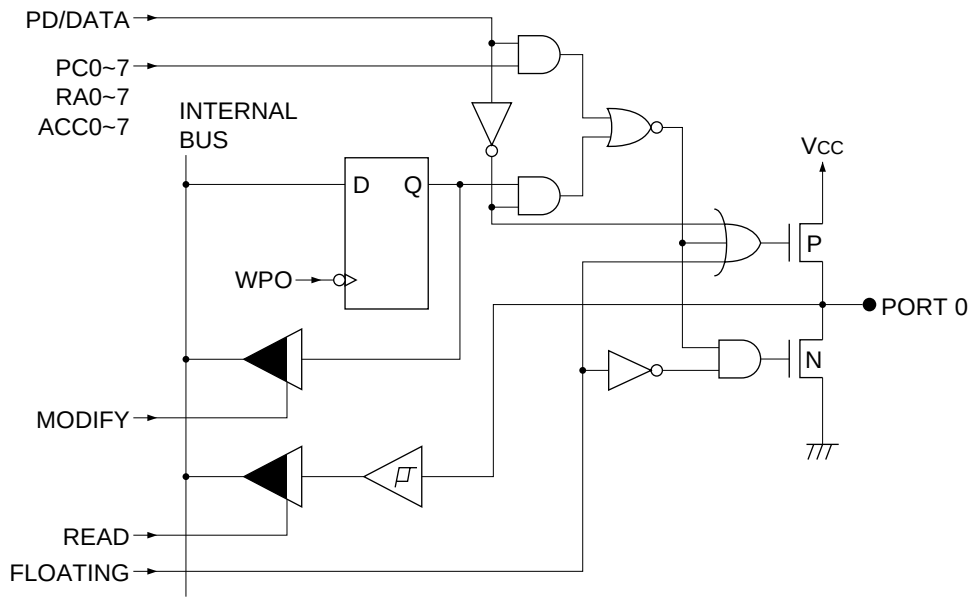


Figure 5-1 Port 0 internal equivalent circuit

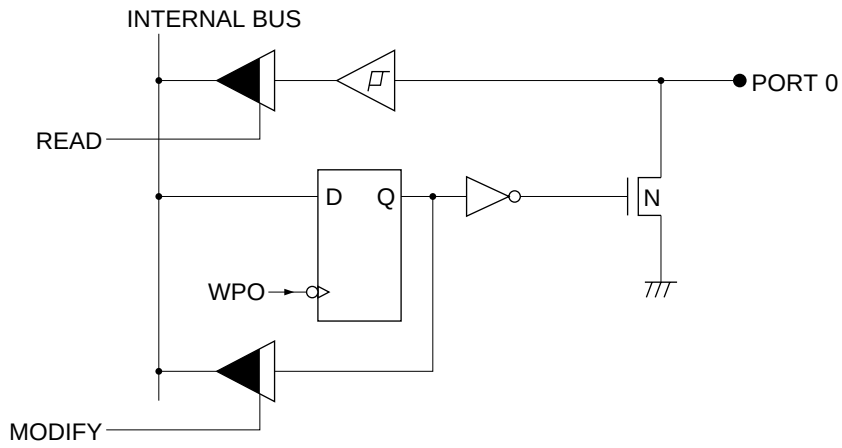


Figure 5-2 Port 0 input/Output port equivalent circuit in internal ROM mode

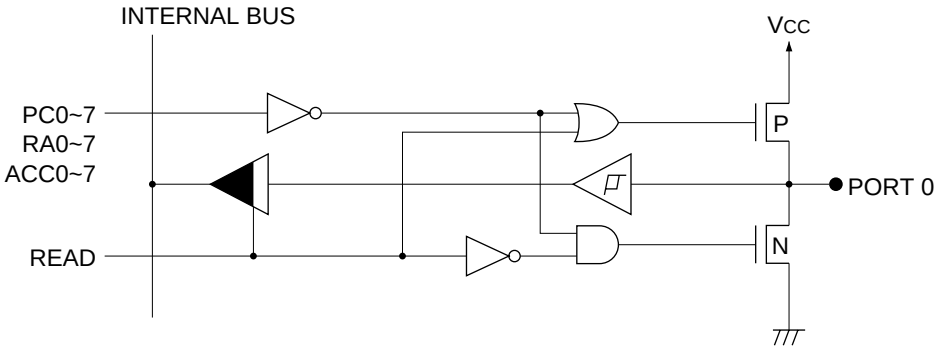


Figure 5-3 Port 0 equivalent circuit during address and data input/output in external ROM/RAM mode

Table 5-1 Port 0 pin table

	PORT0	Accumulator bit	Address
1	P0.0	ACC.0	PCRA –0
2	P0.1	ACC.1	PCRA –1
3	P0.2	ACC.2	PCRA –2
4	P0.3	ACC.3	PCRA –3
5	P0.4	ACC.4	PCRA –4
6	P0.5	ACC.5	PCRA –5
7	P0.6	ACC.6	PCRA –6
8	P0.7	ACC.7	PCRA –7

5.3 Port 1

Port 1 is a quasi-bidirectional port capable of handling input and output of 8-bit data in the circuit configuration outlined in Figure 5-4.
A “quasi-bidirectional port” refers to a port which has internal pull-up resistance when used as an input port. The internal equivalent circuit is shown in Figure 5-5.
If a quasi-bidirectional port is used exclusively as an output port, the port output driver becomes a totem-pole type for driving “1” and “0” data. The output impedance during output of “1” data is approximately 9 kohm, while a sink current is 1.6mA during output of “0” data. When used as an output port, the “1” data accelerator circuit is activated for a period equivalent to two XTAL1·2 oscillator clocks only when the output data is shifted from “0” to “1”. During this data acceleration operation, the “1” output impedance is changed to about 500 ohms, the I_{OH} current is increased, and the output signal leading edge is speeded up. The accelerator circuit operation time chart is given in Figure 5-6. Once port output data has been written in port latch it is preserved until output of the next item of data.
If a quasi-bidirectional port is used exclusively as an input port, “1” data is first set in the port latch in advance. When the input signal applied to the input port is changed from level “1” to level “0”, the port 10 kohm pull-up resistance is disconnected from the V_{CC}, leaving only the 100 kohm pull-up resistance for reducing external IIL current. And when the input signal is changed from level “0” to level “1”, the 10 kohm resistance is reconnected, thereby connecting the 10 and 100 kohm resistances to the V_{CC} supply in parallel. The quasi-bidirectional port input equivalent circuit is outlined in Figure 5-7.
To change port 1 from a quasi-bidirectional input port to a high impedance input port, “1” is set in bit 1 (P1HZ) of the I/O control register (IOCON 0F8H). The output driver circuit is thus disconnected from the port pin and the port becomes a high impedance input port. The signal levels applied to high impedance input ports are normal “0” and “1” level signals. The pins cannot be used in open status.
The bit 0 and bit 1 of port 1 have alternate functions apart from serving as port pins. Bit 0 can function as the external clock input pin for timer/counter 2, and bit 1 can function as the capture signal input pin for timer/counter 2, or as the auto reload signal input pin, or as the external timer flag 2 setting pin, depending on the timer/counter 2 operation mode.
When the bit 0 and 1 pins are to be used as timer/counter 2 control pins, “1” must be set in the port in advance.
And if port output is to be put into floating status during CPU power down mode (PD, HPD), “1” is to be set in bit 1 (ALF) of the I/O control register (IOCON 0F8H) before CPU power down mode is activated. Floated port 1 pins may be either open, or undefined within the –0.5 to V_{CC} +0.5V range.
And when port 1, 2, and 3 quasi-bidirectional ports are used as input ports, the port pull-up resistance may be set only to 100 kohms. If “1” is set in bit 4 (IZC) of the I/O control register (IOCON 0F8H), the 10 kohm pull-up resistance for ports 1, 2, and 3 is all disconnected from V_{CC}, leaving only the 100 kohm resistance. This mode is useful when input data is applied to the quasi-bidirectional port by external devices having low output driving capacity (high output impedance). The port 1 CPU control pin functions are listed in Table 5-2, and the port pin list is given in Table 5-3.

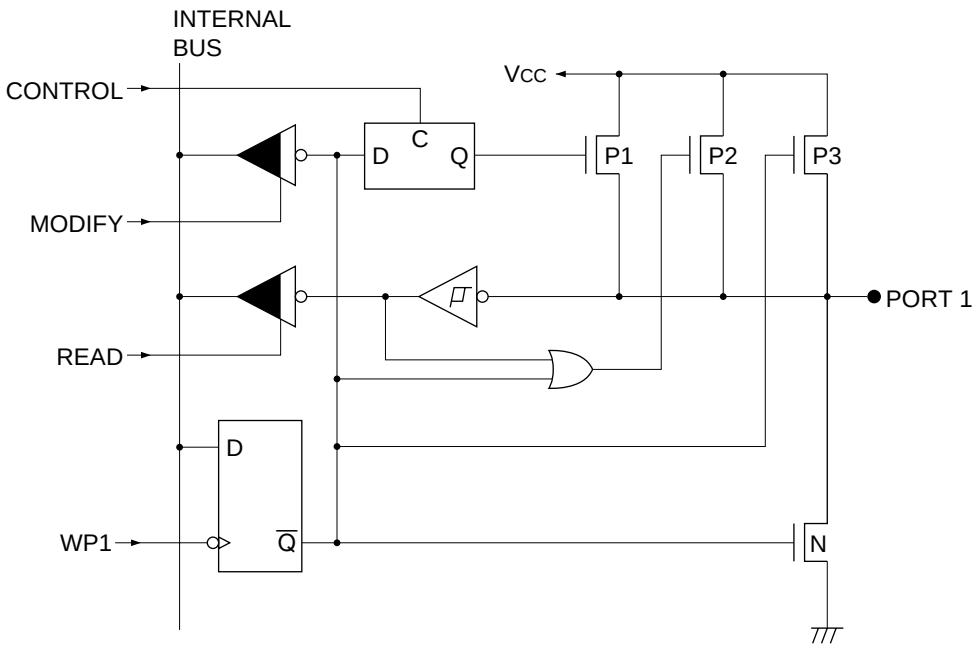
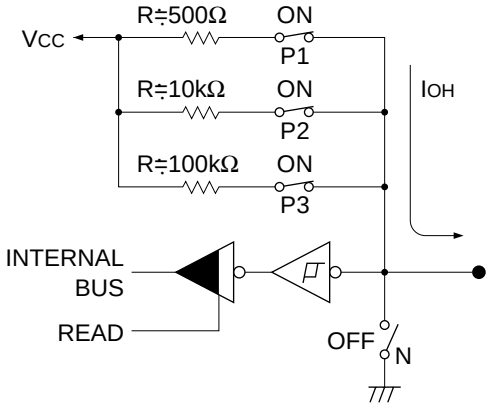
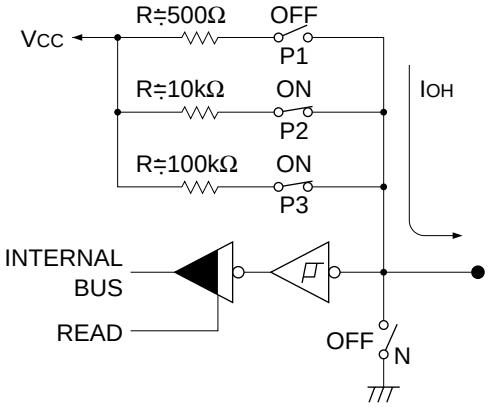


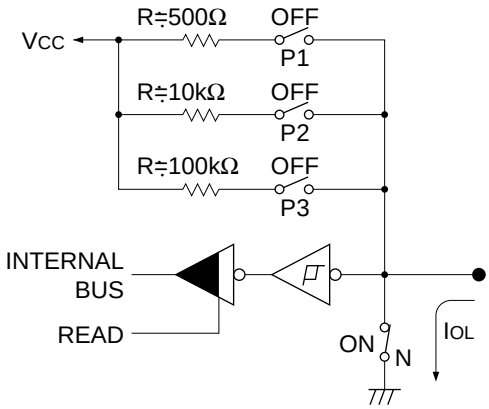
Figure 5-4 Port 1 internal equivalent circuit



(A) When accelerator circuit is activated



(B) When "1" data is held



(C) When "0" data is held

Figure 5-5 Quasi-bidirectional port equivalent circuit

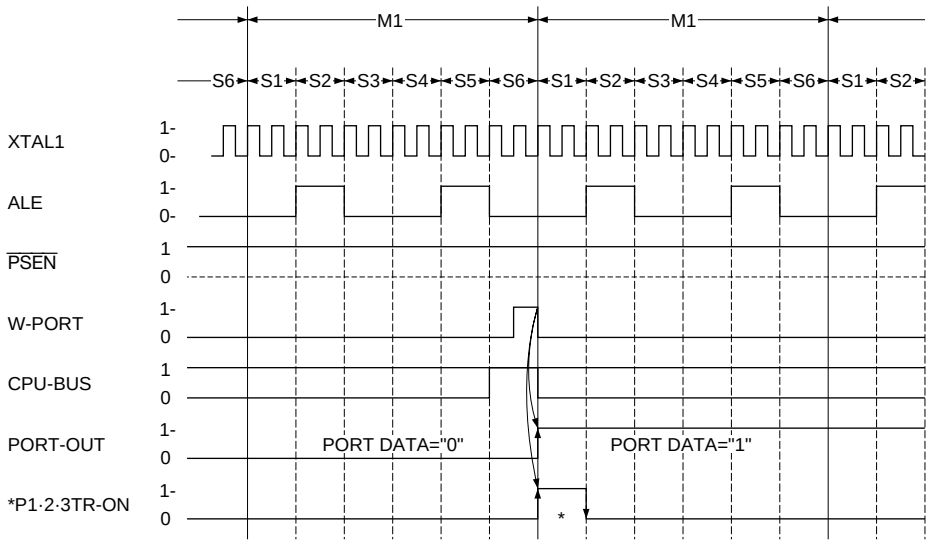
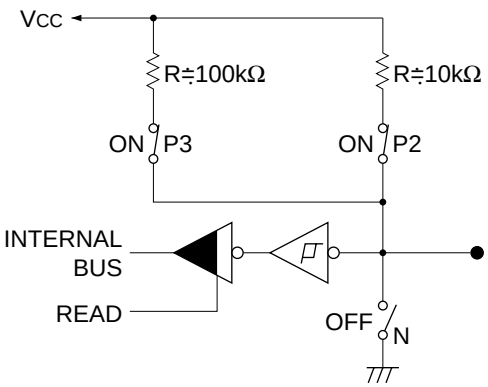
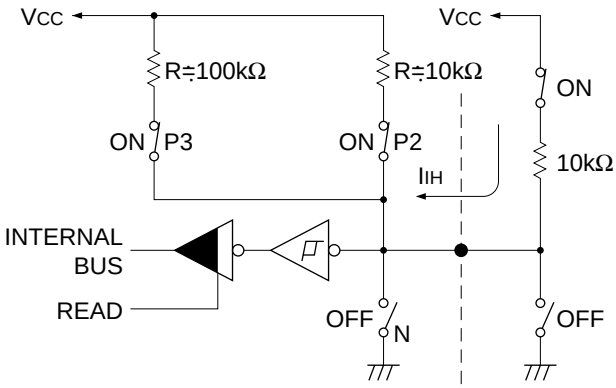


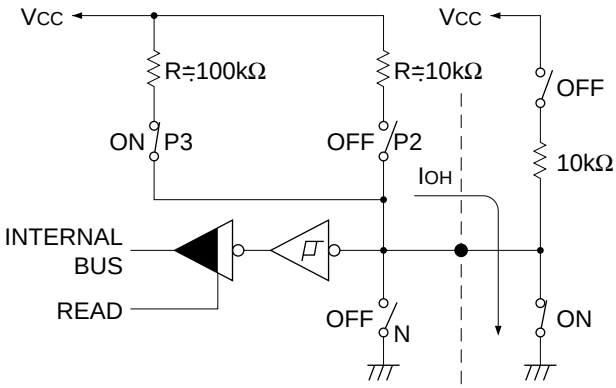
Figure 5-6 Quasi-bidirectional port accelerator circuit operation time chart



(A) "1" data writing equivalent circuit



(B) "1" data input equivalent circuit



(C) "0" data input equivalent circuit

Figure 5-7 Quasi-bidirectional port input equivalent circuit

Table 5-2 Port 1 CPU control pin table

PORT1	Function
P1.0	T2 [TIMER COUNTER 2 EXTERNAL CLOCK]
P1.1	T2EX [TIMER COUNTER 2 EXTERNAL CONTROL]

Table 5-3 Port 1 pin table

	PORT1	Accumulator bit
1	P1.0	ACC.0
2	P1.1	ACC.1
3	P1.2	ACC.2
4	P1.3	ACC.3
5	P1.4	ACC.4
6	P1.5	ACC.5
7	P1.6	ACC.6
8	P1.7	ACC.7

5.4 Port 2

Port 2 can function as a quasi-bidirectional port capable of handling input and output of 8-bit data in the circuit configuration outlined in Figure 5-8. It can also be used for output of addresses 8 thru 15 in external ROM and external RAM (using DPTR) modes. When port 2 is used as a quasi-bidirectional port, it functions in much the same way as port 1. Note, however, that the port 2 “1” data accelerator circuit operates for a period equivalent to four XTAL1·2 oscillator clocks.

Output of addresses 8 thru 15 obtained from port 2 is activated by the circuit outlined in Figure 5-9. When the address output data is “1”, the “1” data accelerator circuit is activated during output of the data, resulting in a higher driving capacity.

To change port 2 from a quasi-bidirectional input port to a high impedance input port, “1” is set in bit 2 (P2HZ) of the I/O control register (IOCON 0F8H). The output driver circuit is thus disconnected from the port pin and the port becomes a high impedance input port. The signal levels applied to high impedance input ports are normal “0” and “1” level signals. The pins cannot be used in open status.

When port outputs are floated in CPU power down mode (PD, HPD), the port 2 pins may be either open, or undefined within the –0.5 to VCC+0.5V range. The port 2 pin table is given in Table 5-4.

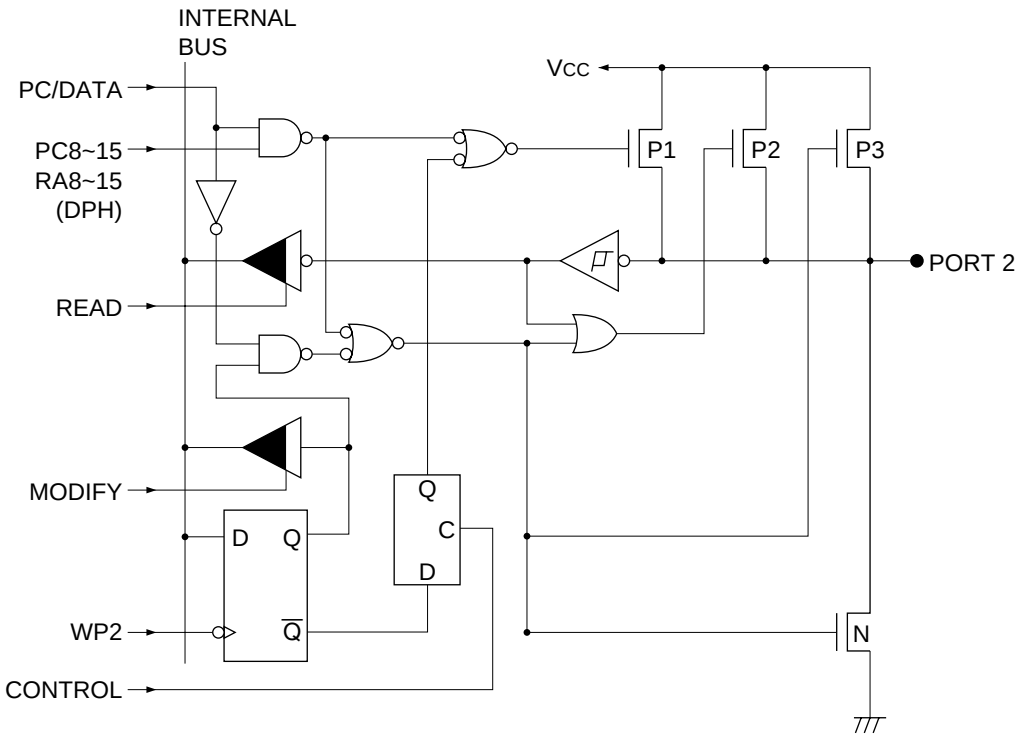


Figure 5-8 Port 2 internal equivalent circuit

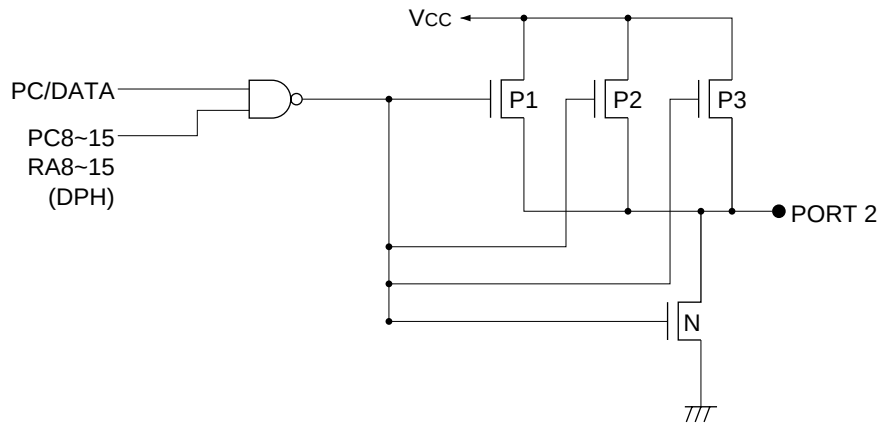


Figure 5-9 Port 2 address output equivalent circuit for external memory

Table 5-4 Port 2 pin table

	PORT2	Accumulator bit	Address
1	P2.0	ACC.0	PC/RA –8
2	P2.1	ACC.1	PC/RA –9
3	P2.2	ACC.2	PC/RA –10
4	P2.3	ACC.3	PC/RA –11
5	P2.4	ACC.4	PC/RA –12
6	P2.5	ACC.5	PC/RA –13
7	P2.6	ACC.6	PC/RA –14
8	P2.7	ACC.7	PC/RA –15

5.5 Port 3

Port 3 can function as a quasi-bidirectional port capable of handling input and output of 8-bit data in the circuit configuration outlined in Figure 5-10, and can also be used as a CPU control pin.

When port 3 is used as a quasi-bidirectional port, all functions are identical to those described for port 1. And when used as a CPU control pin, the port is used after first setting “1” data in the port latch. Note that if the port is used with “0” port latch data, the CPU control signal is ANDed (logical product) with the port “0” data, resulting in the CPU control signal remaining at “0” level.

To change port 3 from a quasi-bidirectional input port to a high impedance input port, “1” is set in bit 3 (P3HZ) of the I/O control register (IOCON 0F8H). The output driver circuit is thus disconnected from the port pin (floating pin status) and the port becomes a high impedance input port. The signal levels applied to high impedance input ports are normal “0” and “1” level signals. The pins cannot be used in open status.

When port outputs are floated in CPU power down mode (PD, HPD), normal “0” and “1” level signals are applied to pins 2 thru 5 of port 3, and pins 0, 1, 6, and 7 may be either open, or undefined within the –0.5 to V_{CC}+0.5V range. The CPU control function pins are listed in Table 5-5, and the port 3 pin table is given in Table 5-6.

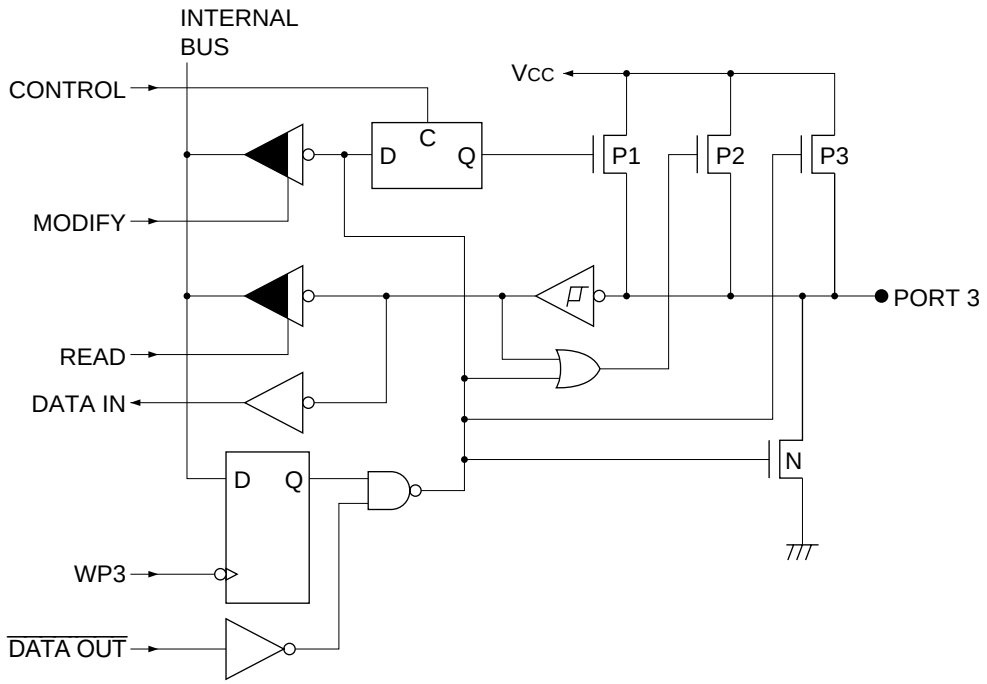


Figure 5-10 Port 3 internal equivalent circuit

Table 5-5 Port 3 CPU control pin function table

PORT3	PORT 3 PIN ALTERNATE FUNCTION
P3.0	RXD [SERIAL INPUT PORT]
P3.1	TXD [SERIAL OUTPUT PORT]
P3.2	$\overline{\text{INT}}$0 [EXTERNAL INTERRUPT 0]
P3.3	$\overline{\text{INT}}$1 [EXTERNAL INTERRUPT 1]
P3.4	T0 [TIMER/COUNTER 0 CLOCK]
P3.5	T1 [TIMER/COUNTER 1 CLOCK] HPDI [HARD POWER DOWN INPUT]
P3.6	$\overline{\text{WR}}$ [EXTERNAL DATA MEMORY WRITE STROBE]
P3.7	$\overline{\text{RD}}$ [EXTERNAL DATA MEMORY READ STROBE]

Table 5-6 Port 3 pin table

	PORT3	Control	Accumulator bit
1	P3.0	RXD	ACC.0
2	P3.1	TXD	ACC.1
3	P3.2	$\overline{\text{INT}}$0	ACC.2
4	P3.3	$\overline{\text{INT}}$1	ACC.3
5	P3.4	T0	ACC.4
6	P3.5	T1/HDPI	ACC.5
7	P3.6	$\overline{\text{WR}}$	ACC.6
8	P3.7	$\overline{\text{RD}}$	ACC.7

5.6 Port 0, 1, 2, and 3 Output and Floating Status Settings in CPU Power Down Mode (PD, HPD)

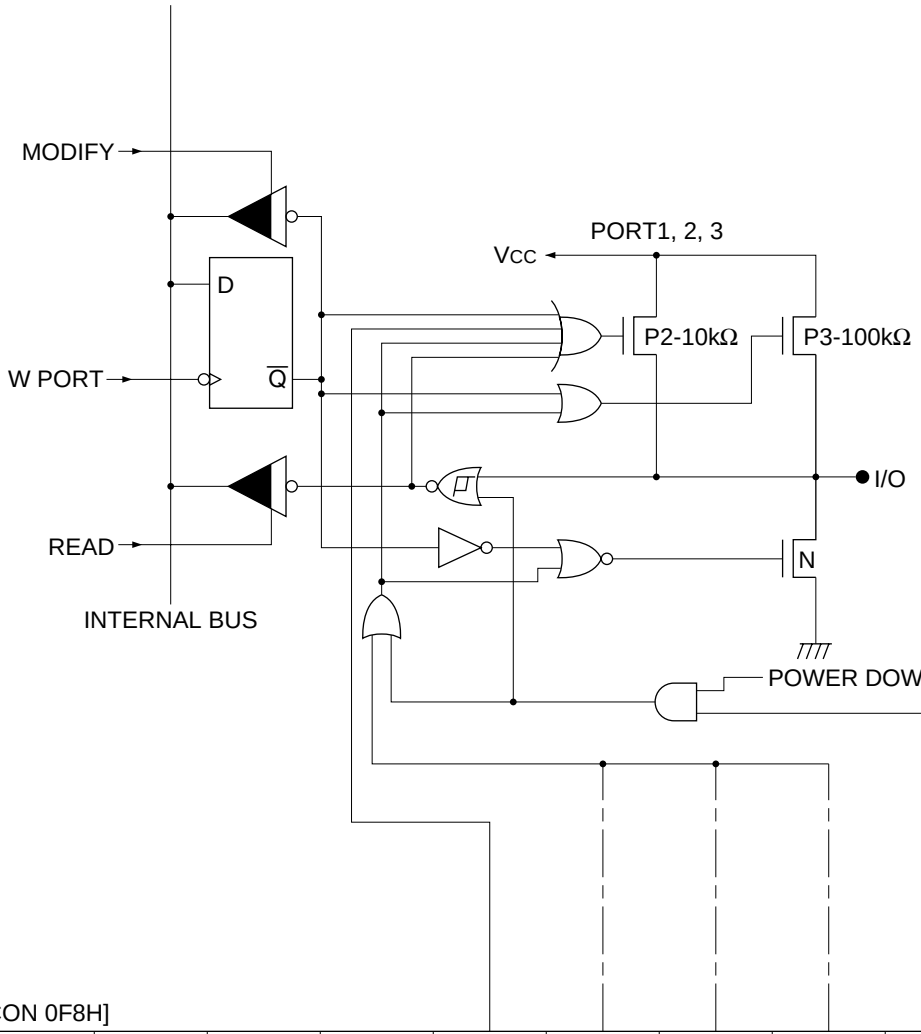
The port 0, 1, 2, and 3 output status can be set to either data output or floating when MSM80C154S/MSM83C154S is in power down mode (PD, HPD).

To set these ports to output status in power down mode, bit 0 (ALF) of the I/O control register (IOCON 0F8H) is reset to "0" before PD or HPD mode is activated (see Figure 5-11). The CPU is then stopped with the ports in data output status when power down mode is started.

And to set the ports to floating status in power down mode, "1" is set in bit 0 (ALF) of the I/O control register (IOCON 0F8H) before PD or HPD mode is activated (see Figure 5-11). The port output driver is disconnected from the port pins when power down mode is started.

If "1" output from port becomes a power supply factor in respect to the external circuits when PD or HPD mode is activated in port data output mode, the PD or HPD mode should be activated after the port data is reset to "0" by software. And in the reverse case, PD or HPD mode is activated after the port data is set to "1".

When port pins are in floating status during PD or HPD mode, the port pin status of all pins except pins 2 thru 5 of port 3 may be either open or undefined in the –0.5 to V_{CC}+0.5V range. This mode is used only in battery back-up of CPU data.



[IOCON 0F8H]

Bit	7	6	5	4	3	2	1	0
Flag	—	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF
Set				•	•	•	•	•

Figure 5-11 Control circuit for ports 0, 1, 2, 3 by IOCON

5.7 High Impedance Input Port Setting of Each Ouasi-bidirectional Port 1, 2, and 3

Each of the quasi-bidirectional input ports 1, 2, and 3 can be set as high impedance input ports.
This high impedance condition is achieved by setting “1” in bits 1 (P1HZ), 2 (P2HZ), and 3 (P3HZ) of the I/O control register (IOCON 0F8H) shown in Figure 5-11. Port 1 is set by P1HZ, port 2 by P2HZ, and port 3 by P3HZ. When the each bit is set to “1”, the port output driver is disconnected from the port pins, and the quasibidirectional input ports become high impedance input ports.
After being changed to high impedance input ports, the port latch data modify instructions and the input instructions for external input signals can still be used.
Normal “0” and “1” level signals must be applied to high impedance input ports. The pins cannot be used in open status.

5.8 100 kohm Pull-Up Resistance Setting for Quasi-bidirectional Input Ports 1, 2, and 3

Another of the MSM80C154S/MSM83C154S functions disconnects the 10 kΩ pull-up resistance from the power supply V_{CC} in the parallel connection of 10/100 kΩ pull-up resistances to the quasi-bidirectional input ports.
In normal operations, the 10 kΩ pull-up resistance is disconnected from the V_{CC} power supply when the level of the signal applied to the quasi-bidirectional input port is changed from “1” to “0”, thereby reducing the external I_{IL} current because of the remaining only the 100 kΩ pull-up resistance.
When the level of the signal applied to the port is then changed from “0” to “1”, the 10 kΩ resistance is reconnected to V_{CC}, and the port is pulled up by the 10 and 100 kΩ resistances connected in parallel. The resultant pull-up resistance is about 9 kΩ and the effect of random “0” noise is suppressed. But where an external device with low driving capacity is used to apply a “0” level signal to a quasi-bidirectional input port, the driving current may not be enough to change the port level to “0”. To overcome this problem, the CPU has been designed to disconnect the 10 kΩ pull-up resistance from the power supply leaving only the 100 kΩ resistance. This enables devices with low driving capacity to drive the quasi-bidirectional input ports.
The pull-up resistance for all quasi-bidirectional input ports 1, 2, and 3 can be set to 100 kΩ by setting “1” in bit 4 (IZC) of the I/O control register (IOCON 0F8H) shown in Figure 5-11 to disconnect the 10 kΩ resistance from V_{CC}.

5.9 Precautions When Driving External Transistors by Ouasi-bidirectional Port Output Signals

The following points must be carefully considered when quasi-bidirectional ports are used to drive a transistor by the circuit shown in Figure 5-12.
Even though the CPU output in this circuit is at “1” level, the port output pin level may be clamped by the base-emitter voltage V_{BE} (0.7V) of an external NPN transistor, resulting in a pin level of “0”.

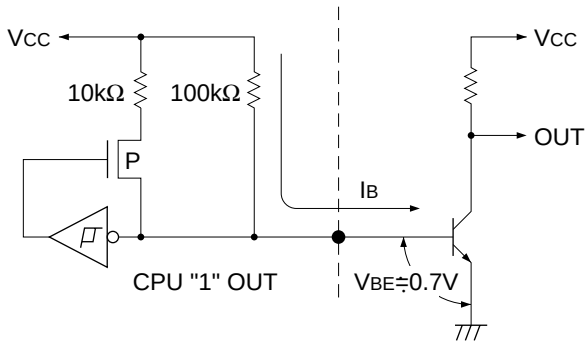


Figure 5-12 NPN transistor direct connection circuit

When the pin level is dropped to “0”, the CPU disconnects the 10 kΩ pull-up resistance from the power supply, leaving only the 100 kΩ pull-up resistance connected. Since the base current I_B of an external NPN transistor is supplied via the 100 kΩ resistance, the transistor collector current I_C may be reduced to a level insufficient for driving purposes.
To resolve this problem, diode can be inserted between the transistor base and CPU pin as indicated in Figure 5-13 to achieve a pin level of “1” by level shift. or by using a PNP transistor as indicated in Figure 5-14 where the external transistor is driven by a “0” level port output, this problem is solved.

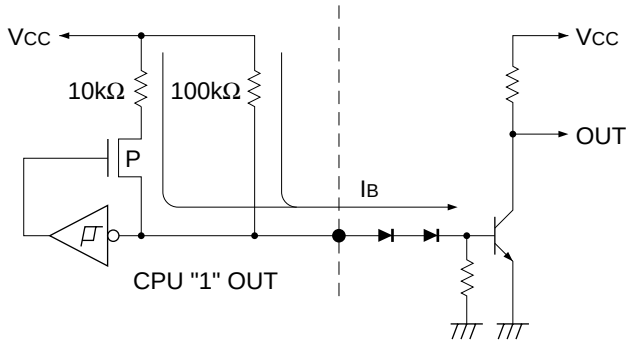


Figure 5-13 Drive circuit for NPN transistor by level shifter

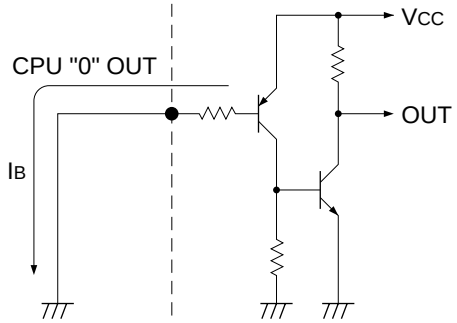
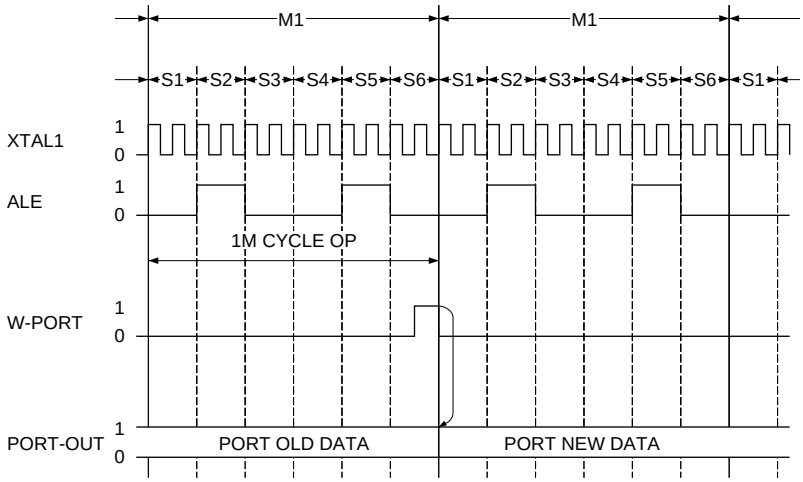


Figure 5-14 PNP transistor direct connection drive circuit

5.10 Port Output Timing

1) One machine cycle instruction output timing



INC data address

DEC data address

MOV data address, A

ORL data address, A

ANL data address, A

XRL data address, A

XCH A, data address

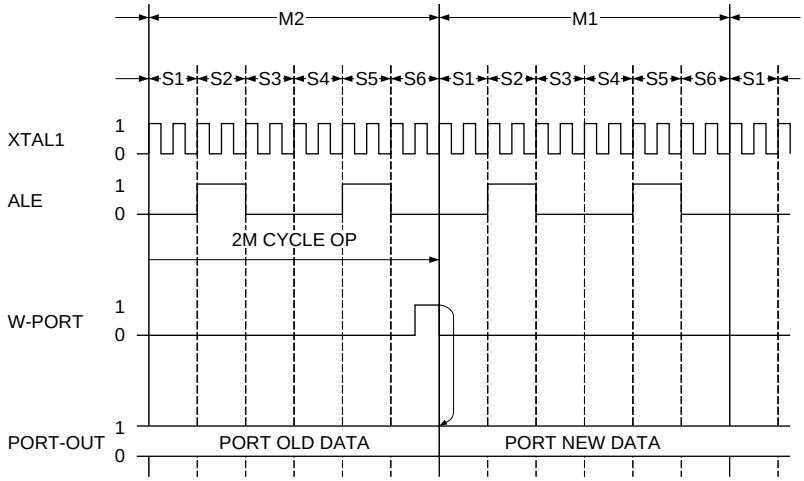
CPL bit address

CLR bit address

SETB bit address

Figure 5-15 One machine cycle instruction port output time chart

2) Two machine cycle instruction output timing



MOV data address, # data

ORL data address, # data

ANL data address, # data

XRL data address, # data

JBC bit address, code address

POP data address

MOV data address, @Rr

MOV data address, Rr

MOV data address 1, data address 2

MOV bit address, C

Figure 5-16 Two machine cycle instruction port output time chart

5.11 Port Data Manipulating Instructions

The MSM80C154S/MSM83C154S port operation instructions for ports 0, 1, 2, and 3 are divided into two groups-one where external signals applied to the port pin are used according to the instruction to be used, and the other where port latch data uneffected by the external signals is used. Instructions which use port latch data are listed below.

INC data address
DEC data address
ORL data address, # data
ANL data address, # data
XRL data address, # data
ORL data address, A
ANL data address, A
XRL data address, A
CPL bit address
JBC bit address, code address
DJNZ data address, code address
PUSH data address

6. ELECTRICAL CHARACTERISTICS

ELECTRICAL CHARACTERISTICS

6. ELECTRICAL CHARACTERISTICS

6.1 Absolute Maximum Ratings

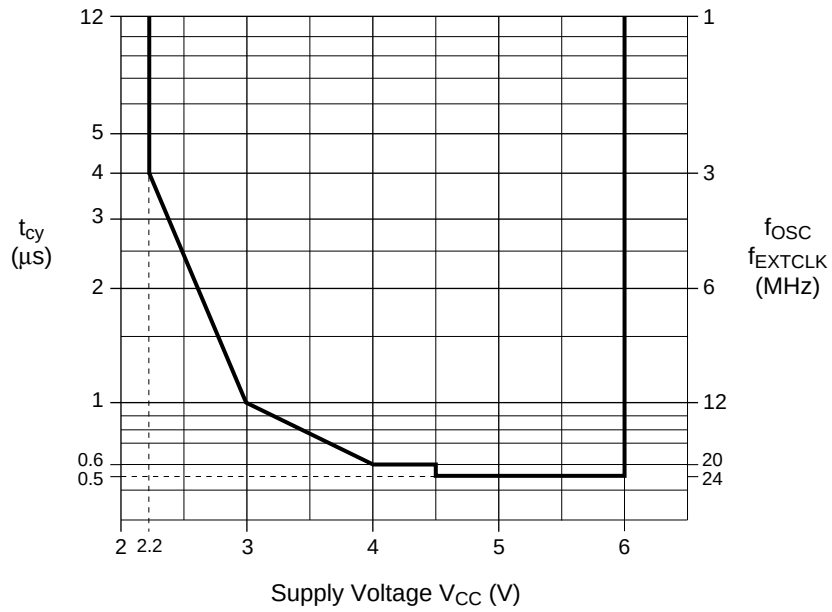
Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{CC}	Ta=25°C	–0.5~7	V
Input voltage	V_I	Ta=25°C	–0.5~V_{CC}+0.5	V
Storage temperature	T_{stg}	—	–55~+150	°C

6.2 Operational Ranges

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{CC}	See below	2.0~6	V
Memory hold voltage	V_{CC}	f_{OSC} = 0 Hz (Oscillation stop)	2~6	V
Oscillation frequency	f_{OSC}	See below	1~24*1	MHz
External clock operating frequency	f_{EXTCLK}	See below	0~24	MHz
Ambient temperature	Ta	—	–40~+85*2	°C

*1 Dpends on the specifications for the oscillator or ceramic resonator.
The MSM85C154HVS is guaranteed for operation at frequencies of up to 22 MHz.

*2 The MSM85C154HVS is guaranteed for operation at ordinary temperatures.



6.3 DC Characteristics 1

(V_{CC}=4.0 to 6.0V,V_{SS}=0V, Ta=–40°C to +85°C)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit	Measuring Circuit
Input Low Voltage	V_{IL}	—	–0.5	—	0.2V_{CC} –0.1	V	1
Input High Voltage	V_{IH}	Except XTAL1, $\overline{EA}$ and RESET	0.2V_{CC} +0.9	—	V_{CC}+0.5	V	
Input High Voltage	V_{IH1}	XTAL1 and $\overline{EA}$ RESET	0.7V_{CC}	—	V_{CC}+0.5	V	
Output Low Voltage [PORT 1, 2,3]	V_{OL}	I_{OL}=1.6mA	—	—	0.45	V	
Output Low Voltage [PORT 0, ALE, $\overline{PSEN}$]	V_{OL1}	I_{OL}=3.2mA	—	—	0.45	V	
Output High Voltage [PORT 1, 2,3]	V_{OH}	I_{OH}=–60µA	2.4	—	—	V	
		I_{OH}=–30µA	0.75V_{CC}	—	—	V	
		I_{OH}=–10µA	0.9V_{CC}	—	—	V	
Output High Voltage [PORT 0, ALE, $\overline{PSEN}$]	V_{OH1}	I_{OH}=–400µA V_{CC}=5V±10%	2.4	—	—	V	
		I_{OH}=–150µA	0.75V_{CC}	—	—	V	
		I_{OH}=–40µA	0.9V_{CC}	—	—	V	
Logical 0 Input Current/ logical 1 Output Current [PORT 1, 2,3]	I_{IL}/I_{OH}	V_I=0.45V/V_O=0.45V	–5	—	–80	µA	2
Logical 1 to 0 Transition Current [PORT 1, 2,3]	I_{TL}	V_I=2.0V	—	—	–500	µA	
Input Leakage Current [PORT 0 loating, $\overline{EA}$]	I_{LI}	V_{SS}<V_I<V_{CC}	—	—	±10	µA	3
RESET Pull-down Resistor	R_{RST}	—	20	40	125	kΩ	2
Pin Capacitance	C_{IO}	Ta=25°C, f=1MHz [except XTAL1]	—	—	10	pF	—
Power Down Current	I_{PD}	—	—	1	50	µA	4

Maximum Power Supply Current
Normal Operation I_{CC} (mA)

V_{CC}	4V	5V	6V
Freq.			
1MHz	2.2	3.1	4.1
3MHz	3.7	5.2	7.0
12MHz	12.0	16.0	20.0
16MHz	16.0	20.0	25.0
20MHz	19.0	25.0	30.0

V_{CC}	4.5V	5V	6V
Freq.			
24MHz	25.0	29.0	35.0

Maximum Power Supply Current
Idle Mode I_{CC} (mA)

V_{CC}	4V	5V	6V
Freq.			
1MHz	0.8	1.2	1.6
3MHz	1.2	1.7	2.3
12MHz	3.1	4.4	5.9
16MHz	3.8	5.5	7.3
20MHz	4.5	6.4	8.6

V_{CC}	4.5V	5V	6V
Freq.			
24MHz	6.4	7.4	9.8

DC Characteristics 2

(V_{CC}=2.2 to 4.0 V, V_{SS}=0 V, Ta=-40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit	Measuring circuit
Input Low Voltage	V_{IL}	—	–0.5	—	0.25 V_{CC}–0.1	V	1
Input High Voltage	V_{IH}	Except XTAL1, $\overline{EA}$, and RESET	0.25 V_{CC}+0.9	—	V_{CC}+0.5	V	
Input High Voltage	V_{IH1}	XTAL1, RESET, and $\overline{EA}$	0.6 V_{CC}+0.6	—	V_{CC}+0.5	V	
Output Low Voltage (PORT 1, 2, 3)	V_{OL}	I_{OL}=10 µA	—	—	0.1	V	
Output Low Voltage (PORT 0, ALE, $\overline{PSEN}$)	V_{OL1}	I_{OL}=20 µA	—	—	0.1	V	
Output High Voltage Output High Voltage (PORT 1, 2, 3)	V_{OH}	I_{OH}=–5 µA	0.75 V_{CC}	—	—	V	
(PORT 0, ALE, $\overline{PSEN}$)	V_{OH1}	I_{OH}=–20 µA	0.75 V_{CC}	—	—	V	
Logical 0 Input Current/ Logical 1 Output Current/ (PORT 1, 2, 3)	I_{IL} / I_{OH}	V_I=0.1 V / V_O=0.1 V	–5	—	–40	µA	2
Logical 1 to 0 Transition Output Current (PORT 1, 2, 3)	I_{TL}	V_I=1.9 V	—	—	–300	µA	
Input Leakage Current (PORT 0 floating, $\overline{EA}$)	I_{LI}	$V_{SS} < V_I < V_{CC}$	—	—	±10	µA	3
RESET Pull-down Resistance	R_{RST}	—	20	40	125	kΩ	2
Pin Capacitance	C_{IO}	Ta=25°C, f=1 MHz (except XTAL1)	—	—	10	pF	—
Power Down Current	I_{PD}	—	—	1	10	µA	4

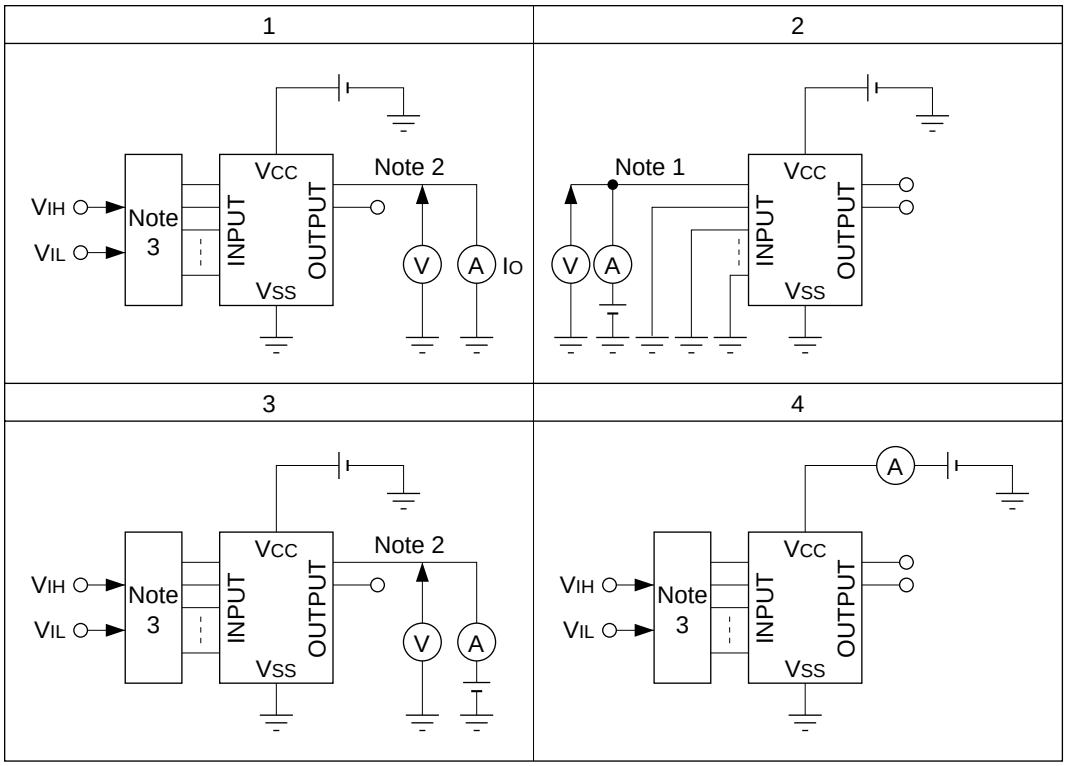
Maximum power supply current normal operation I_{CC} (mA)

V_{CC}	2.2 V	3.0 V	4.0 V
Freq			
1 MHz	0.9	1.4	2.2
3 MHz	1.8	2.4	4.3
12 MHz	—	8.0	12.0
16 MHz	—	—	16.0

Maximum power supply current idle mode I_{CC} (mA)

V_{CC}	2.2 V	3.0 V	4.0 V
Freq			
1 MHz	0.3	0.5	0.8
3 MHz	0.5	0.8	1.2
12 MHz	—	2.0	3.1
16 MHz	—	—	3.8

Measuring circuits



Note 1 : Repeated for specified input pins.
2 : Repeated for specified output pins.
3 : Input logic for specified status.

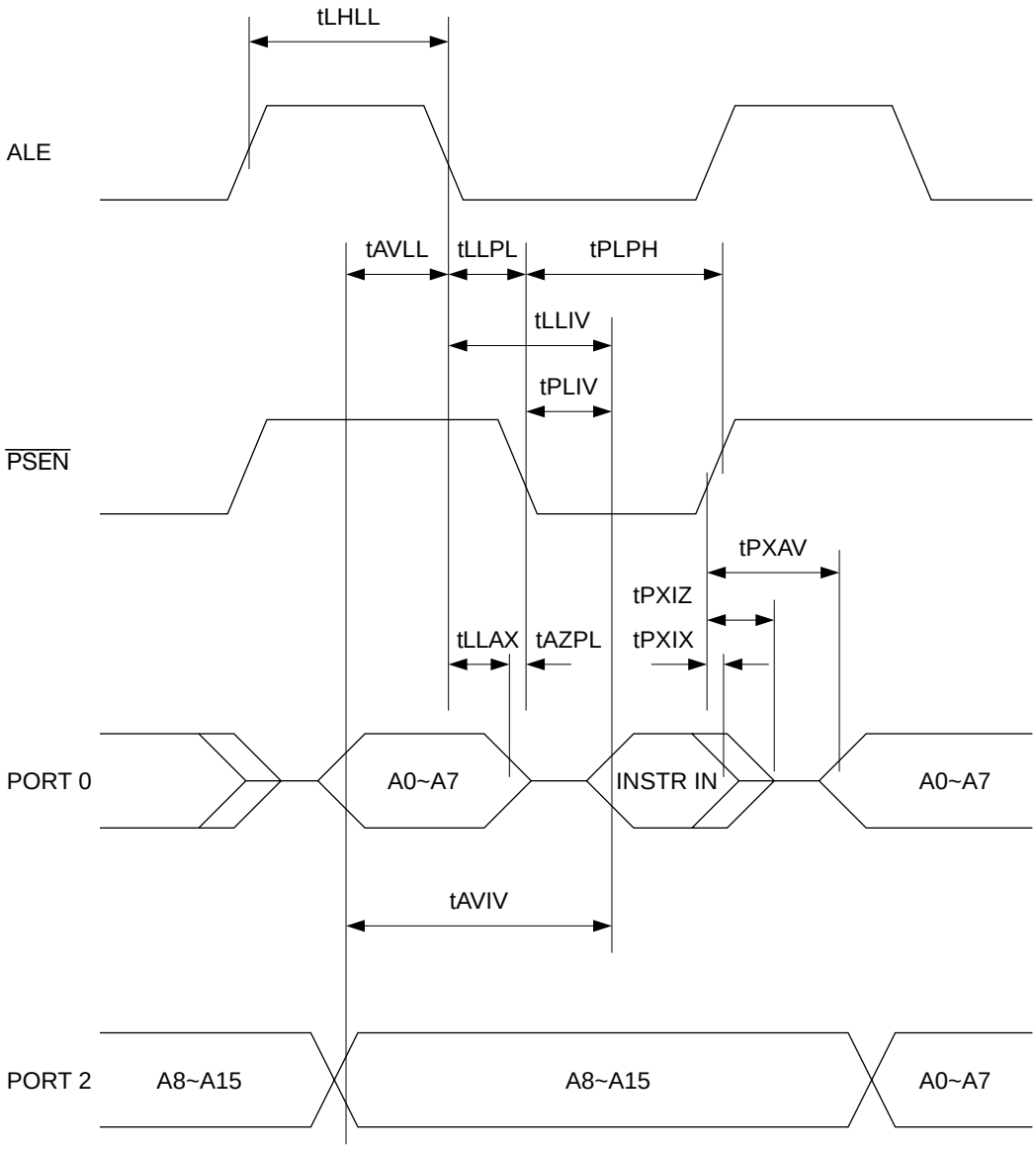
6.4 External Program Memory Access AC Characteristics

V_{CC}=2.2 to 6.0V, V_{SS}=0V, Ta=–40°C to +85°C
PORT 0, ALE, and $\overline{PSEN}$ connected with 100pF load, other connected with 80pF load

Parameter	Symble	Variable clock from*1 1 to 24 MHz		Unit
		Min.	Max.	
XTAL1, XTAL 2 Oscillation Cycle	t_{CLCL}	41.7	1000	ns
ALE Signal Width	t_{LHLL}	$2t_{CLCL}$-40	—	ns
Address Setup Time (to ALE Falling Edge)	t_{AVLL}	$1t_{CLCL}$-15	—	ns
Address Hold Time (from ALE Falling Edge)	t_{LLAX}	$1t_{CLCL}$-35	—	ns
Instruction Data Read Time (from ALE Falling Edge)	t_{LLPL}	—	$4t_{CLCL}$-100	ns
From ALE Falling Edge to $\overline{PSEN}$ Falling Edge	t_{LLPL}	$1t_{CLCL}$-30	—	ns
$\overline{PSEN}$ Signal Width	t_{PLPH}	$3t_{CLCL}$-35	—	ns
Instruction Data Read Time (from $\overline{PSEN}$ Falling Edge)	t_{PLIV}	—	$3t_{CLCL}$-45	ns
Instruction Data Hold Time (from $\overline{PSEN}$ Rising Edge)	t_{PXIX}	0	—	ns
Bus Floating Time after Instruction Data Read (from $\overline{PSEN}$ Rising Edge)	t_{PXIZ}	—	$1t_{CLCL}$-20	ns
Instruction Data Read Time (from Address Output)	t_{AVIV}	—	$5t_{CLCL}$-105	ns
Bus Floating Time($\overline{PSEN}$ Rising Edge from Address float)	t_{AZPL}	0	—	ns
Address Output Time from $\overline{PSEN}$ Rising Edge	t_{PXAV}	$1t_{CLCL}$-20	—	ns

*1 The variable check is from 0 to 24 MHz when the external check is used.

External program memory read cycle



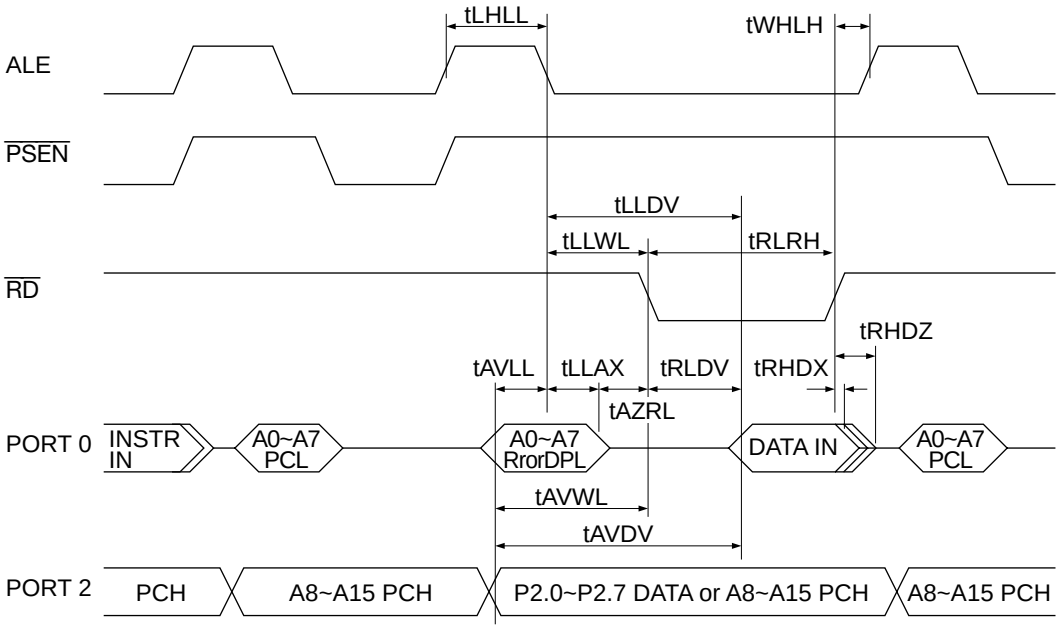
6.5 External Data Memory Access AC Characteristics

VCC=2.2 to 6.0V, VSS=0V, Ta=–40°C to +85°C
PORT 0, ALE, and $\overline{PSEN}$ connected with 100pF load, other connected with 80pF load

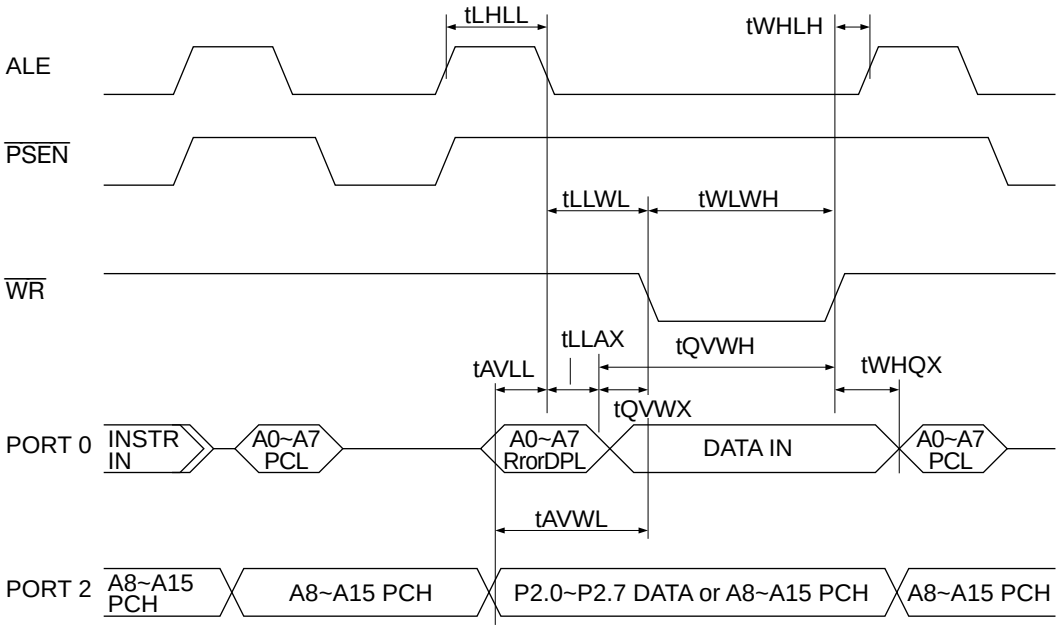
Parameter	Symbol	Variable clock from*1 1 to 24 MHz		Unit
		Min.	Max.	
XTAL1, XTAL2 Oscillator Cycle	t_{CLCL}	45.5	1000	ns
ALE Signal Width	t_{LHLL}	$2t_{CLCL}$-40	—	ns
Address Setup Time (to ALE Falling Edge)	t_{AVLL}	$1t_{CLCL}$-15	—	ns
Address Hold Time (from ALE Falling Edge)	t_{LLAX}	$1t_{CLCL}$-35	—	ns
$\overline{RD}$ Signal Width	t_{RLRL}	$6t_{CLCL}$-100	—	ns
$\overline{WR}$ Signal Width	t_{WLWH}	$6t_{CLCL}$-100	—	ns
RAM Data Read Time (from $\overline{RD}$ Signal Falling Edge)	t_{RLDV}	—	$5t_{CLCL}$-105	ns
RAM Data Read Hold Time (from $\overline{RD}$ Signal Rising Edge)	t_{RHDX}	0	—	ns
Data Bus Floating Time (from $\overline{RD}$ Signal Rising Edge)	t_{RHDZ}	—	$2t_{CLCL}$-70	ns
RAM Data Read Time (from ALE Signal Falling Edge)	t_{LLDV}	—	$8t_{CLCL}$-100	ns
RAM Data Read Time (from Address Output)	t_{AVDV}	—	$9t_{CLCL}$-105	ns
$\overline{RD}$/$\overline{WR}$ Output Time from ALE Falling Edge	t_{LLWL}	$3t_{CLCL}$-40 *2 $3t_{CLCL}$-100	$3t_{CLCL}$+40	ns
$\overline{RD}$/$\overline{WR}$ Output Time from Address Output	t_{AVWL}	$4t_{CLCL}$-70	—	ns
$\overline{WR}$ Output Time from Data Output	t_{QVWX}	$1t_{CLCL}$-40	—	ns
Time from Data to $\overline{WR}$ Rising Edge	t_{QVWH}	$7t_{CLCL}$-105	—	ns
Data Hold Time (from $\overline{WR}$ Rising Edge)	t_{WHQX}	$2t_{CLCL}$-50	—	ns
Time from to Address Float $\overline{RD}$ Output	t_{RLAZ}	0	—	ns
Time from $\overline{RD}$/$\overline{WR}$ Rising Edge to ALE Rising Edge	t_{WHLH}	$1t_{CLCL}$-30	$1t_{CLCL}$+40 *2 $1t_{CLCL}$+100	ns

*1 The variable check is from 0 to 24 MHz when the external check is used.
*2 For $2.2 \leq V_{CC} < 4$ V

External data memory read cycle



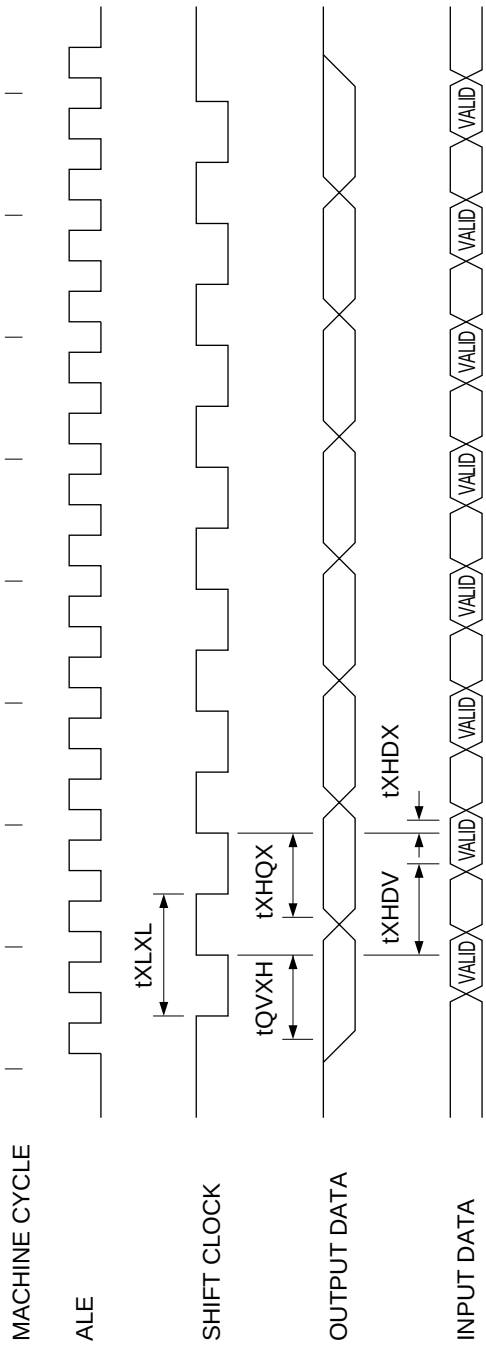
External data memory write cycle



6.6 Serial Port (I/O Extension Mode) AC Characteristics

V_{CC}=2.2 to.0V, V_{SS}=0V, Ta=–40°C to 85°C

Parameter	Symbol	Min	Max	Unit
Serial Port Clock Cycle Time	tXLXL	12tCLCL	—	ns
Output Data Setup to Clock Rising Edge	tQVXH	10tCLCL–133	—	ns
Output Data Hold After Clock Rising Edge	tXHQX	2tCLCL–75	—	ns
Input Data Hold After Clock Rising Edge	tXHDX	0	—	ns
Clock Rising Edge to Input Data Valid	tXHDV	—	10tCLCL–133	ns

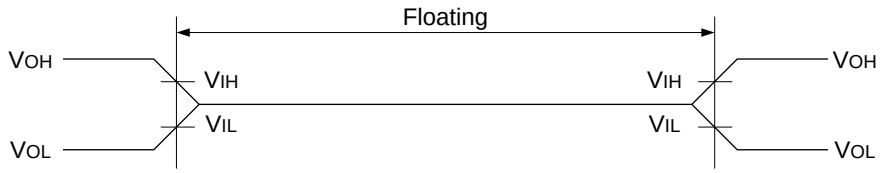

MACHINE CYCLE
ALE
SHIFT CLOCK
OUTPUT DATA
INPUT DATA
tXLXL
tQVXH
tXHQX
tXHDV
tXHDX
VALID

6.7 AC Characteristics Measuring Conditions

1. Input/output signal

* The input signals in AC test mode are either VOH (logic “1”) orVOL (logic “0”). Timing measurements are made atVIH (logic “1”) and VIL (10gic “0”).

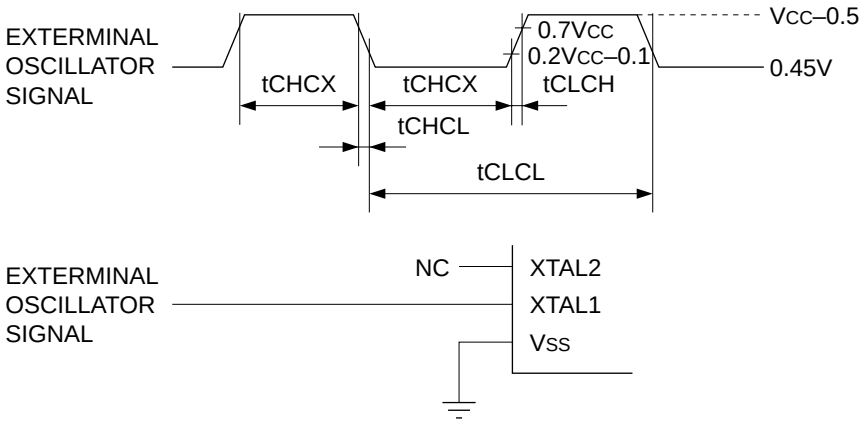
2. Floating



* The port 0 floating interval is measured from the time the port 0 pin Voltage drops below VIH after sinking to GND at 2.4mA when switching to floating status from a “1” output, and from the time the port 0 pin Voltage exceeds VIL after connecting to a 400μA source when switching to floating status from a “0” output.

6.8 XTAL1 External Clock Input Waveform Conditions

Parameter	Symbol	Min	Max	Unit
Oscillator Freq.	1/tCLCL	0	24	MHz
High Time	tCHCX	15	—	ns
Low Time	tCLCX	15	—	ns
Rise Time	tCLCH	—	5	ns
Fall Time	tCHCL	—	5	ns



7. DESCRIPTION OF INSTRUCTIONS

7. DESCRIPTION OF INSTRUCTIONS

7.1 Outline

MSM80C154S/MSM83C154S is a microcontroller designed for parallel processing in an 8-bit ALU. The instructions consist of 8-bit units of data, and are available as 1-word 1 - machine, 2-machine, and 4-machine cycle instructions as well as 2-word 1-machine and 2-machine cycle instructions and 3-word 2-machine cycle instructions. There is a total of 112 instructions classified into the following groups.

(1)	Arithmetic and logic instructions	(15)
(2)	Accumulator operation instructions	(7)
(3)	Increment & decrement instructions	(9)
(4)	Logical operation instructions	(18)
(5)	Immediate data setting instructions	(5)
(6)	Carry flag operation instructions	(7)
(7)	Bit transfer instructions	(3)
(8)	Bit manipulaton instructions	(3)
(9)	Data transfer instructions	(11)
(10)	Constant value instructions	(2)
(11)	Data exchange instructions	(4)
(12)	Subroutine instructions	(6)
(13)	Jump instructions	(4)
(14)	Branching instructions	(13)
(15)	External data memory instructions	(4)
(16)	Other instruction	(1)

7.2 Description of Instruction Symbols

The instruction symbols have the following meanings.

A	Accumulator
AB	Register pair
AC	Auxiliary carry
B	Arithmetic operation register
C	Carry (the bit 7 carry represented by CY is changed to C in Chapter 7.)
DPTR	Data pointer
PC	Program counter
Rr	Register representation (r=0/1, or r=0 thru 7)
SP	Stack pointer
AND	Logical AND
OR	Logical OR
XOR	Exclusive OR
+	Addition
–	Subtraction
×	Multiplication
/	Division
(X)	Representation of the contents of X
((X))	Representation of the contents addressed by contents of X
#	Symbol denoting immediate data
@	Symbol denoting indirect address
=	Equal sign
≠	Not equal
←	Substitution
→	Substitution
—	Negation (upper bar)
<	Smaller than
>	Larger than
bit address	RAM or special function register bit designated address
code address	Absolute address (A_0 thru A_{15}, A_0 thru A_{11})
data	Immediate data (I_0 thru I_7)
relative offset	Corrected relative jump address value
direct address	RAM or special function register data designated address (“direct address” representation changed to “data address” during detailed description of instructions)

7.3 List of Instructions

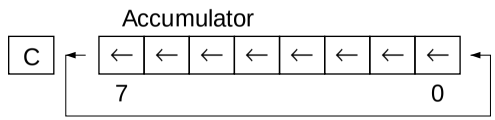
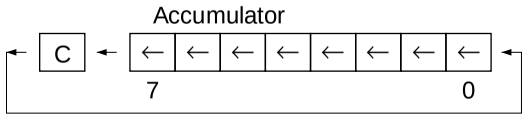
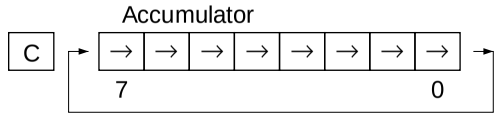
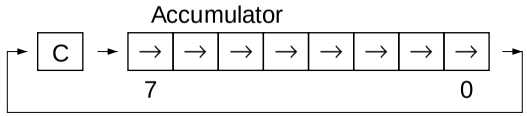
MSM80C154S/MSM83C154S instruction table

H \ L	0 0 0 0 0	1 0 0 0 1	2 0 0 1 0	3 0 0 1 1	4 0 1 0 0	5 0 1 0 1	6 0 1 1 0	7 0 1 1 1	8 1 0 0 0	9 1 0 0 1	A 1 0 1 0	B 1 0 1 1	C 1 1 0 0	D 1 1 0 1	E 1 1 1 0	F 1 1 1 1
0 0000	NOP	AJMP address 11 (Page 0)	LJMP address 16	RR A	INC A	INC direct	INC @R0	INC @R1	INC R0	INC R1	INC R2	INC R3	INC R4	INC R5	INC R6	INC R7
1 0001	JBC bit, rel	ACALL address 11 (Page 0)	LCALL address 16	RRC A	DEC A	DEC direct	DEC @R0	DEC @R1	DEC R0	DEC R1	DEC R2	DEC R3	DEC R4	DEC R5	DEC R6	DEC R7
2 0010	JB bit, rel	AJMP address 11 (Page 1)	RET	RL A	ADD A, #data	ADD A, direct	ADD A, @R0	ADD A, @R1	ADD A, R0	ADD A, R1	ADD A, R2	ADD A, R3	ADD A, R4	ADD A, R5	ADD A, R6	ADD A, R7
3 0011	JNB bit, rel	ACALL address 11 (Page 1)	RETI	RLC A	ADDC A, #data	ADDC A, direct	ADDC A, @R0	ADDC A, @R1	ADDC A, R0	ADDC A, R1	ADDC A, R2	ADDC A, R3	ADDC A, R4	ADDC A, R5	ADDC A, R6	ADDC A, R7
4 0100	JC bit, rel	AJMP address 11 (Page 2)	ORL direct , #data	ORL A, #data	ORL A, direct	ORL A, @R0	ORL A, @R1	ORLA, R0	ORLA, R1	ORLA, R2	ORLA, R3	ORLA, R4	ORLA, R5	ORLA, R6	ORLA, R7
5 0101	JNC rel	ACALL address 11 (Page 2)	ANL direct , #data	ANL A, #data	ANL A, direct	ANL A, @R0	ANL A, @R1	ANLA, R0	ANLA, R1	ANLA, R2	ANLA, R3	ANLA, R4	ANLA, R5	ANLA, R6	ANLA, R7
6 0110	JZ rel	AJMP address 11 (Page 3)	XRL direct , #data	XRL A, #data	XRL A, direct	XRL A, @R0	XRL A, @R1	XRLA, R0	XRLA, R1	XRLA, R2	XRLA, R3	XRLA, R4	XRLA, R5	XRLA, R6	XRLA, R7
7 0111	JNZ rel	ACALL address 11 (Page 3)	ORL C, bit	JMP @A+DPTR	MOV A, #data	MOV direct , #data	MOV @R0, #data	MOV @R1, #data	MOV R0, #data	MOV R1, #data	MOV R2, #data	MOV R3, #data	MOV R4, #data	MOV R5, #data	MOV R6, #data	MOV R7, #data
8 1000	SJMP rel	AJMP address 11 (Page 4)	ANL C, bit	MOVC A, @A+PC	DIV AB	MOV direct 1, direct 2	MOV direct, @R0	MOV direct, @R1	MOV direct, R0	MOV direct, R1	MOV direct, R2	MOV direct, R3	MOV direct, R4	MOV direct, R5	MOV direct, R6	MOV direct, R7
9 1001	MOV DPTR, #data 16	ACALL address 11 (Page 4)	MOV bit, C	MOVC A, @A+DPTR	SUBB A, #data	SUBB A, direct	SUBB A, @R0	SUBB A, @R1	SUBB A, R0	SUBB A, R1	SUBB A, R2	SUBB A, R3	SUBB A, R4	SUBB A, R5	SUBB A, R6	SUBB A, R7
A 1010	ORL C,/bit	AJMP address 11 (Page 5)	MOV C, bit	INC DPTR	MUL AB	—	MOV @R0, direct	MOV @R1, direct	MOV R0, direct	MOV R1, direct	MOV R2, direct	MOV R3, direct	MOV R4, direct	MOV R5, direct	MOV R6, direct	MOV R7, direct
B 1011	ANL C,/bit	ACALL address 11 (Page 5)	CPL bit	CPL C	CJNE A, #data, rel	CJNE A, direct, rel	CJNE@R0, #data, rel	CJNE@R1, #data, rel	CJNE R0, #data, rel	CJNE R1, #data, rel	CJNE R2, #data, rel	CJNE R3, #data, rel	CJNE R4, #data, rel	CJNE R5, #data, rel	CJNE R6, #data, rel	CJNE R7, #data, rel
C 1100	PUSH direct	AJMP address 11 (Page 6)	CLR bit	CLR C	SWAP A	XCH A, direct	XCH A, @R0	XCH A, @R1	XCHA, R0	XCHA, R1	XCHA, R2	XCHA, R3	XCHA, R4	XCHA, R5	XCHA, R6	XCHA, R7
D 1101	POP direct	ACALL address 11 (Page 6)	STEB bit	STEB C	DA A	DJNZ direct, rel	XCHD A, @R0	XCHD A, @R1	DJNZ R0, rel	DJNZ R1, rel	DJNZ R2, rel	DJNZ R3, rel	DJNZ R4, rel	DJNZ R5, rel	DJNZ R6, rel	DJNZ R7, rel
E 1110	MOVX A, @DPTR	AJMP address 11 (Page 7)	MOVX A, @R0	MOVX A, @R1	CLR A	MOV A, direct	MOV A, @R0	MOV A, @R1	MOVA, R0	MOVA, R1	MOVA, R2	MOVA, R3	MOVA, R4	MOVA, R5	MOVA, R6	MOVA, R7
F 1111	MOVX @DPTR, A	ACALL address 11 (Page 7)	MOVX @R0, A	MOVX @R1, A	CPL A	MOV direct A	MOV @R0 A	MOV @R1 A	MOVR0, A	MOVR1, A	MOVR2, A	MOVR3, A	MOVR4, A	MOVR5, A	MOVR6, A	MOVR7, A

7.4 Simplified Description of Instructions

Note that “data address” is represented as “direct address” in this description.

Classification	Mnemonic	Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page
Arithmetic operation instructions	ADD A, Rr	0 0 1 0 1 r2 r1 r0	1	1	(AC),(OV),(C),(A)←(A)+(Rr) r=0~7	249
	ADD A, direct	0 0 1 0 0 1 0 1 a7 a6 a5 a4 a3 a2 a1 a0	2	1	(AC),(OV),(C),(A)←(A)+(direct address)	250
	ADD A, @Rr	0 0 1 0 0 1 1 r	1	1	(AC),(OV),(C),(A)←(A)+((Rr)) r=0 or 1	248
	ADD A, #data	0 0 1 0 0 1 0 0 l7 l6 l5 l4 l3 l2 l1 l0	2	1	(AC),(OV),(C),(A)←(A)+#data	247
	ADDC A, Rr	0 0 1 1 1 r2 r1 r0	1	1	(AC),(OV),(C),(A)←(A)+(C)+(Rr) r=0~7	253
	ADDC A, direct	0 0 1 1 0 1 0 1 a7 a6 a5 a4 a3 a2 a1 a0	2	1	(AC),(OV),(C),(A)←(A)+(C)+(direct address)	254
	ADDC A, @Rr	0 0 1 1 0 1 1 r	1	1	(AC),(OV),(C),(A)←(A)+(C)+((Rr)) r=0 or 1	252
	ADDC A, #data	0 0 1 1 0 1 0 0 l7 l6 l5 l4 l3 l2 l1 l0	2	1	(AC),(OV),(C),(A)←(A)+(C)+#data	251
	SUBB A, Rr	1 0 0 1 1 r2 r1 r0	1	1	(AC),(OV),(C),(A)←(A)–((C)+(Rr)) r=0~7	359
	SUBB A, direct	1 0 0 1 0 1 0 1 a7 a6 a5 a4 a3 a2 a1 a0	2	1	(AC),(OV),(C),(A)←(A)–((C)+(direct address))	360
	SUBB A, @Rr	1 0 0 1 0 1 1 r	1	1	(AC),(OV),(C),(A)←(A)–((C)+((Rr))) r=0 or 1	358
	SUBB A, #data	1 0 0 1 0 1 0 0 l7 l6 l5 l4 l3 l2 l1 l0	2	1	(AC),(OV),(C),(A)←(A)–((C)+#data)	357
	MUL AB	1 0 1 0 0 1 0 0	1	4	(AB)←(A)×(B)	335
	DIV AB	1 0 0 0 0 1 0 0	1	4	(A) quotient, (B) remainder ←(A)/(B)	284
	DA A	1 1 0 1 0 1 0 0	1	1	When the contents of accumulator bit 0 thru 3 exceed 9, and when the auxiliary carry (AC) is 1, 6 is added to bits 0 thru 3. And if examination od bits 4 thru 7 shows that the result of adding the carry following correction of the lower order bits 0 thru 3 by 6 is in excess of 9, or carry (C) is 1, 6 is added to bits 4 thru 7. If a carry is generated as a result, 1 is set in the carry flag.	278

Classification	Mnemonic	Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page
Accumulator operation instructions	CLR A	1 1 1 0 0 1 0 0	1	1	(A)←0	272
	CPL A	1 1 1 1 0 1 0 0	1	1	(A)←($\overline{A}$)	275
	RL A	0 0 1 0 0 0 1 1	1	1		349
	RLC A	0 0 1 1 0 0 1 1	1	1		350
	RR A	0 0 0 0 0 0 1 1	1	1		351
	RRC A	0 0 0 1 0 0 1 1	1	1		352
	SWAP A	1 1 0 0 0 1 0 0	1	1	(A4~7)⇄(A0~3)	361

Classification	Mnemonic		Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page
Increment & decrement instructions	INC	A	0 0 0 0 0 1 0 0	1	1	(A)←(A)+1	290
	INC	Rr	0 0 0 0 1 r_2 r_1 r_0	1	1	(Rr)←(Rr)+1 r=0~7	292
	INC	direct	0 0 0 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(direct address)←(direct address)+1	293
	INC	@Rr	0 0 0 0 0 1 1 r	1	1	((Rr))←((Rr))+1 r=0 or 1	289
	INC	DPTR	1 0 1 0 0 0 1 1	1	2	(DPTR)←(DPTR)+1	291
	DEC	A	0 0 0 1 0 1 0 0	1	1	(A)←(A)–1	281
	DEC	Rr	0 0 0 1 1 r_2 r_1 r_0	1	1	(Rr)←(Rr)–1 r=0~7	282
	DEC	direct	0 0 0 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(direct address)←(direct address)–1	283
	DEC	@Rr	0 0 0 1 0 1 1 r	1	1	((Rr))←((Rr))–1 r=0 or 1	280
Logical operation instructions	ANL	A, Rr	0 1 0 1 1 r_2 r_1 r_0	1	1	(A)←(A)AND(Rr) r=0~7	258
	ANL	A, direct	0 1 0 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(A)←(A)AND(direct address)	259
	ANL	A, @Rr	0 1 0 1 0 1 1 r	1	1	(A)←(A)AND((Rr)) r=0 or 1	257
	ANL	A, #data	0 1 0 1 0 1 0 0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(A)←(A)AND#data	256
	ANL	direct, A	0 1 0 1 0 0 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(direct address)←(direct address)AND(A)	263
	ANL	direct,#data	0 1 0 1 0 0 1 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	3	2	(direct address)←(direct address)AND#data	262
	ORL	A, Rr	0 1 0 0 1 r_2 r_1 r_0	1	1	(A)←(A)OR(Rr) r=0~7	339
	ORL	A, direct	0 1 0 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(A)←(A)OR(direct address)	340
	ORL	A, @Rr	0 1 0 0 0 1 1 r	1	1	(A)←(A)OR((Rr)) r=0 or 1	338

Classifi-cation	Mnemonic		Instruction code D7	D6	D5	D4	D3	D2	D1	D0	Byte	Cycle	Description	Page
Logical operation instructions	ORL	A, #data	0	1	0	0	0	1	0	0	2	1	(A)←(A)OR#data	337
			I7	I6	I5	I4	I3	I2	I1	I0				
	ORL	direct, A	0	1	0	0	0	0	1	0	2	1	(direct address)←(direct address)OR(A)	344
			a7	a6	a5	a4	a3	a2	a1	a0				
	ORL	direct,#data	0	1	0	0	0	0	1	1	3	2	(direct address)←(direct address)OR#data	343
			a7	a6	a5	a4	a3	a2	a1	a0				
			I7	I6	I5	I4	I3	I2	I1	I0				
	XRL	A, Rr	0	1	1	0	1	r2	r1	r0	1	1	(A)←(A)XOR(Rr) r=0~7	368
	XRL	A, direct	0	1	1	0	0	1	0	1	2	1	(A)←(A)XOR(direct address)	369
			a7	a6	a5	a4	a3	a2	a1	a0				
	XRL	A, @Rr	0	1	1	0	0	1	1	r	1	1	(A)←(A)XOR((Rr)) r=0 or 1	367
	XRL	A, #data	0	1	1	0	0	1	0	0	2	1	(A)←(A)XOR#data	366
			I7	I6	I5	I4	I3	I2	I1	I0				
	XRL	direct, A	0	1	1	0	0	0	1	0	2	1	(direct address)←(direct address)XOR(A)	371
			a7	a6	a5	a4	a3	a2	a1	a0				
	XRL	direct,#data	0	1	1	0	0	0	1	1	3	2	(direct address)←(direct address)XOR#data	370
			a7	a6	a5	a4	a3	a2	a1	a0				
			I7	I6	I5	I4	I3	I2	I1	I0				
Immediate data setting instructions	MOV	A, #data	0	1	1	1	0	1	0	0	2	1	(A)←#data	314
			I7	I6	I5	I4	I3	I2	I1	I0				
	MOV	Rr, #data	0	1	1	1	1	r2	r1	r0	2	1	(Rr)←#data r=0~7	320
			I7	I6	I5	I4	I3	I2	I1	I0				
	MOV	direct, #data	0	1	1	1	0	1	0	1	3	2	(direct address)←#data	324
			a7	a6	a5	a4	a3	a2	a1	a0				
			I7	I6	I5	I4	I3	I2	I1	I0				

Classification	Mnemonic		Instruction code D7	D6	D5	D4	D3	D2	D1	D0	Byte	Cycle	Description	Page
Immediate data setting instructions	MOV	@Rr, #data	0	1	1	1	0	1	1	r	2	1	((Rr))←#data r=0 or 1	311
			I7	I6	I5	I4	I3	I2	I1	I0				
	MOV	DPTR, #data	1	0	0	1	0	0	0	0	3	2	(DPTR)←#data	319
			I15	I14	I13	I12	I11	I10	I9	I8				
			I7	I6	I5	I4	I3	I2	I1	I0				
Carry flag operation instructions	CLR	C	1	1	0	0	0	0	1	1	1	1	(C)←0	273
	SETB	C	1	1	0	1	0	0	1	1	1	1	(C)←1	353
	CPL	C	1	0	1	1	0	0	1	1	1	1	(C)←$\overline{(C)}$	276
	ANL	C, bit	1	0	0	0	0	0	1	0	2	2	(C)←(C)AND(bit address)	260
			b7	b6	b5	b4	b3	b2	b1	b0				
	ANL	C,/bit	1	0	1	1	0	0	0	0	2	2	(C)←(C)AND$\overline{(\text{bit address})}$	261
			b7	b6	b5	b4	b3	b2	b1	b0				
	ORL	C, bit	0	1	1	1	0	0	1	0	2	2	(C)←(C)OR(bit address)	341
			b7	b6	b5	b4	b3	b2	b1	b0				
	ORL	C,/bit	1	0	1	0	0	0	0	0	2	2	(C)←(C)OR$\overline{(\text{bit address})}$	342
			b7	b6	b5	b4	b3	b2	b1	b0				
Bit transfer instructions	MOV	C, bit	1	0	1	0	0	0	1	0	2	1	(C)←(bit address)	318
			b7	b6	b5	b4	b3	b2	b1	b0				
	MOV	bit, C	1	0	0	1	0	0	1	0	2	2	(bit address)←(C)	323
			b7	b6	b5	b4	b3	b2	b1	b0				
Bit manipulation instructions	SETB	bit	1	1	0	1	0	0	1	0	2	1	(bit address)←1	354
			b7	b6	b5	b4	b3	b2	b1	b0				
	CLR	bit	1	1	0	0	0	0	1	0	2	1	(bit address)←0	274
			b7	b6	b5	b4	b3	b2	b1	b0				

Classification	Mnemonic		Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page
Bit manipulation instructions	CPL	bit	1 0 1 1 0 0 1 0 b7 b6 b5 b4 b3 b2 b1 b0	2	1	(bit address)←$\overline{\text{(bit address)}}$	277
Data transfer instructions	MOV	A, Rr	1 1 1 0 1 r2 r1 r0	1	1	(A)←(Rr) r=0~7	316
	MOV	A, direct	1 1 1 0 0 1 0 1 a7 a6 a5 a4 a3 a2 a1 a0	2	1	(A)←(direct address)	317
	MOV	A, @Rr	1 1 1 0 0 1 1 r	1	1	(A)←((Rr)) r=0 or 1	315
	MOV	Rr, A	1 1 1 1 1 r2 r1 r0	1	1	(Rr)←(A) r=0~7	321
	MOV	Rr, direct	1 0 1 0 1 r2 r1 r0 a7 a6 a5 a4 a3 a2 a1 a0	2	2	(Rr)←(direct address) r=0~7	322
	MOV	direct, A	1 1 1 1 0 1 0 1 a7 a6 a5 a4 a3 a2 a1 a0	2	1	(direct address)←(A)	326
	MOV	direct, Rr	1 0 0 0 1 r2 r1 r0 a7 a6 a5 a4 a3 a2 a1 a0	2	2	(direct address)←(Rr) r=0~7	327
	MOV	direct1, direct 2	1 0 0 0 0 1 0 1 a_7^2 a_6^2 a_5^2 a^2 a_3^2 a_2^2 a_1^2 a_0^2 a_7^1 a_6^1 a_5^1 a_4^1 a_3^1 a_2^1 a_1^1 a_0^1	3	2	(direct address 1)←(direct address 2)	328
	MOV	direct, @Rr	1 0 0 0 0 1 1 r a7 a6 a5 a4 a3 a2 a1 a0	2	2	(direct address)←((Rr)) r=0 or 1	325
	MOV	@Rr, A	1 1 1 1 0 1 1 r	1	1	((Rr))←(A) r=0 or 1	312
	MOV	@Rr, direct	1 0 1 0 0 1 1 r a7 a6 a5 a4 a3 a2 a1 a0	2	2	((Rr))←(direct address) r=0 or 1	313
Constant value instructions	MOVC	A,@A+DPTR	1 0 0 1 0 0 1 1	1	2	(A)←((A)+(DPTR))	329
	MOVC	A, @A+PC	1 0 0 0 0 0 1 1	1	2	(PC)←(PC)+1 (A)←((A)+(PC))	330

Classification	Mnemonic	Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page
Data exchange instructions	XCH A, Rr	0 0 0 1 1 r_2 r_1 r_0	1	1	(A)⇄(Rr) r=0~7	363
	XCH A, direct	1 1 0 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(A)⇄(direct address)	364
	XCH A, @Rr	1 1 0 0 0 1 1 r	1	1	(A)⇄((Rr)) r=0 or 1	362
	XCHD A, @Rr	1 1 0 1 0 1 1 r	1	1	($A_{0\sim3}$)⇄(($Rr_{0\sim3}$)) r=0 or 1	365
Subroutine instructions	PUSH direct	1 1 0 0 0 0 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	(SP)←(SP)+1 ((SP))←(direct address)	346
	POP direct	1 1 0 1 0 0 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	(direct address)←((SP)) (SP)←(SP)–1	345
	ACALL addr 11	A_{10} A_9 A_8 1 0 0 0 1 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	2	2	(PC)←(PC)+2 (SP)←(SP)+1 ((SP))←($PC_{0\sim7}$) (SP)←(SP)+1 ((SP))←($PC_{8\sim15}$) ($PC_{0\sim10}$)←$A_{0\sim10}$	246
	LCALL addr 16	0 0 0 1 0 0 1 0 A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	3	2	(PC)←(PC)+3 (SP)←(SP)+1 ((SP))←($PC_{0\sim7}$) (SP)←(SP)+1 ((SP))←($PC_{8\sim15}$) ($PC_{0\sim15}$)←$A_{0\sim15}$	309
	RET	0 0 1 0 0 0 1 0	1	2	($PC_{8\sim15}$)←((SP)) (SP)←(SP)–1 ($PC_{0\sim7}$)←((SP)) (SP)←(SP)–1	347

Classification	Mnemonic	Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page
Subroutine instructions	RETI	0 0 1 1 0 0 1 0	1	2	(PC$_{8\sim15}$)←((SP)) (SP)←(SP)–1 (PC$_{0\sim7}$)←((SP)) (SP)←(SP)–1 *INTERRUPT ENABLE	348
Jump instructions	AJMP addr 11	A_{10} A_9 A_8 0 0 0 0 1 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	2	2	(PC)←(PC)+2 (PC$_{0\sim10}$)←A$_{0\sim10}$	255
	LJMP addr 16	0 0 0 0 0 0 1 0 A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	3	2	(PC$_{0\sim15}$)←A$_{0\sim15}$	310
	SJMP rel	1 0 0 0 0 0 0 0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	2	2	(PC)←(PC)+2 (PC)←(PC)+relative offset	355
	JMP @A+DPTR	0 1 1 1 0 0 1 1	1	2	(PC)←(A)+(DPTR)	300

Branching instructions	CJNE A, direct, rel	1 0 1 1 0 1 0 1
		a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0
		R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0
			3	2	(PC)←(PC)+3
					IF (A)≠(direct address)
					THEN
					(PC)←(PC)+relative offset
					IF (A)<(direct address)
					THEN
					(C)←1
					ELSE
					(C)←0
						268

Classification	Mnemonic	Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page

Branching instructions	CJNE A, #data, rel	1 0 1 1 0 1 0 0
		I7 I6 I5 I4 I3 I2 I1 I0
		R7 R6 R5 R4 R3 R2 R1 R0
			3	2	(PC)←(PC)+3
					IF (A)≠#data
					THEN
					(PC)←(PC)+relative offset
					IF (A)<#data
					THEN
					(C)←1
					ELSE
					(C)←0
						266

	CJNE Rr,#data,rel	1 0 1 1 1 r2 r1 r0
		I7 I6 I5 I4 I3 I2 I1 I0
		R7 R6 R5 R4 R3 R2 R1 R0
			3	2	(PC)←(PC)+3
					IF (Rr)≠#data r=0~7
					THEN
					(PC)←(PC)+relative offset
					IF (A)<#data r=0~7
					THEN
					(C)←1
					ELSE
					(C)←0
						270

Classification	Mnemonic	Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page

Branching instructions	CJNE @Rr, #data, rel	1 0 1 1 0 1 1 r
		I7 I6 I5 I4 I3 I2 I1 I0
		R7 R6 R5 R4 R3 R2 R1 R0
			3	2	(PC)←(PC)+3
					IF ((Rr))≠#data r=0 or 1
					THEN
					(PC)←(PC)+relative offset
					IF ((Rr))<#data r=0 or 1
					THEN
					(C)←1
					ELSE
					(C)←0
						264

	DJNZ Rr, rel	1 1 0 1 1 r2 r1 r0 R7 R6 R5 R4 R3 R2 R1 R0	2	2	(PC)←(PC)+2 (Rr)←(Rr)–1 r=0~7 IF (Rr)≠0 r=0~7 THEN (PC)←(PC)+relative offset	285
	DJNZ direct, rel	1 1 0 1 0 1 0 1 a7 a6 a5 a4 a3 a2 a1 a0 R7 R6 R5 R4 R3 R2 R1 R0	3	2	(PC)←(PC)+3 (direct address)←(direct address)–1 IF (direct address)≠0 THEN (PC)←(PC)+relative offset	287
	JZ rel	0 1 1 0 0 0 0 0 R7 R6 R5 R4 R3 R2 R1 R0	2	2	(PC)←(PC)+2 IF (A)=0 THEN (PC)←(PC)+relative offset	307

Classification	Mnemonic	Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page
Branching instructions	JNZ rel	0 1 1 1 0 0 0 0 R7 R6 R5 R4 R3 R2 R1 R0	2	2	(PC)←(PC)+2 IF (A)≠0 THEN (PC)←(PC)+relative offset	305
	JC rel	0 1 0 0 0 0 0 0 R7 R6 R5 R4 R3 R2 R1 R0	2	2	(PC)←(PC)+2 IF (C)=1 THEN (PC)←(PC)+relative offset	298
	JNC rel	0 1 0 1 0 0 0 0 R7 R6 R5 R4 R3 R2 R1 R0	2	2	(PC)←(PC)+2 IF (C)=0 THEN (PC)←(PC)+relative offset	303
	JB bit, rel	0 0 1 0 0 0 0 0 b7 b6 b5 b4 b3 b2 b1 b0 R7 R6 R5 R4 R3 R2 R1 R0	3	2	(PC)←(PC)+3 IF (bit address)=1 THEN (PC)←(PC)+relative offset	294
	JNB bit, rel	0 0 1 1 0 0 0 0 b7 b6 b5 b4 b3 b2 b1 b0 R7 R6 R5 R4 R3 R2 R1 R0	3	2	(PC)←(PC)+3 IF (bit address)=0 THEN (PC)←(PC)+relative offset	301
	JBC bit, rel	0 0 0 1 0 0 0 0 b7 b6 b5 b4 b3 b2 b1 b0 R7 R6 R5 R4 R3 R2 R1 R0	3	2	(PC)←(PC)+3 IF (bit address)=1 THEN (bit address)←0 (PC)←(PC)+relative offset	296

Classification	Mnemonic	Instruction code D7 D6 D5 D4 D3 D2 D1 D0	Byte	Cycle	Description	Page
External memory instructions	MOVX A, @Rr	1 1 1 0 0 0 1 r	1	2	(A)←((Rr)) EXTERNAL RAM r=0 or 1	334
	MOVX A, @DPTR	1 1 1 0 0 0 0 0	1	2	(A)←((DPTR)) EXTERNAL RAM	333
	MOVX @Rr, A	1 1 1 1 0 0 1 r	1	2	((Rr))←(A) EXTERNAL RAM r=0 or 1	332
	MOVX @DPTR, A	1 1 1 1 0 0 0 0	1	2	((DPTR))←(A) EXTERNAL RAM	331
Other instruction	NOP	0 0 0 0 0 0 0 0	1	1	(PC)←(PC)+1	336

7.5 Detailed Description of MSM80C154S/MSM83C154S Instructions

Note: “direct address” is represented as “data address” in this detailed description.

1. ACALL code address (Absolute call within 2K bytes page)

Instruction code : 7 0

A_{10}	A_9	A_8	1	0	0	0	0

Byte 1

Call address 7 0

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0

Byte 2

Operations : (PC)←(PC)+2
(SP)←(SP)+1
((SP))←($PC_{0\sim7}$)
(SP)←(SP)+1
((SP))←($PC_{8\sim15}$)
($PC_{0\sim10}$)←$A_{0\sim10}$

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : This instruction stores the program counter value (return address) in the stack following an increment operation. The program counter data $PC_{0\sim10}$ following PC+2 is replaced by 11-bit page address data $A_{0\sim10}$. The destination address for this instruction must always be within the 2K byte page, but if the instruction is placed at address X7FEH or X7FFH, execution proceeds from the call address on the next page.

2. ADD A, #data (Add immediate data)

Instruction code : Byte 1

7							0
0	0	1	0	0	1	0	0

#data Byte 2

7							0
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Operation : (A)←(A)+#data

Number of bytes : 2

Number of cycles : 1

Flags (PSW) :

C	AC	F0	RS1	RS0	OV	F1	P
•	•				•		•

Description : An 8-bit immediate data value is added to the accumulator. The result is placed in the accumulator, and the flags are updated.

Example ADD A, #07H

Instruction code : Byte 1

7							0
0	0	1	0	0	1	0	0

Byte 2

7							0
0	0	0	0	0	1	1	1

Before execution
Accumulator

0	1	1	0	0	0	1	0
7							0

After execution
Accumulator

0	1	1	0	1	0	0	1
7							0

3. ADD A, @Rr (Add indirect address)

Instruction code : 7 0

0	0	1	0	0	1	1	r

Byte 1

Operation : (A)←(A)+((Rr)) r=0 or 1

Number of bytes : 1

Number of cycles : 1

Flags :

C	AC	F0	RS1	RS0	OV	F1	P
(PSW)	•	•				•		•

Description : The data memory location contents addressed by the register r contents are added to the accumulator. The result is placed in the accumulator, and the flags are updated.

Example ADD A, @R0

Instruction code : 7 0

0	0	1	0	0	1	1	0

Byte 1

Before execution

Accumulator

0	1	0	0	1	1	0	1

7 0

Register 0

0	1	0	1	1	1	0	0

7 0

5CH

0	1	1	0	1	0	0	1

7 0

After execution

Accumulator

1	0	1	1	0	1	1	0

7 0

Register 0

0	1	0	1	1	1	0	0

7 0

5CH

0	1	1	0	1	0	0	1

7 0

4. ADD A, Rr (Add register)

Instruction code : 7 0

0	0	1	0	1	r2	r1	r0

Byte 1

Operation : (A)←(A)+(Rr) r=0 thru 7

Number of bytes : 1

Number of cycles : 1

Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P
•	•				•		•

Description : The register r contents are added to the accumulator. The result is placed in the accumulator, and the flags are updated.

Example ADD A, R6

Instruction code : 7 0

0	0	1	0	1	1	1	0

Byte 1

Before execution

Accumulator

0	1	0	1	0	1	0	0

7 0

Register 6

0	1	0	1	1	1	0	1

7 0

After execution

Accumulator

1	0	1	1	0	0	0	1

7 0

Register 6

0	1	0	1	1	1	0	1

7 0

5. ADD A, data address (Add memory)

Instruction code : Byte 1

7							0
0	0	1	0	0	1	0	1

Data address Byte 2

7							0
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Operation : (A)←(A)+(data address)

Number of bytes : 2

Number of cycles : 1

Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P
•	•				•		•

Description : The specified data address contents are added to the accumulator. The result is placed in the accumulator, and the flags are updated.

Example ADD A, P1

Instruction code : Byte 1

7							0
0	0	1	0	0	1	0	1

Byte 2

7							0
1	0	0	1	0	0	0	0

Before execution

Accumulator

7							0
0	1	1	1	0	0	1	0

Port 1(90H)

7							0
1	1	0	0	0	1	1	0

After execution

Accumulator

7							0
0	0	1	1	1	0	0	0

Port 1(90H)

7							0
1	1	0	0	0	1	1	0

6. ADDC A, #data (Add carry plus immediate data to accumulator)

Instruction code : Byte 1

7							0
0	0	1	1	0	1	0	0

#data Byte 2

7							0
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Operation : (A)←(A)+(C)+#data

Number of bytes : 2

Number of cycles : 1

Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P
•	•				•		•

Description : The carry flag is added to the accumulator, and an 8-bit immediate data is added to that result. The result is placed in the accumulator, and the flags are updated.

Example ADDC A, #76H

Instruction code : Byte 1

7							0
0	0	1	1	0	1	0	0

Byte 2

7							0
0	1	1	1	0	1	1	0

Before execution

Accumulator

0	1	0	1	1	0	0	1
7							0

Carry flag

1

After execution

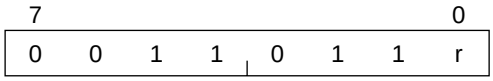
Accumulator

1	1	0	1	0	0	0	0
7							0

Carry flag

0

7. ADDC A, @Rr (Add carry plus indirect address to accumulator)

Instruction code : 

Operation : (A)←(A)+(C)+((Rr)) r=0 or 1

Number of bytes : 1

Number of cycles : 1

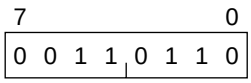
Flags : C AC F0 RS1 RS0 OV F1 P

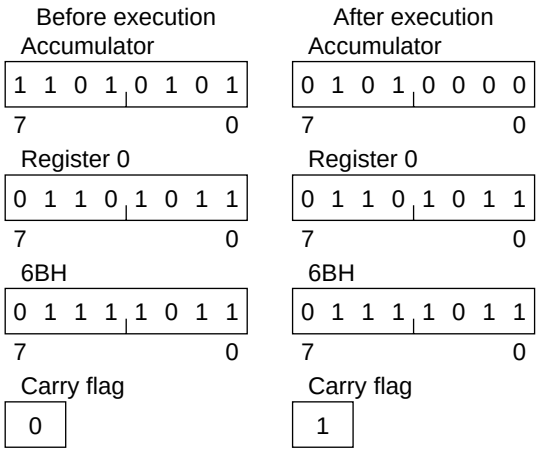
(PSW)

C	AC	F0	RS1	RS0	OV	F1	P
•	•				•		•

Description : The carry flag is added to the accumulator, and the contents of data memory location addressed by the register r contents are added to the accumulator. The result is placed in the accumulator, and the flags are updated.

Example ADDC A, @R0

Instruction code : 



8. ADD A, Rr (Add carry plus register to accumulator)

Instruction code : 7 0 | 0 0 1 1 1 r2 r1 r0 | Byte 1

Operation : (A)←(A)+(C)+(Rr) r=0 thru 7

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • • | | | | • | | •

Description : The carry flag is added to the accumulator,and the register r contents are added to the result. The result is placed in the accumulator, and the flags are updated.

Example ADDC A, R2

Instruction code : 7 0 | 0 0 1 1 1 0 1 0 | Byte 1

Before execution

Accumulator: 0 1 1 0 1 0 0 0 (7 0)

Register 2: 0 1 1 0 1 1 1 0 (7 0)

Carry flag: 1

After execution

Accumulator: 1 1 0 1 0 1 1 1 (7 0)

Register 2: 0 1 1 0 1 1 1 0 (7 0)

Carry flag: 0

9. ADDC A, data address (Add carry plus memory to accumulator)

Instruction code : 7 0

0	0	1	1	0	1	0	1	Byte 1

Data address 7 0

a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2

Operation : (A)←(A)+(C)+(data address)

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

•	•				•		•

Description : The carry flag is added to the accumulator,and the specified data address contents are added to that result. The result is placed in the accumulator, and the flags are updated.

Example ADDC A, 45H

Instruction code : 7 0

0	0	1	1	0	1	0	1	Byte 1

7 0

0	1	0	0	0	1	0	1	Byte 2

Before execution

Accumulator

0	0	1	1	0	0	1	1

7 0

45H

0	1	0	1	1	1	1	0

7 0

Carry flag

1

After execution

Accumulator

1	0	0	1	0	0	1	0

7 0

45H

0	1	0	1	1	1	1	0

7 0

Carry flag

0

10. AJMP code address (Absolute jump within 2K byte page)

Instruction code : 7 0

A_{10}	A_9	A_8	0	0	0	0	1	Byte 1

Call address : 7 0

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Byte 2

Operations : $(PC) \leftarrow (PC)+2$
$(PC_{0\sim10}) \leftarrow A_{0\sim10}$

Number of bytes : 2

Number of cycles : 2

Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P

Description : After an increment ,the program counter $PC_{0\sim10}$ is replaced by 11-bit page address data $A_{0\sim10}$. The destination address for this instruction must always be within the 2K byte page, but if the instruction is placed at address X7FEH or X7FFH, execution proceeds from the jump address on the next page.

11. ANL A, #data (Logical AND immediate data to accumulator)

Instruction code : 0 1 0 1 0 1 0 0 Byte 1

#data : I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0 Byte 2

Operation : (A)←(A) AND #data

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • (P)

Description : The logical AND between an 8-bit immediate data value and the accumulator contents is determined. The result is placed in the accumulator and the flag is updated.

Example ANL A, #0AH

Instruction code :

0 1 0 1 0 1 0 0 Byte 1

0 0 0 0 1 0 1 0 Byte 2

Before execution Accumulator	After execution Accumulator
1 0 1 1 1 1 0 1	0 0 0 0 1 0 0 0
7 0	7 0

12. ANL A, @Rr (Logical AND indirect address to accumulator)

Instruction code : 7 0 | 0 1 0 1 0 1 1 r | Byte 1

Operation : (A)←(A) AND ((Rr)) r=0 or 1

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : | | | | | | | • |

Description : The logical AND between the accumulator contents and the data memory location contents addressed by the register r contents is determined. The result is placed in the accumulator and the flag is updated.

Example ANL A, @R0

Instruction code : 7 0 | 0 1 0 1 0 1 1 0 | Byte 1

Before execution

Accumulator: 1 0 1 0 1 1 1 0 (7 0)

Register 0: 0 1 0 1 1 0 0 0 (7 0)

RAM 58H: 1 1 1 1 1 0 1 0 (7 0)

After execution

Accumulator: 1 0 1 0 1 0 1 0 (7 0)

Register 0: 0 1 0 1 1 0 0 0 (7 0)

RAM 58H: 1 1 1 1 1 0 1 0 (7 0)

13. ANL A, Rr (Logical AND register to accumulator)

Instruction code	:	7 0 / 0 1 0 1 1 r2 r1 r0 / Byte 1
Operation	:	(A)←(A) AND (Rr) r=0 thru 7
Number of bytes	:	1
Number of cycles	:	1
Flags	:	C AC F0 RS1 RS0 OV F1 P
(PSW)		•
Description	:	The logical AND between the accumulator contents and the register r contents is determined. The result is placed in the accumulator and the flag is updated.

Example ANL A, R5

Instruction code : 7 0 / 0 1 0 1 1 1 0 1 / Byte 1

Before execution		After execution	
Accumulator		Accumulator	
1 1 0 1 1 0 1 1		0 1 0 1 0 0 0 1	
7 0		7 0	
Register 5		Register 5	
0 1 0 1 0 1 0 1		0 1 0 1 0 1 0 1	
7 0		7 0	

14. ANL A, data address (Logical AND memory to accumulator)

Instruction code : 7 0 | 0 1 0 1 0 1 0 1 | Byte 1

Data address : 7 0 | a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 | Byte 2

Operation : (A)←(A) AND (data address)

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : | | | | | | | • |

Description : The logical AND between the accumulator contents and the specified data address contents is determined. The result is placed in the accumulator and the flag is updated.

Example ANL A, P1

Instruction code : 7 0 | 0 1 0 1 0 1 0 1 | Byte 1

7 0 | 1 0 0 1 0 0 0 0 | Byte 2

Before execution

Accumulator: 1 1 1 0 0 1 0 1 (7 0)

Port 1: 1 0 1 0 1 1 1 1 (7 0)

After execution

Accumulator: 1 0 1 0 0 1 0 1 (7 0)

Port 1: 1 0 1 0 1 1 1 1 (7 0)

15. ANL C, bit address (Logical AND bit to carry flag)

Instruction code : 1 0 0 0 0 0 1 0 Byte 1

Bit address : b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 Byte 2

Operation : (C)←(C) AND (bit address)

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • | | | | | | | |

Description : The logical AND between the carry flag and the specified bit address contents is determined. The result is placed in the carry flag.

Example ANL C, ACC.5

Instruction code : 1 0 0 0 0 0 1 0 Byte 1

1 1 1 0 0 1 0 1 Byte 2

Before execution
Carry flag: 1

After execution
Carry flag: 0

Accumulator: 1 0 0 1 1 0 1 0 (7 5 0)

Accumulator: 1 0 0 1 1 0 1 0 (7 5 0)

16. ANL C,/bit address (Logical AND complement bit to carry flag)

Instruction code : 7 0

1	0	1	1	0	0	0	0	Byte 1

Bit address 7 0

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Byte 2

Operation : (C)←(C) AND ($\overline{\text{bit address}}$)

Number of bytes : 2

Number of cycles : 2

Flags :

C	AC	F0	RS1	RS0	OV	F1	P
(PSW)	•							

Description : The logical AND between the carry flag and the complement of specified bit address contents is determined. The result is placed in the carry flag.

Example ANL C,/P1.3

Instruction code : 7 0

1	0	1	1	0	0	0	0	Byte 1

7 0

1	0	0	1	0	0	1	1	Byte 2

Before execution

Carry flag

1

After execution

Carry flag

0

Port 1

0	0	0	0	1	0	1	0

7 3 0

Port 1

0	0	0	0	1	0	1	0

7 3 0

17. ANL data address, #data (Logical AND immediate data to memory)

Instruction code : 7 0 | 0 1 0 1 0 0 1 1 | Byte 1

Data address 7 0 | a7 a6 a5 a4 a3 a2 a1 a0 | Byte 2

#data 7 0 | I7 I6 I5 I4 I3 I2 I1 I0 | Byte 3

Operation : (data address)←(data address) AND #data

Number of bytes : 3

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The logical AND between an 8-bit immediate data value and the specified data address contents is determined. The result is placed in the specified data address.

Example ANL DPH, #0AAH

Instruction code :

7 0 | 0 1 0 1 0 0 1 1 | Byte 1

7 0 | 1 0 0 0 0 0 1 1 | Byte 2

7 0 | 1 0 1 0 1 0 1 0 | Byte 3

Before execution	After execution
DPH	DPH
1 1 1 1 1 1 1 1	1 0 1 0 1 0 1 0
7 0	7 0

18. ANL data address, A (Logical AND accumulator to memory)

Instruction code : 7 0 — 0 1 0 1 0 0 1 0 — Byte 1

Data address — 7 0 — a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 — Byte 2

Operation : (data address)←(data address) AND (A)

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The logical AND between the accumulator and the specified data address contents is determined. The result is placed in the specified data address.

Example ANL TCON, A

Instruction code :

7 0 — 0 1 0 1 0 0 1 0 — Byte 1

7 0 — 1 0 0 0 1 0 0 0 — Byte 2

Before execution		After execution	
Accumulator		Accumulator	
0 1 0 1 1 0 1 0		0 1 0 1 1 0 1 0	
7 0		7 0	
TCON		TCON	
1 0 1 1 0 1 0 1		0 0 0 1 0 0 0 0	
7 0		7 0	

19. CJNE @Rr, #data, code address
(Compare indirect address to immediate data, jump if not equal)

		7							0	
Instruction code	:	1	0	1	1	0	1	1	r	Byte 1
		7							0	
#data		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Byte 2
		7							0	
Relative offset		R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0	Byte 3

Operations : (PC)←(PC)+3
IF ((Rr))≠#data r=0 or 1
THEN
(PC)←(PC)+relative offset
IF ((Rr))<#data r=0 or 1
THEN
(C)←1
ELSE
(C)←0

Number of bytes : 3

Number of cycles : 2

Flags :

	C	AC	F0	RS1	RS0	OV	F1	P
(PSW)	•							

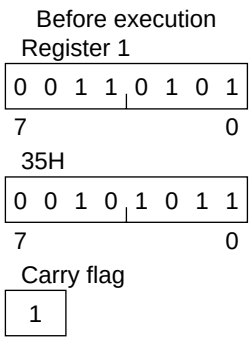
Description : The data memory location contents addressed by the register r contents are compared with an immediate data value. Control is shifted to a relative jump address if the compared data is not equal. If the compared data is equal, control is shifted to the next address following this instruction. The carry flag is set to 1 if the immediate data value is greater than the specified address contents, but is set to 0 if otherwise.

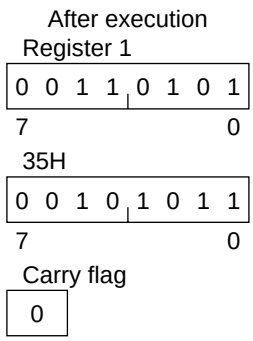
Example CJNE @R1, #05H, TEST

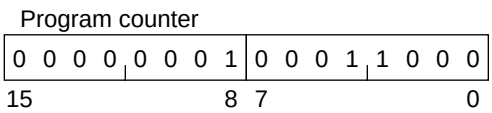
LOC	OBJ	SOURCE
00B4	2155	TEST:AJMP TEST1
0118	B70599	COMP:CJNE @R1, #05H, TEST
011B	020500	OUT:LJMP OUT1

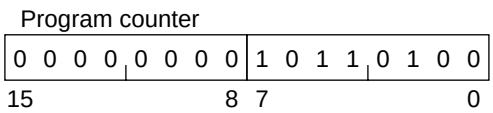
Instruction code :

	7							0	
Byte 1	1	0	1	1	0	1	1	1	
Byte 2	0	0	0	0	0	1	0	1	
Byte 3	1	0	0	1	1	0	0	1	









20. CJNE A, #data, code address
(Compare immediate data to accumulator, jump if not equal)

Instruction code :

7							0	
1	0	1	1	0	1	0	0	Byte 1

#data

7							0	
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Byte 2

Relative offset

7							0	
R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0	Byte 3

Operations : (PC)←(PC)+3
IF (A)≠#data
THEN
(PC)←(PC)+relative offset
IF (A)<#data
THEN
(C)←1
ELSE
(C)←0

Number of bytes : 3

Number of cycles : 2

Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P
•							

Description : The accumulator contents are compared with an immediate data value, and control is shifted to a relative jump address if the compared data is not equal. If the compared data is equal, control is shifted to the next address following this instruction. The carry flag is set to 1 if the immediate data value is greater than the accumulator contents, but is set to 0 if otherwise.

Example CJNE A, #0AH, SS1

LOC	OBJ	SOURCE
0064	FF	SS1:MOV R7, A
00C8	B40599	COMP:CJNE A, #0AH, SS1
00CB	0D	INCR:INC R5

Instruction code :

	7							0	
Byte 1	1	0	1	1	0	1	0	0	
Byte 2	0	0	0	0	1	0	1	0	
Byte 3	1	0	0	1	1	0	0	1	

Before execution

Accumulator

7							0
0	1	0	1	0	0	0	0

Carry flag

1

After execution

Accumulator

7							0
0	1	0	1	0	0	0	0

Carry flag

0

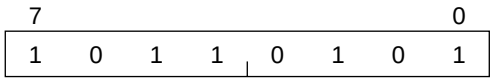
Program counter

15							8	7							0
0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0

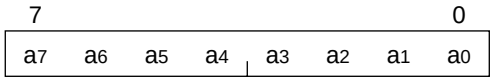
Program counter

15							8	7							0
0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0

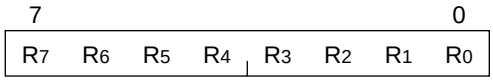
21. CJNE A, data address, code address (Compare memory to accumulator, jump if not equal)

Instruction code : 

Byte 1

Data address 

Byte 2

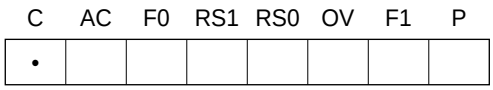
Relative offset 

Byte 3

Operations : (PC)←(PC)+3
IF (A)≠(data address)
THEN
(PC)←(PC)+relative offset
IF (A)<(data address)
THEN
(C)←1
ELSE
(C)←0

Number of bytes : 3

Number of cycles : 2

Flags : 

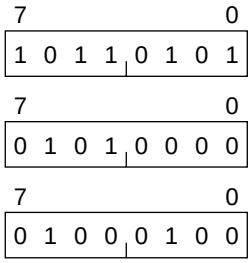
(PSW)

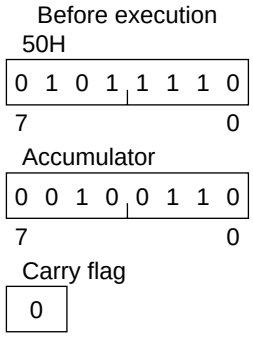
Description : The accumulator contents are compared with the specified data address contents, and control is shifted to a relative jump address if the compared data is not equal. If the compared data is equal, control is shifted to the next address following this instruction. The carry flag is set to 1 if the specified data address contents are greater than the accumulator contents, but is set to 0 if otherwise.

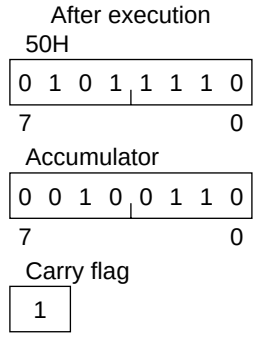
Example CJNE A, 50H, NEXT

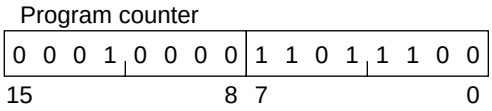
LOC	OBJ	SOURCE
10DC	B55044	COMP:CJNE A, 50H, NEXT
10DF	120100	CAL:LCALL TEST
1123	14	NEXT:DEC A

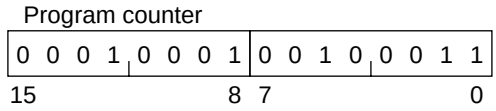
Instruction code :











22. CJNE Rr, #data, code address
(Compare immediate data to register, jump if not equal)

Instruction code :

7							0	
1	0	1	1	1	r_2	r_1	r_0	Byte 1

#data

7							0	
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Byte 2

Relative offset

7							0	
R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0	Byte 3

Operations : (PC)←(PC)+3
IF ((Rr))≠#data r=0 thru 7
THEN
(PC)←(PC)+relative offset
IF ((Rr))<#data r=0 thru 7
THEN
(C)←1
ELSE
(C)←0

Number of bytes : 3

Number of cycles : 2

Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P
•							

Description : The register r contents are compared with an immediate data value, and control is shifted to a relative jump address if the compared data is not equal. If the compared data is equal, control is shifted to the next address following this instruction. The carry flag is set to 1 if the immediate data value is greater than the register r contents, but is set to 0 if otherwise.

Example CJNE R4, #32H, COUNT

LOC	OBJ	SOURCE
0473	0C	COUNT:INC R4
0482	BC32EE	COMP:CJNE R4, #32H, COUNT

Instruction code :

	7							0	
Byte 1	1	0	1	1	1	1	0	0	
Byte 2	0	0	1	1	0	0	1	0	
Byte 3	1	1	1	0	1	1	1	0	

Before execution

Register 4

7							0
0	0	0	0	0	0	0	1

Carry flag

1

After execution

Register 4

7							0
0	0	0	0	0	0	0	1

Carry flag

1

Program counter

15							8	7							0
0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0

Program counter

15							8	7							0
0	0	0	0	0	1	0	0	0	1	1	1	0	0	1	1

23. CLR A (Clear accumulator)

Instruction code : 7 6 5 4 3 2 1 0 — 1 1 1 0 0 1 0 0 — Byte 1

Operation : (A)←0

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : | | | | | | | | • |

Description : The accumulator is cleared to 0 and flag is updated.

Example CLR A

Instruction code : 7 6 5 4 3 2 1 0 — 1 1 1 0 0 1 0 0 — Byte 1

Before execution Accumulator								After execution Accumulator							
1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0
7							0	7							0

24. CLR C (Clear carry flag)

Instruction code : 7 0

1	1	0	0	0	0	1	1

Byte 1

Operation : (C)←0

Number of bytes : 1

Number of cycles : 1

Flags :

C	AC	F0	RS1	RS0	OV	F1	P
•							

(PSW)

Description : The carry flag is cleared to 0.

Example CLR C

Instruction code : 7 0

1	1	0	0	0	0	1	1

Byte 1

Before execution	After execution
Carry flag	Carry flag
1	0

25. CLR bit address (Clear bit)

Instruction code : 7 0 — 1 1 0 0 0 0 1 0 — Byte 1

Bit address — b7 b6 b5 b4 b3 b2 b1 b0 — Byte 2

Operation : (bit address)←0

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The specified bit address content is cleared to 0.

Example CLR P1.5

Instruction code :

Byte 1: 1 0 0 0 0 0 1 0

Byte 2: 1 1 1 0 0 1 0 1

Before execution — Port 1: 1 1 1 1 1 1 1 1 (7 5 0)

After execution — Port 1: 1 1 0 1 1 1 1 1 (7 5 0)

26. CPL A (Complement accumulator)

Instruction code : 7 6 5 4 3 2 1 0 — 1 1 1 1 0 1 0 0 — Byte 1

Operation : (A)←($\overline{A}$)

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : Accumulator data 0 is set to 1 and 1 is set to 0.

Example CPL A

Instruction code : 7 6 5 4 3 2 1 0 — 1 1 1 1 0 1 0 0 — Byte 1

Before execution
Accumulator
1 1 0 1 0 1 0 1
7 0

After execution
Accumulator
0 0 1 0 1 0 1 0
7 0

27. CPL C (Complement carry flag)

Instruction code : 7 0

1	0	1	1	0	0	1	1	Byte 1

Operation : $(C) \leftarrow (\overline{C})$

Number of bytes : 1

Number of cycles : 1

Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P
•							

Description : The carry flag is set to 1 if 0, set to 0 if 1.

Example CPL C

Instruction code : 7 0

1	0	1	1	0	0	1	1	Byte 1

Before execution	After execution
Carry flag 1	Carry flag 0
Carry flag 0	Carry flag 1

28. CPL bit address (Complement bit)

Instruction code : 7 0 — 1 0 1 1 0 0 1 0 — Byte 1

Bit address — 7 0 — b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 — Byte 2

Operation : (bit address)←$\overline{\text{(bit address)}}$

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The specified bit address content is set to 1 if 0, and set to 0 if 1.

Example CLR B.7

Instruction code : 7 0 — 1 0 1 1 0 0 1 0 — Byte 1

7 0 — 1 1 1 1 0 1 1 1 — Byte 2

Before execution
B register
0 1 0 1 0 1 1 1
7 0

After execution
B register
1 1 0 1 0 1 1 1
7 0

29. DA A (Decimal adjust accumulator)

Instruction code :

7							0	
1	1	0	1	0	1	0	0	Byte 1

Operations : 10^0+6←(AC)=1 or 10^0>10

10^1+6, (C)←1 ←(C)=1 or 10^1>10

Number of bytes : 1

Number of cycles : 1

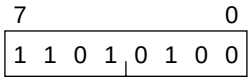
Flags :

C	AC	F0	RS1	RS0	OV	F1	P
•							•

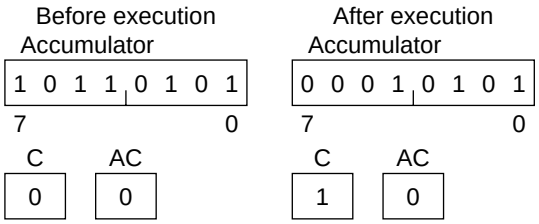
(PSW)

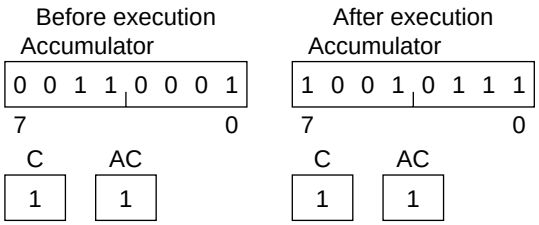
Description : The arithmetic operation result located in the accumulator following an addition between two 2-digit decimal number is converted to a normal decimal number. When the contents of accumulator bits 0 thru 3 (10^0 digit) are greater than 9, or when the auxiliary carry (AC) is 1, 6 is added to accumulator bits 0 thru 3. And if the contents of accumulator bits 4 thru 7 (10^1 digit) exceed 9, or if the result obtained by adding a carry from the lower order digits after compensation is greater than 9, or if the carry flag is 1, 6 is added to the data in accumulator bits 4 thru 7. The flags are also updated.

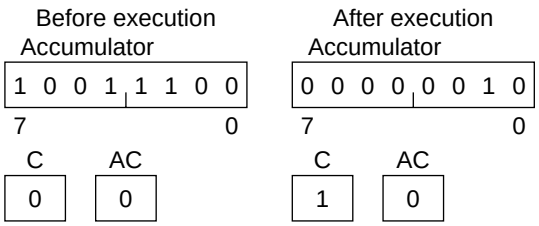
Example DA A

Instruction code : 

Byte 1







30. DEC @Rr (Decrement indirect address)

Instruction code : 7 0 | 0 0 0 1 0 1 1 r | Byte 1

Operation : ((Rr))←((Rr))–1 r=0 or 1

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The contents of the data memory location addressed by the register r contents are decremented by 1.

Example DEC @R0

Instruction code : 7 0 | 0 0 0 1 0 1 1 0 | Byte 1

Before execution

Register 0: 0 1 1 0 1 0 1 0 (7 0)

6AH: 1 0 0 1 0 0 0 0 (7 0)

After execution

Register 0: 0 1 1 0 1 0 1 0 (7 0)

6AH: 1 0 0 0 1 1 1 1 (7 0)

31. DEC A (Decrement accumulator)

Instruction code : 7 0

0	0	0	1	0	1	0	0

Byte 1

Operation : (A)←(A)–1

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

							•

Description : The accumulator contents are decremented by 1, and the flag is updated.

Example DEC A

Instruction code : 7 0

0	0	0	1	0	1	0	0

Byte 1

Before execution
Accumulator

1	0	1	0	1	0	0	0

7 0

After execution
Accumulator

1	0	1	0	0	1	1	1

7 0

32. DEC Rr (Decrement register)

Instruction code : 7 0 | 0 0 0 1 1 r2 r1 r0 | Byte 1

Operation : (Rr)←(Rr)–1 r=0 thru 7

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The register r contents are decremented by 1.

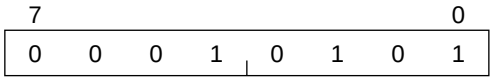
Example DEC R7

Instruction code : 7 0 | 0 0 0 1 1 1 1 1 | Byte 1

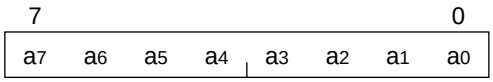
Before execution
Register 7
7 0 | 1 0 0 0 0 0 0 0 | 0

After execution
Register 7
7 0 | 0 1 1 1 1 1 1 1 | 0

33. DEC data address (Decrement memory)

Instruction code : 

Byte 1

Data address 

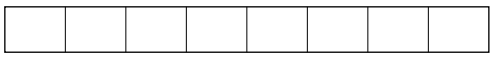
Byte 2

Operation : (data address)←(data address)–1

Number of bytes : 2

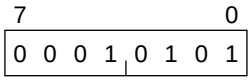
Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

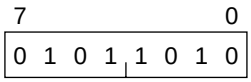
(PSW) 

Description : The specified data address contents are decremented by 1.

Example DEC 5AH

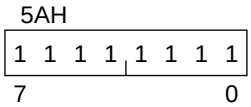
Instruction code : 

Byte 1

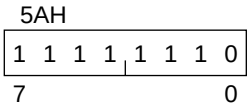


Byte 2

Before execution



After execution



34. DIV AB (Divide accumulator by B)

Instruction code : 7 0

1	0	0	0	0	1	0	0	Byte 1

Operation : (A) quotient←(A)/(B)
(B) remainder

Number of bytes : 1

Number of cycles : 4

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

•					•		•

Description : The accumulator contents are devided by the contents of arithmetic operation register (B). The two data values are handled as integers without sign. The quotient is placed in the accumulator, and the remainder in the arithmetic operation register (B). The carry flag is always cleared, and the overflow flag (OV) is set to 1 if division by 0 is executed. This flag is cleared in all other cases. If division by 0 is executed, the accumulator and arithmetic operation register (B) contents remain unchanged.

Example DIV AB(0AEH÷7H=18……remainder 6H)

Instruction code : 7 0

1	0	0	0	0	1	0	0	Byte 1

Before execution

Accumulator

1	0	1	0	1	1	1	0

7 0

B register

0	0	0	0	0	1	1	1

7 0

After execution

Accumulator

0	0	0	1	1	0	0	0

7 0

B register

0	0	0	0	0	1	1	0

7 0

35. DJNZ Rr, code address (Decrement register, and jump if not zero)

Instruction code :

7							0	
1	1	0	1	1	r_2	r_1	r_0	Byte 1

Relative offset

7							0	
R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0	Byte 2

Operations : (PC)←(PC)+2
(Rr)←(Rr)–1 r=0 thru 7
IF (Rr)≠0
THEN
(PC)←(PC)+relative offset

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The register r contents are decremented by 1. Control is shifted to a relative jump address if the register r contents are not 0 as a result of the decrement. Control is shifted to the next address following this instruction if the result is 0.

Example DJNZ R1, LOOP

LOC	OBJ	SOURCE
00FE	2F	LOOP:ADD A, R7
010B	D9F1	COUNT:DJNZ R1, LOOP

Instruction code :

	7	6	5	4	3	2	1	0	
Byte 1	1	1	0	1	1	0	0	1	
Byte 2	1	1	1	1	0	0	0	1	

Before execution

Register 1

7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	0

Program counter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1

After execution

Register 1

7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1

Program counter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0

36. DJNZ data address, code address (Decrement memory, and jump if not zero)

Instruction code : 1 1 0 1 0 1 0 1 Byte 1

Data address : a7 a6 a5 a4 a3 a2 a1 a0 Byte 2

Relative offset : R7 R6 R5 R4 R3 R2 R1 R0 Byte 3

Operations : (PC)←(PC)+3
(data address)←(data address)–1
IF (data address)≠0
THEN
(PC)←(PC)+relative offset

Number of bytes : 3

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

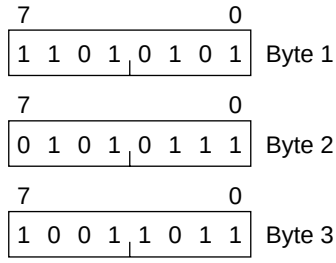
(PSW)

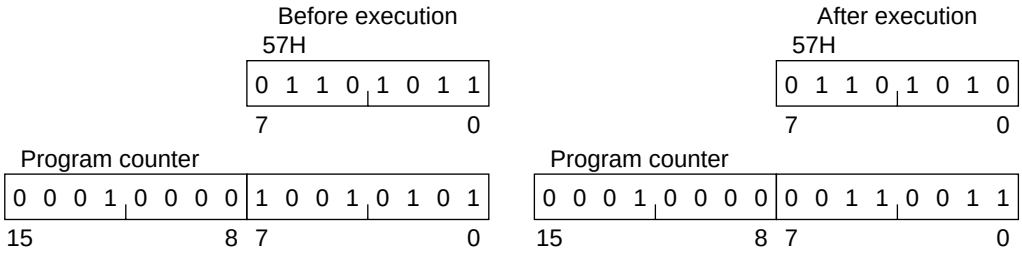
Description : The specified data address contents are decremented by 1. Control is shifted to a relative jump address if data address contents are not 0 as a result of the decrement. Control is shifted to the next address following this instruction if the result is 0.

Example DJNZ 57H, LOOP 1

LOC	OBJ	SOURCE
1033	A957	LOOP 1:MOV R1, 57H
1095	D5579B	COUNT:DJNZ 57H, LOOP 1

Instruction code :





37. INC @Rr (Increment indirect address)

Instruction code : 7 0 | 0 0 0 0 0 1 1 r | Byte 1

Operation : ((Rr))←((Rr))+1 r=0 or 1

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The contents of the data memory location addressed by the register r contents are incremented by 1.

Example INC @R1

Instruction code : 7 0 | 0 0 0 0 0 1 1 1 | Byte 1

Before execution

Register 1: 0 1 1 0 0 1 0 1 (7 0)

65H: 0 0 0 0 1 1 1 1 (7 0)

After execution

Register 1: 0 1 1 0 0 1 0 1 (7 0)

65H: 0 0 0 1 0 0 0 0 (7 0)

38. INC A (Increment accumulator)

Instruction code : 7 0 | 0 0 0 0 0 1 0 0 | Byte 1

Operation : (A)←(A)+1

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : | | | | | | | • |

Description : The accumulator contents are incremented by 1, and the flag is updated.

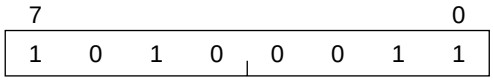
Example INC A

Instruction code : 7 0 | 0 0 0 0 0 1 0 0 | Byte 1

Before execution
Accumulator
1 0 1 1 0 1 1 1
7 0

After execution
Accumulator
1 0 1 1 1 0 0 0
7 0

39. INC DPTR (Increment data pointer)

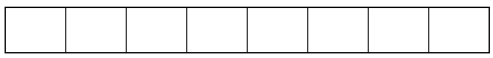
Instruction code : 

Operation : (DPTR)←(DPTR)+1

Number of bytes : 1

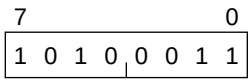
Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

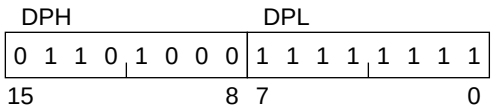
(PSW) : 

Description : 16-bit contents od the data pointer (DPH·DPL) are incremented by 1.

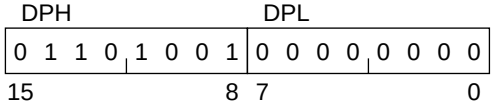
Example INC DPTR

Instruction code : 

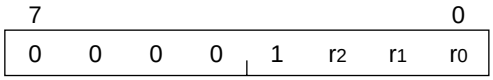
Before execution



After execution



40. INC Rr (Increment register)

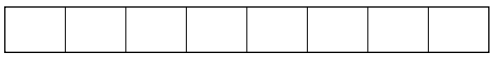
Instruction code : 

Operation : (Rr)←(Rr)+1 r=0 thru 7

Number of bytes : 1

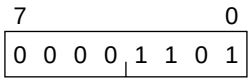
Number of cycles : 1

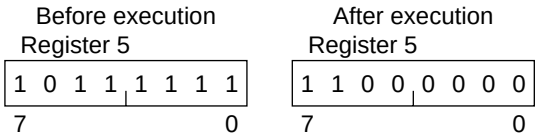
Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) 

Description : The register r contents are incremented by 1.

Example INC R5

Instruction code : 



41. INC data address (Increment memory)

Instruction code : 7 0 | 0 0 0 0 0 1 0 1 | Byte 1

Data address : 7 0 | a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 | Byte 2

Operation : (data address)←(data address)+1

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The specified data address contents are incremented by 1.

Example INC P1

Instruction code : 7 0 | 0 0 0 0 0 1 0 1 | Byte 1

Data address : 7 0 | 1 0 0 1 0 0 0 0 | Byte 2

Before execution — Port 1: 0 0 0 0 1 1 1 1 (7 … 0)

After execution — Port 1: 0 0 0 1 0 0 0 0 (7 … 0)

42. JB bit address, code address (Jump if bit is set)

Instruction code : 7 0 | 0 0 1 0 0 0 0 0 | Byte 1

Bit address : 7 0 | b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 | Byte 2

Relative offset : 7 0 | R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0 | Byte 3

Operations : (PC)←(PC)+3
IF (bit address)=1
THEN
(PC)←(PC)+relative offset

Number of bytes : 3

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

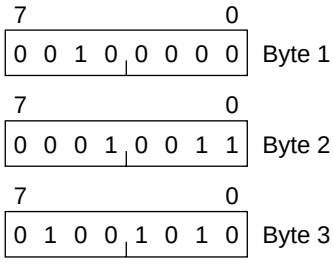
(PSW) : | | | | | | | |

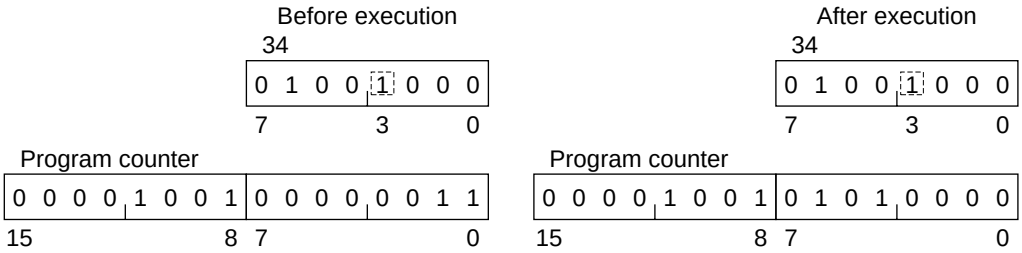
Description : Control is shifted to a relative jump address if the specified bit address content is 1.
Control is shifted to the next address following this instruction if the content is 0.

Example JB 34.3, ENTER

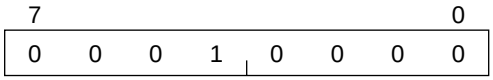
LOC	OBJ	SOURCE
0903	20134A	BITTS:JB 34.3, ENTER
0950	ACA0	ENTER:MOV R4, 0A0H

Instruction code :

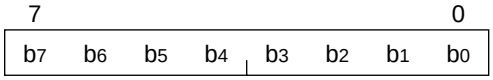




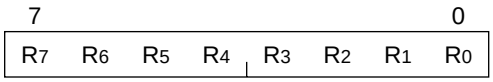
43. JBC bit address, code address (Jump and clear if bit is set)

Instruction code : 

Byte 1

Bit address 

Byte 2

Relative offset 

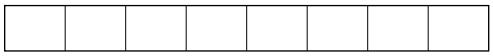
Byte 3

Operations : (PC)←(PC)+3
IF (bit address)=1
THEN
(bit address)←0
(PC)←(PC)+relative offset

Number of bytes : 3

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) 

Description : Control is shifted to a relative jump address if the specified bit address content is 1, and that bit is cleared to 0.
Control is shifted to the next address following this instruction if the content is 0.

Example JBC 46.1, COUNT 4

LOC	OBJ	SOURCE
00DC	C281	COUNT 4:CLR 128.1
0136	1071A3	BTEST:JBC46.1, COUNT 4

Instruction code :

	7							0	
Byte 1	0	0	0	1	0	0	0	0	
Byte 2	0	1	1	1	0	0	0	1	
Byte 3	1	0	1	0	0	0	1	1	

Before execution

46								
	1	0	1	0	1	0	1	0
	7						1	0

Program counter

15							8	7							0
	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	0

After execution

46								
	1	0	1	0	1	0	0	0
	7						1	0

Program counter

15							8	7							0
	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0

44. JC code address (Jump if carry is set)

Instruction code : Byte 1

7							0
0	1	0	0	0	0	0	0

Relative offset : Byte 2

7							0
R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0

Operations : (PC)←(PC)+2
IF (C)=1
THEN
(PC)←(PC)+relative offset

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : Control is shifted to a relative jump address if the carry flag is 1. Control is shifted to the next address following this instruction if the content is 0.

Example JC CARRY

LOC	OBJ	SOURCE
16DC	7110	CHECK:ACALL ADDR
16DE	4015	JMPC:JC CARRY
16F5	07	CARRY:INC @R1

Instruction code :

7							0	
0	1	0	0	0	0	0	0	Byte 1

7							0	
0	0	0	1	0	1	0	1	Byte 2

Before execution

Carry flag: 1

Program counter

15								8	7							0
0	0	0	1	0	1	1	0		1	1	0	1	1	1	1	0

After execution

Carry flag: 1

Program counter

15								8	7							0
0	0	0	1	0	1	1	0		1	1	1	1	0	1	0	1

45. JMP @A + DPTR (Jump to sum of accumulator and data pointer)

Instruction code : 7 0 | 0 1 1 1 0 0 1 1 | Byte 1

Operation : (PC)←(A)+(DPTR)

Number of bytes : 1

Number of cycles : 2

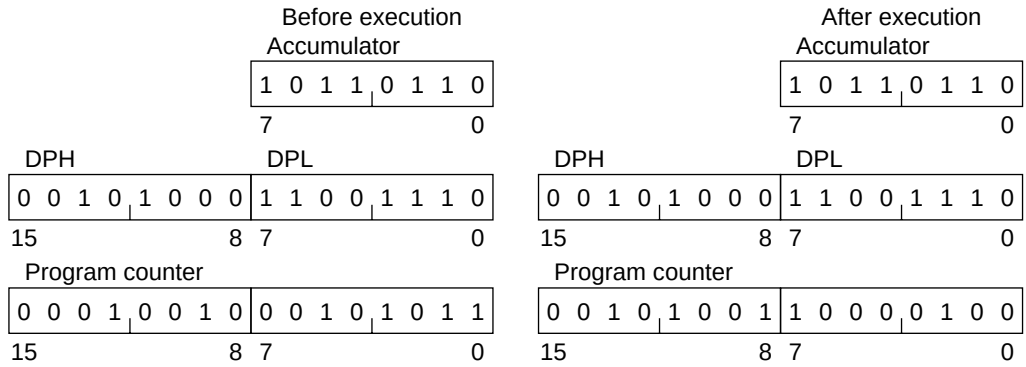
Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The accumulator contents are added to the data pointer contents, and the resulting sum is placed in the program counter.

Example JMP @A+DPTR

Instruction code : 7 0 | 0 1 1 1 0 0 1 1 | Byte 1



46. JNB bit address, code address (Jump if bit is not set)

Instruction code :

7							0	
0	0	1	1	0	0	0	0	Byte 1

Bit address

7							0	
b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Byte 2

Relative offset

7							0	
R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0	Byte 3

Operations : (PC)←(PC)+3
IF (bit address)=0
THEN
(PC)←(PC)+relative offset

Number of bytes : 3

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

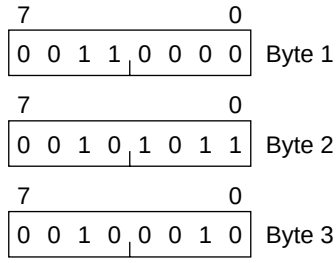
(PSW)

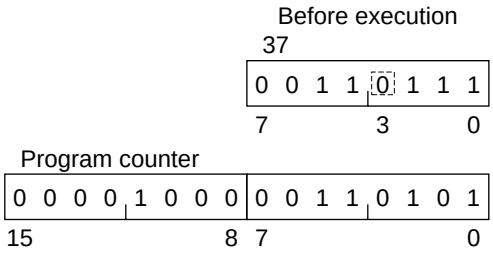
Description : Control is shifted to a relative jump address if the specified bit address content is 0, but shifted to the next address following this instruction if the content is 1.

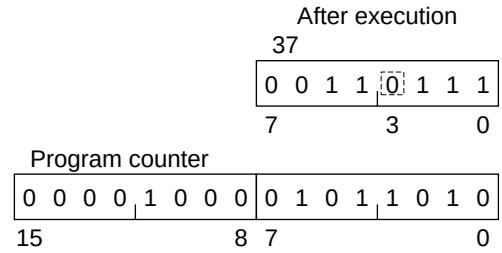
Example JNB 37.3, EXIT

LOC	OBJ	SOURCE
0835	302B22	TEST:JNB 37.3, EXIT
085A	E6	EXIT:MOV A, @R0

Instruction code :







47. JNC code address (Jump if carry is not set)

Instruction code : Byte 1

7							0
0	1	0	1	0	0	0	0

Relative offset : Byte 2

7							0
R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0

Operations : (PC)←(PC)+2
IF (C)=0
THEN
(PC)←(PC)+relative offset

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

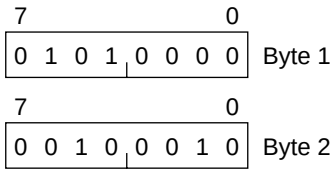
(PSW)

Description : Control is shifted to a relative jump address if the carry flag is 0. Control is shifted to the next address following this instruction if the content is 1.

Example JNC EXIT

LOC	OBJ	SOURCE
0835	5022	TEST:JNC EXIT
0859	85E0F0	EXIT:MOV B, ACC

Instruction code :



Before execution

Carry flag: 0

Program counter: 0 0 0 0 1 0 0 0 | 0 0 1 1 0 1 0 1 (15 8 7 0)

After execution

Carry flag: 0

Program counter: 0 0 0 0 1 0 0 0 | 0 1 0 1 1 0 0 1 (15 8 7 0)

48. JNZ code address (Jump if accumulator is not 0)

Instruction code : Byte 1

7							0
0	1	1	1	0	0	0	0

Relative offset Byte 2

7							0
R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0

Operations : (PC)←(PC)+2
IF (A)≠0
THEN
(PC)←(PC)+relative offset

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : Control is shifted to a relative jump address if the accumulator contents are not 0. Control is shifted to the next address following this instruction if the contents are 0.

Example JNZ TEST

LOC	OBJ	SOURCE
00FC	7030	CHECK:JNZ TEST
012E	FB	TEST:MOV R3, A

Instruction code :

7							0	
0	1	1	1	0	0	0	0	Byte 1

7							0	
0	0	1	1	0	0	0	0	Byte 2

Before execution

Accumulator

7							0
0	1	0	1	1	1	0	1

Program counter

15							8	7							0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0

After execution

Accumulator

7							0
0	1	0	1	1	1	0	1

Program counter

15							8	7							0
0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	0

49. JZ code address (Jump if accumulator is not 0)

Instruction code : Byte 1

7							0
0	1	1	0	0	0	0	0

Relative offset : Byte 2

7							0
R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0

Operations : (PC)←(PC)+2
IF (A)=0
THEN
(PC)←(PC)+relative offset

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : Control is shifted to a relative jump address if the accumulator contents are 0. Control is shifted to the next address following this instruction if the contents are not 0.

Example JZ EMPTY

LOC	OBJ	SOURCE
0099	04	EMPTY:INC A
00CA	60CD	CHECK:JZ EMPTY

Instruction code :

7							0	
0	1	1	0	0	0	0	0	Byte 1

7							0	
1	1	0	0	1	1	0	1	Byte 2

Before execution

Accumulator

7							0
0	0	0	0	0	0	0	0

Program counter

15							8	7							0
0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0

After execution

Accumulator

7							0
0	0	0	0	0	0	0	0

Program counter

15							8	7							0
0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1

50. LCALL code address (Long call)

Instruction code : 7 0 | 0 0 0 1 0 0 1 0 | Byte 1

Call address : 7 0 | A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 | Byte 2

Call address : 7 0 | A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0 | Byte 3

Operations : (PC)←(PC)+3
(SP)←(SP)+1
((SP))←($PC_{0\sim7}$)
(SP)←(SP)+1
((SP))←($PC_{8\sim15}$)
($PC_{0\sim15}$)←$A_{0\sim15}$

Number of bytes : 3

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : | | | | | | | |

Description : The contents of the program counter (return address) are pushed in the stack following an increment.
Call address $A_{0\sim15}$ specified by operand are placed in the program counter $PC_{0\sim15}$.
This instruction is capable of call to anywhere within the entire range of 64K words.

51. LJMP code address (Long jump)

		7							0	
Instruction code	:	0	0	0	0	0	0	1	0	Byte 1
		7							0	
Jump address		A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	Byte 2
		7							0	
Jump address		A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Byte 3

Operation : $(PC_{0\sim15}) \leftarrow A_{0\sim15}$

Number of bytes : 3

Number of cycles : 2

Flags	:	C	AC	F0	RS1	RS0	OV	F1	P
(PSW)									

Description : Jump address $A_{0\sim15}$ specified by operand are placed in the program counter $PC_{0\sim15}$.
This instruction is capable of jump to anywhere within the entire range of 64K words.

52. MOV @Rr, #data (Move immediate data to indirect address)

Instruction code :

7							0	
0	1	1	1	0	1	1	r	Byte 1

Data address

7							0	
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Byte 2

Operation : ((Rr))←#data r=0 or 1

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : An 8-bit immediate data value is copied to the data memory location addressed by the register r contents.

Example MOV @R1, #0AAH

Instruction code :

7							0	
0	1	1	1	0	1	1	1	Byte 1
1	0	1	0	1	0	1	0	Byte 2

Before execution

Register 1

7							0
0	1	1	0	1	0	1	0

6AH

7							0
0	1	1	1	0	1	1	1

After execution

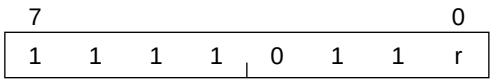
Register 1

7							0
0	1	1	0	1	0	1	0

6AH

7							0
1	0	1	0	1	0	1	0

53. MOV @Rr, A (Move accumulator to indirect address)

Instruction code : 

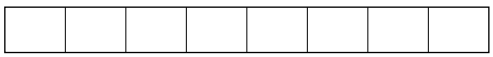
Byte 1

Operation : ((Rr))←(A) r=0 or 1

Number of bytes : 1

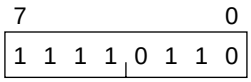
Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

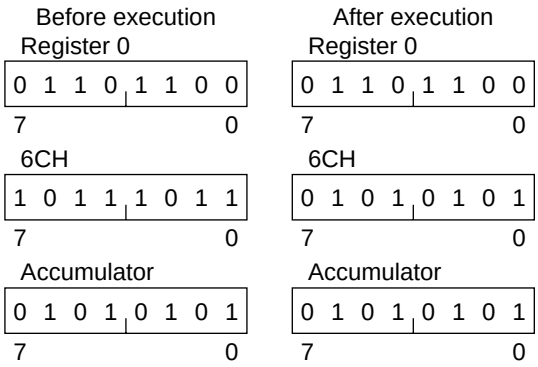
(PSW) 

Description : The accumulator contents are copied to the data memory location addressed by the register r contents.

Example MOV @R0, A

Instruction code : 

Byte 1



54. MOV @Rr, data address (Move memory to indirect address)

Instruction code :

7							0	
1	0	1	0	0	1	1	r	Byte 1

Data address

7							0	
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2

Operation : ((Rr))←(data address) r=0 or 1

Number of bytes : 2

Number of cycles : 2

Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P

Description : The specified data address contents are copied to the data memory location addressed by the register r contents.

Example MOV @R0, 0E0H

Instruction code :

7							0	
1	0	1	0	0	1	1	0	Byte 1

7							0	
1	1	1	0	0	0	0	0	Byte 2

Before execution								
Accumulator								
0	1	0	1	0	1	1	1	
7							0	
Register 0								
0	1	1	1	0	0	1	0	
7							0	
72H								
0	0	1	1	1	1	0	0	
7							0	

After execution								
Accumulator								
0	1	0	1	0	1	1	1	
7							0	
Register 0								
0	1	1	1	0	0	1	0	
7							0	
72H								
0	1	0	1	0	1	1	1	
7							0	

55. MOV A, #data (Move immediate data to accumulator)

Instruction code : 7 0

0	1	1	1	0	1	0	0	Byte 1

#data 7 0

I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Byte 2

Operation : (A)←#data

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

							•

Description : An 8-bit immediate data is copied to the accumulator, and the flag is updated.

Example MOV A, #05H

Instruction code : 7 0

0	1	1	1	0	1	0	0	Byte 1

7 0

0	0	0	0	0	1	0	1	Byte 2

Before execution
Accumulator

0	1	1	1	0	1	1	1

7 0

After execution
Accumulator

0	0	0	0	0	1	0	1

7 0

56. MOV A, @Rr (Move indirect address to accumulator)

Instruction code : 7 6 5 4 3 2 1 0 — 1 1 1 0 0 1 1 r — Byte 1

Operation : (A)←((Rr)) r=0 or 1

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • (P)

Description : The data memory location contents addressed by the register r contents are copied to the accumulator, and the flag is updated.

Example MOV A, @R0

Instruction code : 7 6 5 4 3 2 1 0 — 1 1 1 0 0 1 1 0 — Byte 1

Before execution

Register 0: 0 1 1 1 0 0 1 0 (7 … 0)

72H: 1 0 1 1 0 1 1 1 (7 … 0)

Accumulator: 0 1 0 0 1 1 0 0 (7 … 0)

After execution

Register 0: 0 1 1 1 0 0 1 0 (7 … 0)

72H: 1 0 1 1 0 1 1 1 (7 … 0)

Accumulator: 1 0 1 1 0 1 1 1 (7 … 0)

57. MOV A, Rr (Move register to accumulator)

Instruction code : 7 0

1	1	1	0	1	r2	r1	r0

Byte 1

Operation : (A)←(Rr) r=0 thru 7

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

							•

Description : The register r contents are copied to the accumulator, and the flag is updated.

Example MOV A, R6

Instruction code : 7 0

1	1	1	0	1	1	1	0

Byte 1

Before execution

Register 6

1	0	1	0	0	1	0	1

7 0

Accumulator

0	0	0	0	1	0	1	1

7 0

After execution

Register 6

1	0	1	0	0	1	0	1

7 0

Accumulator

1	0	1	0	0	1	0	1

7 0

58. MOV A, data address (Move memory to accumulator)

Instruction code : Byte 1

7							0
1	1	1	0	0	1	0	1

Data address Byte 2

7							0
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Operation : (A)←(data address)

Number of bytes : 2

Number of cycles : 1

Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P
							•

Description : The specified data address contents are copied to the accumulator, and the flag is updated.

Example MOV A, P1

Instruction code : Byte 1

7							0
1	1	1	0	0	1	0	1

Byte 2

7							0
1	0	0	1	0	0	0	0

Before execution

Port 1

7							0
0	1	1	0	0	1	1	1

Accumulator

7							0
0	1	0	0	0	0	1	0

After execution

Port 1

7							0
0	1	1	0	0	1	1	1

Accumulator

7							0
0	1	1	0	0	1	1	1

59. MOV C, bit address (Move bit to carry flag)

Instruction code : 7 0 — 1 0 1 0 0 0 1 0 — Byte 1

Bit address — 7 0 — b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 — Byte 2

Operation : (C)←(bit address)

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • | | | | | | |

Description : The specified bit address content is copied to the carry flag.

Example MOV C, P3.4

Instruction code : 7 0 — 1 0 1 0 0 0 1 0 — Byte 1

7 0 — 1 0 1 1 0 1 0 0 — Byte 2

Before execution

Port 3: 0 0 0 [1] 0 1 1 0 (7 4 0)

Carry flag: 0

After execution

Port 3: 0 0 0 [1] 0 1 1 0 (7 4 0)

Carry flag: 1

60. MOV DPTR, #data (Move immediate data to data pointer)

Instruction code : 1 0 0 1 0 0 0 0 Byte 1

#data : I_{15} I_{14} I_{13} I_{12} I_{11} I_{10} I_9 I_8 Byte 2

#data : I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0 Byte 3

Operation : (DPTR)←#data
(DPH)←$I_{8\sim15}$
(DPL)←$I_{0\sim7}$

Number of bytes : 3

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : A 16-bit immediate data value is copied to the data pointer (DPH·DPL).

Example MOV DPTR, #0AF5H

Instruction code :
1 0 0 1 0 0 0 0 Byte 1
0 0 0 0 1 0 1 0 Byte 2
1 1 1 1 0 1 0 1 Byte 3

Before execution

DPH								DPL							
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
15							8	7							0

After execution

DPH								DPL							
0	0	0	0	1	0	1	0	1	1	1	1	0	1	0	1
15							8	7							0

61. MOV Rr, #data (Move immediate data to register)

Instruction code : 7 0 | 0 1 1 1 1 r2 r1 r0 | Byte 1

#data 7 0 | I7 I6 I5 I4 I3 I2 I1 I0 | Byte 2

Operation : (Rr)←#data r=0 thru 7

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : An 8-bit immediate data value is copied to the register r.

Example MOV R5, #0AH

Instruction code :

7 0 | 0 1 1 1 1 1 0 1 | Byte 1

7 0 | 0 0 0 0 1 0 1 0 | Byte 2

Before execution — Register 5: 7 0 | 1 0 1 0 1 0 1 1

After execution — Register 5: 7 0 | 0 0 0 0 1 0 1 0

62. MOV Rr, A (Move accumulator to register)

Instruction code : 7 0

1	1	1	1	1	r2	r1	r0

Byte 1

Operation : (Rr)←(A) r=0 thru 7

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The accumulator contents are copied to the register r.

Example MOV R1, A

Instruction code : 7 0

1	1	1	1	1	0	0	1

Byte 1

Before execution

Register 1

0	1	1	1	1	1	1	0

7 0

Accumulator

1	0	0	1	1	0	0	1

7 0

After execution

Register 1

1	0	0	1	1	0	0	1

7 0

Accumulator

1	0	0	1	1	0	0	1

7 0

63. MOV Rr, data address (Move memory to register)

Instruction code : 7 0

1	0	1	0	1	r2	r1	r0	Byte 1

Data address 7 0

a7	a6	a5	a4	a3	a2	a1	a0	Byte 2

Operation : (Rr)←(data address) r=0 thru 7

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The specified data address contents are copied to the register r.

Example MOV R0, 5AH

Instruction code : 7 0

1	0	1	0	1	0	0	0	Byte 1

7 0

0	1	0	1	1	0	1	0	Byte 2

Before execution

Register 0

0	1	1	1	1	0	1	1

7 0

5AH

1	0	1	0	1	0	1	0

7 0

After execution

Register 0

1	0	1	0	1	0	1	0

7 0

5AH

1	0	1	0	1	0	1	0

7 0

64. MOV bit address, C (Move carry flag to bit)

Instruction code : 7 0 — 1 0 0 1 0 0 1 0 — Byte 1

Bit address — b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 — Byte 2

Operation : (bit address)←(C)

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The carry flag content is copied to the specified bit address.

Example MOV P1.4, C

Instruction code :

Byte 1: 1 0 0 1 0 0 1 0

Byte 2: 1 0 0 1 0 1 0 0

Before execution

Port 1: 1 1 1 [1] 1 1 1 1 (7 4 0)

Carry flag: 0

After execution

Port 1: 1 1 1 [0] 1 1 1 1 (7 4 0)

Carry flag: 0

65. MOV data address, #data (Move immediate data to memory)

Instruction code : Byte 1

7							0
0	1	1	1	0	1	0	1

Data address Byte 2

7							0
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

#data Byte 3

7							0
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Operation : (data address)←#data

Number of bytes : 3

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : An 8-bit immediate data value is copied to the specified data address.

Example MOV TCON, #50H

Instruction code :

Byte 1

7							0
0	1	1	1	0	1	0	1

Byte 2

7							0
1	0	0	0	1	0	0	0

Byte 3

7							0
0	1	0	1	0	0	0	0

Before execution
TCON(88H)

7							0
0	0	0	0	0	0	0	0

After execution
TCON(88H)

7							0
0	1	0	1	0	0	0	0

66. MOV data address, @Rr (Move indirect address to memory)

Instruction code : 7 0 | 1 0 0 0 0 1 1 r | Byte 1

Data address : 7 0 | a7 a6 a5 a4 a3 a2 a1 a0 | Byte 2

Operation : (data address)←((Rr)) r=0 or 1

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The data memory location contents addressed by the register r contents are copied to the specified data address.

Example MOV ACC, @R1

Instruction code : 7 0 | 1 0 0 0 0 1 1 1 | Byte 1

7 0 | 1 1 1 0 0 0 0 0 | Byte 2

Before execution

Accumulator: 0 0 0 0 0 0 0 0 (7 0)

Register 1: 0 0 1 0 0 1 0 1 (7 0)

25H: 0 1 1 0 1 1 1 1 (7 0)

After execution

Accumulator: 0 1 1 0 1 1 1 1 (7 0)

Register 1: 0 0 1 0 0 1 0 1 (7 0)

25H: 0 1 1 0 1 1 1 1 (7 0)

67. MOV data address, A (Move accumulator to memory)

Instruction code : Byte 1

7							0
1	1	1	1	0	1	0	1

Data address Byte 2

7							0
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Operation : (data address)←(A)

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The accumulator contents are copied to the specified data address.

Example MOV P3, A

Instruction code : Byte 1

7							0
1	1	1	1	0	1	0	1

Byte 2

7							0
1	0	1	1	0	0	0	0

Before execution

Port 3

7							0
1	1	1	1	1	1	1	1

Accumulator

7							0
1	1	1	0	1	1	0	0

After execution

Port 3

7							0
1	1	1	0	1	1	0	0

Accumulator

7							0
1	1	1	0	1	1	0	0

68. MOV data address, Rr (Move register to memory)

Instruction code :

7							0	
1	0	0	0	1	r_2	r_1	r_0	Byte 1

Data address

7							0	
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2

Operation : (data address)←(Rr) r=0 thru 7

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The register r contents are copied to the specified data address.

Example MOV 6BH, R2

Instruction code :

7							0	
1	0	0	0	1	0	1	0	Byte 1

7							0	
0	1	1	0	1	0	1	1	Byte 2

Before execution

6BH

7							0
1	0	1	1	0	1	1	0

Register 2

7							0
0	1	0	1	0	1	0	1

After execution

6BH

7							0
0	1	0	1	0	1	0	1

Register 2

7							0
0	1	0	1	0	1	0	1

69. MOV data address 1, data address 2 (Move memory to memory)

Instruction code : 1 0 0 0 0 1 0 1 Byte 1

Data address 2 : a_7^2 a_6^2 a_5^2 a_4^2 a_3^2 a_2^2 a_1^2 a_0^2 Byte 2

Data address 1 : a_7^1 a_6^1 a_5^1 a_4^1 a_3^1 a_2^1 a_1^1 a_0^1 Byte 3

Operation : (data address 1)←(data address 2)

Number of bytes : 3

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The source data address (data address 2) contents are copied to the destination data address (data address 1).

Example MOV ACC, P1

Instruction code :

1 0 0 0 0 1 0 1 Byte 1

1 0 0 1 0 0 0 0 Byte 2

1 1 1 0 0 0 0 0 Byte 3

Before execution

Port 1: 1 0 1 1 0 1 0 0

Accumulator: 0 1 1 1 1 0 1 1

After execution

Port 1: 1 0 1 1 0 1 0 0

Accumulator: 1 0 1 1 0 1 0 0

70. MOVC A, @A + DPTR
(Move code memory offset from data pointer to accumulator)

Instruction code : 7 0 — 1 0 0 1 0 0 1 1 — Byte 1

Operation : (A)←((A)+(DPTR))

Number of bytes : 1

Number of cycles : 2

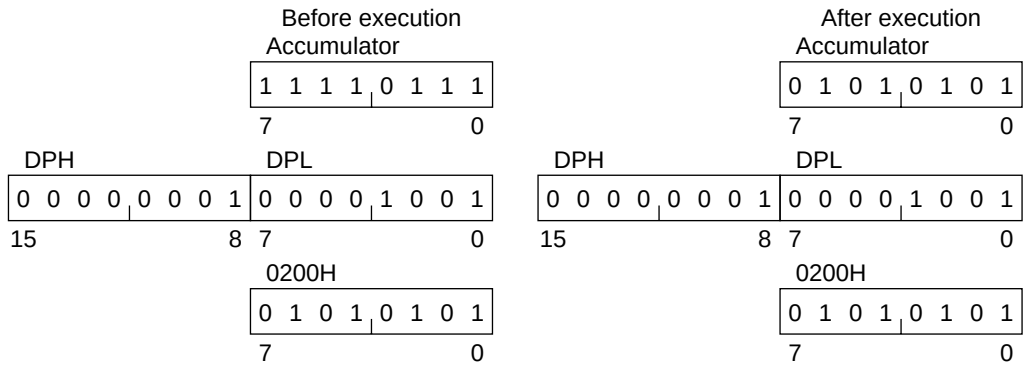
Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • (P)

Description : The data pointer contents are added to the accumulator contents, and after temporary storage of the sum in the program counter, the ROM data contents specified by the program counter are stored in the accumulator. The program counter contents are then restored to former contents, and the flag is updated.

Example MOVC A, @A+DPTR

Instruction code : 7 0 — 1 0 0 1 0 0 1 1 — Byte 1



71. MOVC A, @A + PC
(Move code memory offset from program counter to accumulator)

Instruction code : 7 0

1	0	0	0	0	0	1	1

Byte 1

Operations : (PC)←(PC)+1
(A)←((A)+(PC))

Number of bytes : 1

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

							•

Description : The program counter contents following an increment are added to the accumulator contents, and after temporary storage of the sum in the program counter, the ROM data contents specified by the program counter are stored in the accumulator. The program counter contents are then restored to former contents, and the flag is also updated.

Example MOVC A, @A+PC

Instruction code : 7 0

1	0	0	0	0	0	1	1

Byte 1

Before execution

Accumulator

1	1	1	0	0	0	0	0

7 0

Program counter

0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0

15 8 7 0

0301H

1	1	1	0	1	1	1	0

7 0

After execution

Accumulator

1	1	1	0	1	1	1	0

7 0

Program counter

0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1

15 8 7 0

0301H

1	1	1	0	1	1	1	0

7 0

72. MOVX @DPTR, A
(Move accumulator to external memory addressed by data pointer)

Instruction code : 7 6 5 4 3 2 1 0 — 1 1 1 1 0 0 0 0 — Byte 1

Operation : ((DPTR))←(A)

Number of bytes : 1

Number of cycles : 2

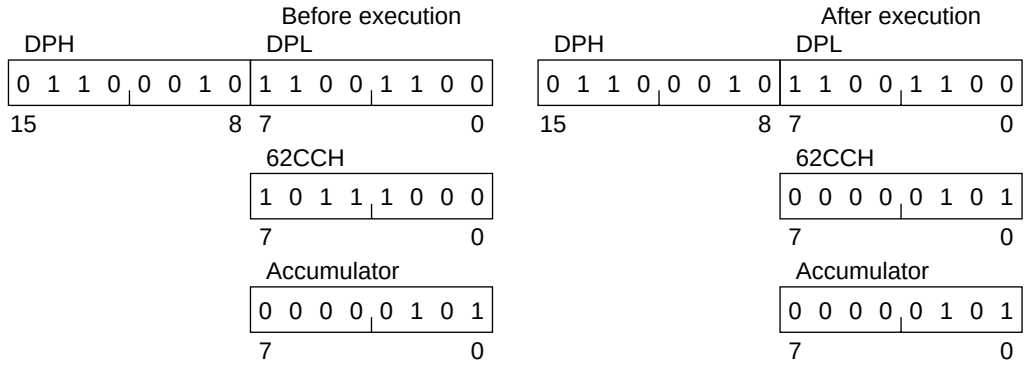
Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

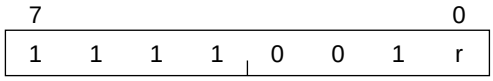
Description : The accumulator contents are stored in external data memory (RAM) addressed by the data pointer contents.

Example MOVX @DPTR, A

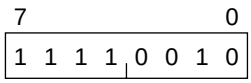
Instruction code : 7 6 5 4 3 2 1 0 — 1 1 1 1 0 0 0 0 — Byte 1

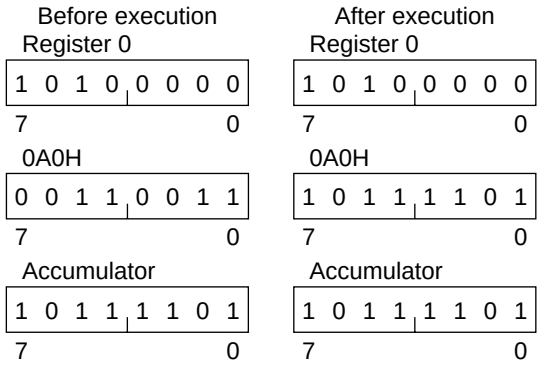


73. MOVX @Rr, A
(Move accumulator to external memory addressed by register)

Instruction code	:	 Byte 1
Operation	:	((Rr))←(A) r=0 or 1
Number of bytes	:	1
Number of cycles	:	2
Flags	:	C AC F0 RS1 RS0 OV F1 P
(PSW)		
Description	:	The accumulator contents are stored in external data memory addressed by the register r contents.

Example MOVX @R0, A

Instruction code :  Byte 1

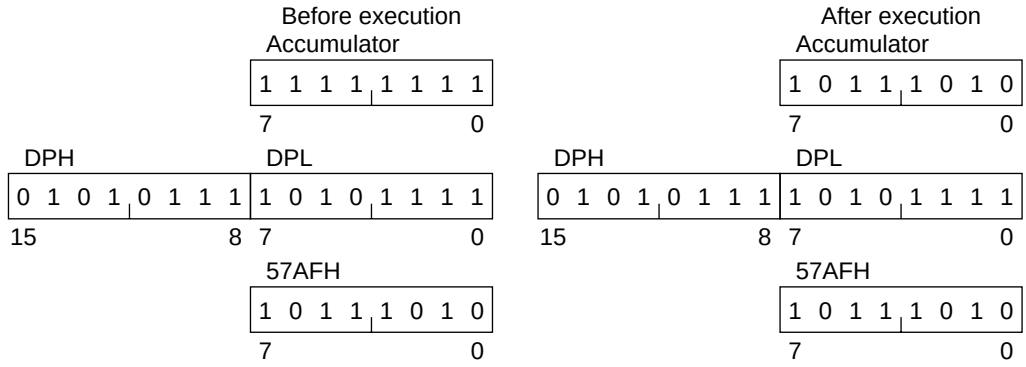


74. MOVX A, @DPTR
(Move external memory addressed by data pointer to accumulator)

Instruction code	:	7 6 5 4 3 2 1 0 = 1 1 1 0 0 0 0 0 Byte 1
Operation	:	(A)←((DPTR))
Number of bytes	:	1
Number of cycles	:	2
Flags	:	C AC F0 RS1 RS0 OV F1 P
(PSW)		•
Description	:	External data memory (RAM) contents addressed by the data pointer are stored in the accumulator, and the flag is updated.

Example MOVX A, @DPTR

Instruction code : 7 6 5 4 3 2 1 0 = 1 1 1 0 0 0 0 0 Byte 1



75. MOVX A, @Rr (Move external memory addressed by register to accumulator)

Instruction code :

7							0	
1	1	1	0	0	0	1	r	Byte 1

Operation : (A)←((Rr)) r=0 or 1

Number of bytes : 1

Number of cycles : 2

Flags :

C	AC	F0	RS1	RS0	OV	F1	P
							•

(PSW)

Description : External data memory (RAM) contents addressed by the register r contents are stored in the accumulator, and the flag is updated.

Example MOVX A, @R1

Instruction code :

7							0	
1	1	1	0	0	0	1	1	Byte 1

Before execution								
Accumulator								
0	1	1	1	1	0	1	0	
7							0	
Register 1								
1	0	1	1	1	1	1	0	
7							0	
0BEH								
0	0	1	0	1	0	0	0	
7							0	

After execution								
Accumulator								
0	0	1	0	1	0	0	0	
7							0	
Register 1								
1	0	1	1	1	1	1	0	
7							0	
0BEH								
0	0	1	0	1	0	0	0	
7							0	

76. MUL AB (Multiply accumulator by B)

Instruction code : 7 0

1	0	1	0	0	1	0	0

Byte 1

Operations : $(A)_{0\sim7} \leftarrow (A) \times (B)$
$(B)_{8\sim15}$

Number of bytes : 1

Number of cycles : 4

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

•					•		•

Description : The accumulator contents are multiplied by the arithmetic operation register (B) contents. The operand is always handled as an integer without sign. The lower order byte of the result is placed in the accumulator, and the higher order byte is placed in the arithmetic operation register (B). The carry flag is always cleared. The overflow flag is set to 1 if the product is greater than 00FFH, and to 0 in all other cases.

Example MUL AB(6AH × 15H=8B2H)

Instruction code : 7 0

1	0	1	0	0	1	0	0

Byte 1

Before execution

Accumulator

0	1	1	0	1	0	1	0

7 0

Register B

0	0	0	1	0	1	0	1

7 0

Overflow flag

0

After execution

Accumulator

1	0	1	1	0	0	1	0

7 0

Register B

0	0	0	0	1	0	0	0

7 0

Overflow flag

1

77. NOP (No operation)

		7							0
Instruction code	:	0	0	0	0	0	0	0	0	Byte 1

Operation : (PC)←(PC)+1

Number of bytes : 1

Number of cycles : 1

Flags	:	C	AC	F0	RS1	RS0	OV	F1	P
(PSW)									

Description : The program counter is incremented by 1 without any other change in the CPU. Control is shifted to the next instruction.

78. ORL A, #data (Logical OR immediate data to accumulator)

Instruction code : 7 0 | 0 1 0 0 0 1 0 0 | Byte 1

#data : 7 0 | I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0 | Byte 2

Operation : (A)←(A) OR #data

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : | | | | | | | • |

Description : The logical OR between an 8-bit immediate data value and the accumulator contents is determined. The result is placed in the accumulator and the flag is updated.

Example ORL A, #5FH

Instruction code :

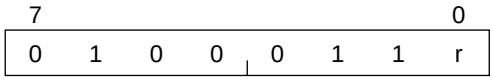
7 0 | 0 1 0 0 0 1 0 0 | Byte 1

7 0 | 0 1 0 1 1 1 1 1 | Byte 2

Before execution
Accumulator
1 0 0 0 0 1 0 0
7 0

After execution
Accumulator
1 1 0 1 1 1 1 1
7 0

79. ORL A, @Rr (Logical OR indirect address to accumulator)

Instruction code : 

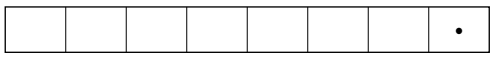
Byte 1

Operation : (A)←(A) OR ((Rr)) r=0 or 1

Number of bytes : 1

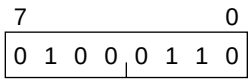
Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

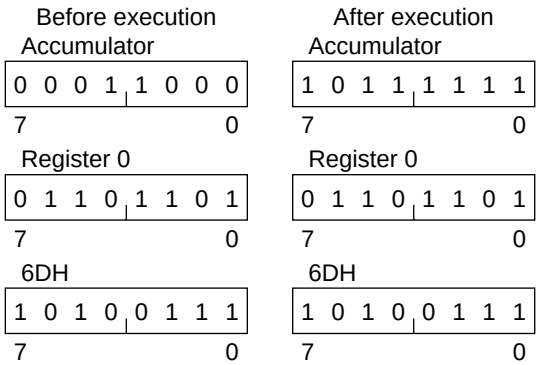
(PSW) : 

Description : The logical OR between the accumulator contents and the data memory location contents addressed by the register r contents is determined. The result is placed in the accumulator and the flag is updated.

Example ORL A, @R0

Instruction code : 

Byte 1



80. ORL A, Rr (Logical OR register to accumulator)

Instruction code : 7 0 | 0 1 0 0 1 r2 r1 r0 | Byte 1

Operation : (A)←(A) OR (Rr) r=0 thru 7

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : | | | | | | | • |

Description : The logical OR between the accumulator contents and the register r contents is determined. The result is placed in the accumulator and the flag is updated.

Example ORL A, R5

Instruction code : 7 0 | 0 1 0 0 1 1 0 1 | Byte 1

Before execution

Accumulator: 0 0 0 0 0 0 0 0 (7 0)

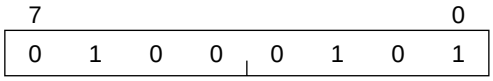
Register 5: 0 1 1 1 0 1 0 1 (7 0)

After execution

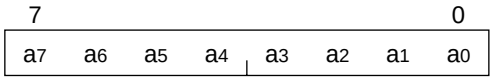
Accumulator: 0 1 1 1 0 1 0 1 (7 0)

Register 5: 0 1 1 1 0 1 0 1 (7 0)

81. ORL A, data address (Logical OR memory to accumulator)

Instruction code : 

Byte 1

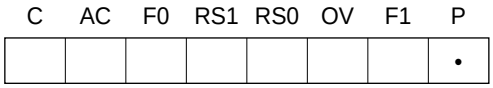
Data address 

Byte 2

Operation : (A)←(A) OR (data address)

Number of bytes : 2

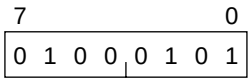
Number of cycles : 1

Flags : 

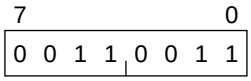
(PSW)

Description : The logical OR between the accumulator contents and the specified data address contents is determined. The result is placed in the accumulator and the flag is updated.

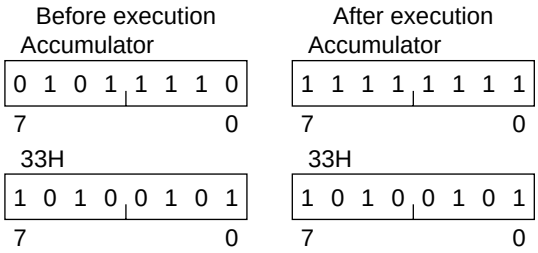
Example ORL A, 33H

Instruction code : 

Byte 1



Byte 2



82. ORL C, bit address (Logical OR bit to carry flag)

Instruction code : 7 0 | 0 1 1 1 0 0 1 0 | Byte 1

Bit address : 7 0 | b7 b6 b5 b4 b3 b2 b1 b0 | Byte 2

Operation : (C)←(C) OR (bit address)

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • | | | | | | | |

Description : The logical OR between the carry flag and the specified bit address content is determined. The result is placed in the carry flag.

Example ORL C, ACC.6

Instruction code :

7 0 | 0 1 1 1 0 0 1 0 | Byte 1

7 0 | 1 1 1 0 0 1 1 0 | Byte 2

Before execution
Carry flag: 0

After execution
Carry flag: 1

Accumulator: 0 1 1 1 0 0 1 0 (7 6 ... 0)

Accumulator: 0 1 1 1 0 0 1 0 (7 6 ... 0)

83. ORL C,/bit address (Logical OR complement of bit to carry flag)

Instruction code : 1 0 1 0 0 0 0 0 Byte 1

Bit address : b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 Byte 2

Operation : (C)←(C) OR ($\overline{\text{bit address}}$)

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • | | | | | | |

Description : The logical OR between the carry flag and the complement of specified bit address content is determined. The result is placed in the carry flag.

Example ORL C,/25H.5

Instruction code :

Byte 1: 1 0 1 0 0 0 0 0

Byte 2: 0 0 1 0 1 1 0 1

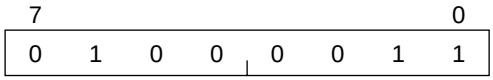
Before execution — Carry flag: 0

After execution — Carry flag: 1

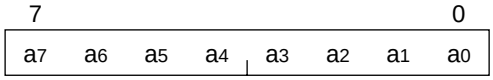
25H: 1 0 0 0 1 0 1 0 (bit 5)

25H: 1 0 0 0 1 0 1 0 (bit 5)

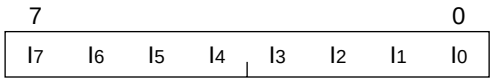
84. ORL data address, #data (Logical OR immediate data to memory)

Instruction code : 

Byte 1

Data address 

Byte 2

#data 

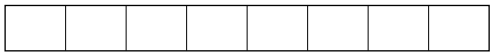
Byte 3

Operation : (data address)←(data address) OR #data

Number of bytes : 3

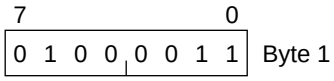
Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

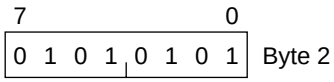
(PSW) 

Description : The logical OR between an 8-bit immediate data value and the specified data address contents is determined. The result is placed in the specified data address.

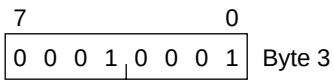
Example ORL 55H, #11H

Instruction code : 

Byte 1

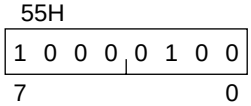


Byte 2

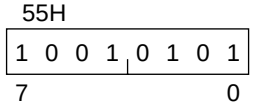


Byte 3

Before execution



After execution



85. ORL data address, A (Logical OR accumulator to memory)

Instruction code : 7 0 — 0 1 0 0 0 0 1 0 — Byte 1

Data address — 7 0 — a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 — Byte 2

Operation : (data address)←(data address) OR (A)

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The logical OR between the accumulator and the specified data address contents is determined. The result is placed in the specified data address.

Example ORL 50H, A

Instruction code :

7 0 — 0 1 0 0 0 0 1 0 — Byte 1

7 0 — 0 1 0 1 0 0 0 0 — Byte 2

Before execution		After execution	
Accumulator		Accumulator	
1 0 0 0 1 1 1 1		1 0 0 0 1 1 1 1	
7 0		7 0	
50H		50H	
0 0 1 0 0 1 0 1		1 0 1 0 1 1 1 1	
7 0		7 0	

86. POP data address (Pop stack to memory)

Instruction code : 7 0

1	1	0	1	0	0	0	0	Byte 1

Data address 7 0

a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2

Operations : (data address)←((SP))
(SP)←(SP)–1

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : Stack contents addressed by the stack pointer are popped in the specified data address, and the stack pointer is decremented by 1.

Example POP PSW:No change to parity bit.

Instruction code : 7 0

1	1	0	1	0	0	0	0	Byte 1

7 0

1	1	0	1	0	0	0	0	Byte 2

	Before execution	After execution
Accumulator	1 0 1 0 1 1 0 0	1 0 1 0 1 1 0 0
PSW (0D0H)	1 0 1 0 1 1 0 0	1 1 1 1 0 0 1 0
Stack pointer	0 0 0 1 0 0 0 0	0 0 0 0 1 1 1 1
10H	1 1 1 1 0 0 1 1	1 1 1 1 0 0 1 1

(7 0 for each)

87. PUSH data address (Push memory onto stack)

Instruction code : Byte 1

7							0
1	1	0	0	0	0	0	0

Data address Byte 2

7							0
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Operations : (SP)←(SP)+1
((SP))←(data address)

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The stack pointer is incremented by 1, and the specified data address contents are pushed in the stack addressed by the stack pointer.

Example PUSH P1

Instruction code :

7							0
1	1	0	0	0	0	0	0

Byte 1

7							0
1	0	0	1	0	0	0	0

Byte 2

Before execution							
Port 1(90H)							
1	1	0	1	0	1	0	1
7							0
Stack pointer							
0	0	0	1	0	0	0	0
7							0
11H (Stack)							
0	0	0	0	0	0	0	0
7							0

After execution							
Port 1(90H)							
1	1	0	1	0	1	0	1
7							0
Stack pointer							
0	0	0	1	0	0	0	1
7							0
11H (Stack)							
1	1	0	1	0	1	0	1
7							0

88. RET (Return from subroutine, non interrupt)

		7							0	
Instruction code	:	0	0	1	0	0	0	1	0	Byte 1

Operations : $(PC_{8\sim15}) \leftarrow ((SP))$
$(SP) \leftarrow (SP)-1$
$(PC_{0\sim7}) \leftarrow ((SP))$
$(SP) \leftarrow (SP)-1$

Number of bytes : 1

Number of cycles : 2

Flags	:	C	AC	F0	RS1	RS0	OV	F1	P
(PSW)									

Description : The stack contents addressed by the stack pointer are popped in the upper order 8 thru 15 of the program counter, and the stack pointer is decremented by 1. Then the stack contents addressed by the updated stack pointer are popped in the lower order 0 thru 7 of the program counter, again decrementing the stack pointer by 1. The program counter is updated with the stack contents, and control is shifted to the address after updating.

89. RETI (Return from interrupt routine)

Instruction code :

7							0	
0	0	1	1	0	0	1	0	Byte 1

Operations : ($PC_{8\sim15}$)←((SP))
(SP)←(SP)–1
($PC_{0\sim7}$)←((SP))
(SP)←(SP)–1
*INTERRUPT ENABLE

Number of bytes : 1

Number of cycles : 2

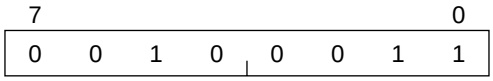
Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

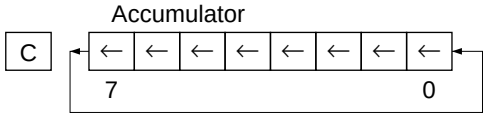
Description : This return instruction functions as an interrupt routine terminating instruction. If a priority interrupt is generated while a non priority interrupt routine is being executed, the CPU commences to process the priority interrupt. And once processing of this interrupt is commenced, no other interrupts can be processed until the RETI instruction is executed.

Stack contents addressed by the stack pointer are popped in the upper order 8 thru 15 of the program counter, and the stack pointer is decremented by 1. Then the stack contents addressed by the updated stack pointer are popped in the lower order 0 thru 7 of the program counter, again decrementing the stack pointer by 1. The program counter is updated with the stack contents, and control is shifted to the address after updating. If a new interrupt is generated, the CPU commences to process the interrupt.

90. RL A (Rotate accumulator left)

Instruction code : 

Operation :



Number of bytes : 1

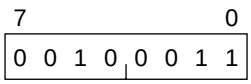
Number of cycles : 1

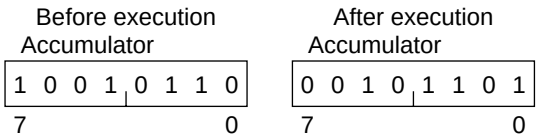
Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

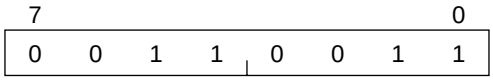
Description : All accumulator bits are shifted by one bit to the left. The MSB (bit 7) is shifted to the LSB bit position (bit 0).

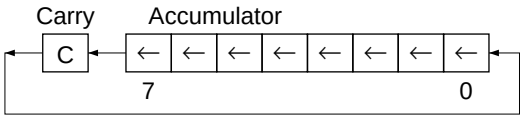
Example RL A

Instruction code : 



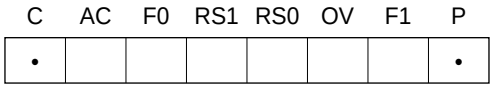
91. RLC A (Rotate accumulator and carry flag left)

Instruction code : 

Operation : 

Number of bytes : 1

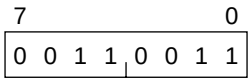
Number of cycles : 1

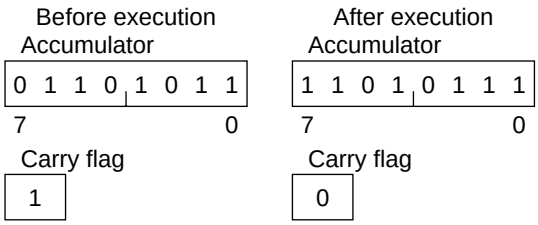
Flags : 

(PSW)

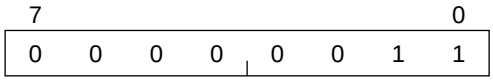
Description : The accumulator and the carry flag are connected, and all bits are shifted by one bit to the left. The carry flag is shifted to the accumulator LSB (bit 0), and the accumulator MSB (bit 7) is shifted to the carry flag.

Example RLC A

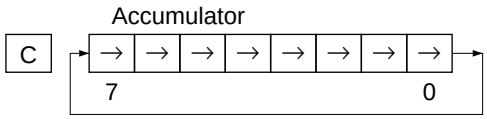
Instruction code : 



92. RR A (Rotate accumulator right)

Instruction code : 

Operation :

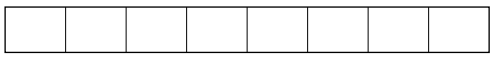


Number of bytes : 1

Number of cycles : 1

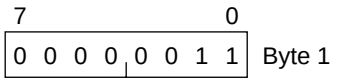
Flags : C AC F0 RS1 RS0 OV F1 P

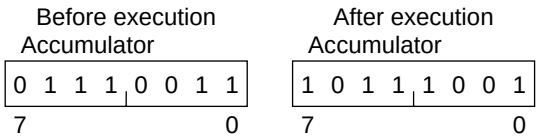
(PSW)



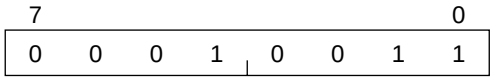
Description : All accumulator bits are shifted by one bit to the right. The LSB (bit 0) is shifted to the MSB bit position (bit 7).

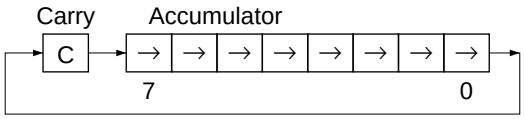
Example RR A

Instruction code : 



93. RRC A (Rotate accumulator and carry flag right)

Instruction code : 

Operation : 

Number of bytes : 1

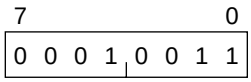
Number of cycles : 1

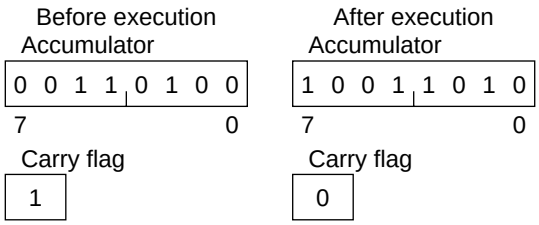
Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • • •

Description : The accumulator and the carry flag are connected, and all bits are shifted by one bit to the right. The carry flag is shifted to the accumulator MSB (bit 7), and the accumulator LSB (bit 0) is shifted to the carry flag.

Example RRC A

Instruction code : 



94. SETB C (Set carry flag)

Instruction code : 7 0

1	1	0	1	0	0	1	1

Byte 1

Operation : (C)←1

Number of bytes : 1

Number of cycles : 1

Flags :

C	AC	F0	RS1	RS0	OV	F1	P
•							

(PSW)

Description : The carry flag is cleared to 1.

Example SETB C

Instruction code : 7 0

1	1	0	1	0	0	1	1

Byte 1

Before execution	After execution
Carry flag	Carry flag
0	1

95. SETB bit address (Set bit)

Instruction code : 7 0 — 1 1 0 1 0 0 1 0 — Byte 1

Bit address — b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 — Byte 2

Operation : (bit address)←1

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The specified bit address content is set to 1.

Example SETB IE.7

Instruction code :

Byte 1: 1 1 0 1 0 0 1 0

Byte 2: 1 0 1 0 1 1 1 1

Before execution IE (0A8H)	After execution IE (0A8H)
0 1 1 0 0 1 1 1	1 1 1 0 0 1 1 1
7 0	7 0

96. SJMP code address (Short jump)

Instruction code : 1 0 0 0 0 0 0 0 Byte 1

Relative offset : R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0 Byte 2

Operations : (PC)←(PC)+2
(PC)←(PC)+relative offset

Number of bytes : 2

Number of cycles : 2

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : Relative offset jump data is added/subtracted to/from the program counter contents following an increment. The program counter contents are updated, and control is then shifted to the updated address. The range in which relative jumps can be executed by this instruction is +127 to –128 in respect to the incremented program counter contents. There is no page field restrictions.

Example SJMP CHECK

LOC	OBJ	SOURCE
0111	8010	SJUMP:SJMP CHECK
0123	33	CHECK:RLC A

Instruction code :

7							0	
1	0	0	0	0	0	0	0	Byte 1

7							0	
0	0	0	1	0	0	0	0	Byte 2

Before execution

Program counter

15							8	7							0
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1

After execution

Program counter

15							8	7							0
0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1

97. SUBB A, #data (Substract immediate data from accumulator with borrow)

Instruction code : Byte 1

7							0
1	0	0	1	0	1	0	0

#data Byte 2

7							0
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Operation : (A)←(A)–((C)+#data)

Number of bytes : 2

Number of cycles : 1

Flags (PSW) :

C	AC	F0	RS1	RS0	OV	F1	P
•	•				•		•

Description : The carry flag content and an immediate data value are substracted from the accumulator contents. The result is placed in the accumulator, and the flags are updated.

Example SUBB A, #05H

Instruction code : Byte 1

7							0
1	0	0	1	0	1	0	0

Byte 2

7							0
0	0	0	0	0	1	0	1

	Before execution	After execution
Carry flag	1	0
Auxiliary carry flag	0	0
Overflow flag	1	0
Accumulator	1 0 1 0 1 0 0 1	1 0 1 0 0 0 1 1

98. SUBB A, @Rr (Substract indirect address from accumulator with borrow)

Instruction code : 7 0

1	0	0	1	0	1	1	r

Byte 1

Operation : (A)←(A)–((C)+((Rr))) r=0 or 1

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

•	•				•		•

Description : The carry flag content and the data memory location contents addressed by the register r contents are substracted from the accumulator contents. The result is placed in the accumulator, and the flags are updated.

Example SUBB A, @R0

Instruction code : 7 0

1	0	0	1	0	1	1	0

Byte 1

	Before execution	After execution
Carry flag	0	1
Auxiliary carry flag	0	0
Overflow flag	0	1
Register 0	0 1 0 0 0 1 1 1 (7 0)	0 1 0 0 0 1 1 1 (7 0)
47H	1 1 0 1 0 0 1 0 (7 0)	1 1 0 1 0 0 1 0 (7 0)
Accumulator	0 1 0 1 0 1 0 0 (7 0)	1 0 0 0 0 0 1 0 (7 0)

99. SUBB A, Rr (Substract register from accumulator with borrow)

Instruction code : 7 0

1	0	0	1	1	r2	r1	r0

Byte 1

Operation : (A)←(A)–((C)+(Rr))

Number of bytes : 1

Number of cycles : 1

Flags :

C	AC	F0	RS1	RS0	OV	F1	P
(PSW)	•	•				•		•

Description : The carry flag content and the register r contents are substracted from the accumulator contents. The result is placed in the accumulator, and the flags are updated.

Example SUBB A, R7

Instruction code : 7 0

1	0	0	1	1	1	1	1

Byte 1

Before execution	After execution
Carry flag: 1	Carry flag: 0
Auxiliary carry flag: 0	Auxiliary carry flag: 1
Overflow flag: 0	Overflow flag: 1
Register 7: 0 1 0 1 1 0 0 0 (7 … 0)	Register 7: 0 1 0 1 1 0 0 0 (7 … 0)
Accumulator: 1 0 0 0 0 1 0 0 (7 … 0)	Accumulator: 0 0 1 0 1 0 1 1 (7 … 0)

100. SUBB A, data address (Substract memory from accumulator with borrow)

Instruction code : 7 0 — 1 0 0 1 0 1 0 0 — Byte 1

Data address — 7 0 — a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 — Byte 2

Operation : (A)←(A)–((C)+(data address))

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • • • • •

Description : The carry flag contents and the specified data address contents are substracted from the accumulator contents. The result is placed in the accumulator, and the flags are updated.

Example SUBB A, DPH

Instruction code : 7 0 — 1 0 0 1 0 1 0 1 — Byte 1

7 0 — 1 0 0 0 0 0 1 1 — Byte 2

	Before execution	After execution
Carry flag	0	1
Auxiliary carry flag	0	1
Overflow flag	0	0
DPH	1 0 1 0 1 0 1 0	1 0 1 0 1 0 1 0
Accumulator	0 1 0 1 0 1 0 1	1 0 1 0 1 0 1 1

101. SWAP A (Exchange nibble in accumulator)

Instruction code : 7 0

1	1	0	0	0	1	0	0

Byte 1

Operation : $(A_{4\sim7}) \rightleftarrows (A_{0\sim3})$

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

Description : The contents of the four higher order bits (4 thru 7) of the accumulator are exchanged with the contents of the four lower order bits (0 thru 3)

Example SWAP A

Instruction code : 7 0

1	1	0	0	0	1	0	0

Byte 1

Before execution

Accumulator

1	0	1	1	0	1	0	0

7 0

After execution

Accumulator

0	1	0	0	1	0	1	1

7 0

102. XCH A, @Rr (Exchange indirect address with accumulator)

Instruction code : Byte 1

7							0
1	1	0	0	0	1	1	r

Operation : (A)⇄((Rr)) r=0 or 1

Number of bytes : 1

Number of cycles : 1

Flags : (PSW)

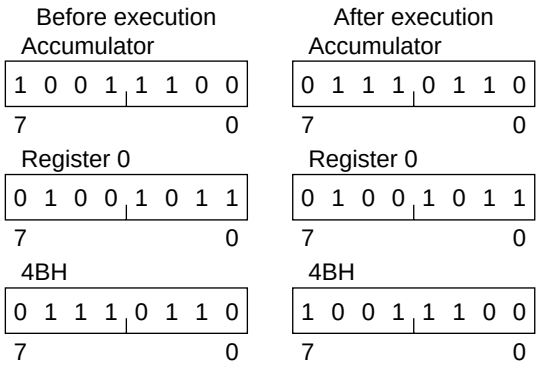
C	AC	F0	RS1	RS0	OV	F1	P
							•

Description : The accumulator contents are exchanged with the data memory location contents addressed by the register r, and the flag is updated.

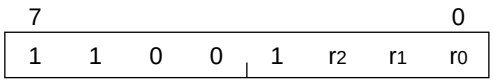
Example XCH A, @R0

Instruction code : Byte 1

7							0
1	1	0	0	0	1	1	0



103. XCH A, Rr (Exchange register with accumulator)

Instruction code : 

Operation : (A)⇄(Rr) r=0 thru 7

Number of bytes : 1

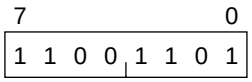
Number of cycles : 1

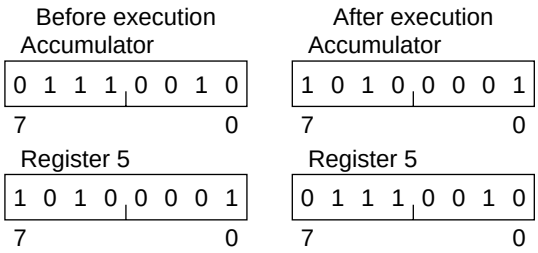
Flags : (PSW)

C	AC	F0	RS1	RS0	OV	F1	P
							•

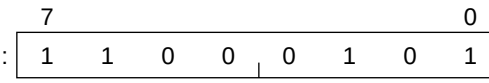
Description : The accumulator contents are exchanged with the register r contents, and the flag is updated.

Example XCH A, R5

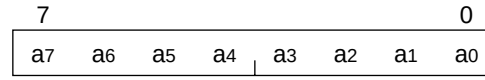
Instruction code : 



104. XCH A, data address (Exchange memory with accumulator)

Instruction code : 

Byte 1

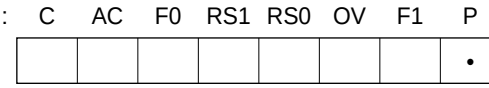
Data address 

Byte 2

Operation : (A)⇄(data address)

Number of bytes : 2

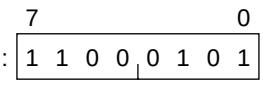
Number of cycles : 1

Flags : 

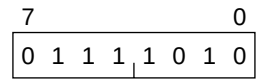
(PSW)

Description : The accumulator contents are exchanged with the specified data address contents, and the flag is updated.

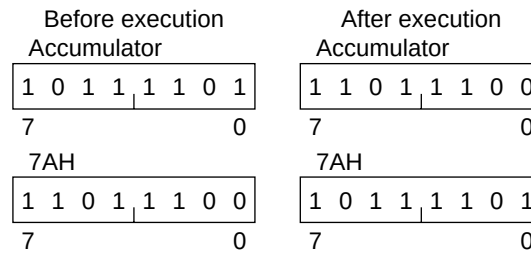
Example XCH A, 7AH

Instruction code : 

Byte 1



Byte 2



105. XCHD A, @Rr (Exchange low nibbles of indirect address with accumulator)

Instruction code : 7 0

1	1	0	1	0	1	1	r

Byte 1

Operation : $(A_{0\sim3}) \rightleftarrows ((Rr_{0\sim3}))$ r=0 or 1

Number of bytes : 1

Number of cycles : 1

Flags :

C	AC	F0	RS1	RS0	OV	F1	P
							•

(PSW)

Description : The lower order bits (0 thru 3) of the accumulator contents are exchanged with contents of the lower order bits (0 thru 3) of the data memory location addressed by the register r contents. The flag is updated.

Example XCHD A, @R0

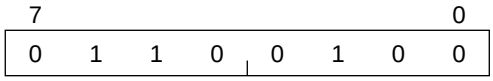
Instruction code : 7 0

1	1	0	1	0	1	1	0

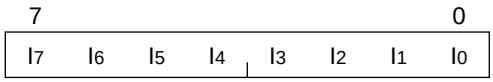
Byte 1

Before execution								After execution							
Accumulator								Accumulator							
1	1	1	1	0	1	1	0	1	1	1	1	1	1	0	1
7							0	7							0
Register 0								Register 0							
0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0
7							0	7							0
60H								60H							
0	0	0	0	1	1	0	1	0	0	0	0	0	1	1	0
7							0	7							0

106. XRL A, #data (Logical exclusive OR immediate data to accumulator)

Instruction code : 

Byte 1

#data 

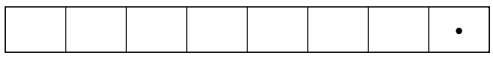
Byte 2

Operation : (A)←(A) XOR #data

Number of bytes : 2

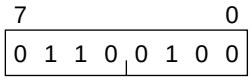
Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

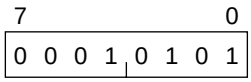
(PSW) 

Description : The exclusive OR operation is executed between an immediate data value and the accumulator contents. The result is placed in the accumulator, and the flag is updated.

Example XRL A, #15H

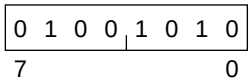
Instruction code : 

Byte 1

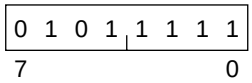


Byte 2

Before execution
Accumulator



After execution
Accumulator



107. XRL A, @Rr (Logical exclusive OR indirect address to accumulator)

Instruction code : 7 0

0	1	1	0	0	1	1	r

Byte 1

Operation : (A)←(A) XOR ((Rr)) r=0 or 1

Number of bytes : 1

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW)

							•

Description : The exclusive OR operation is executed between the accumulator contents and the data memory location contents addressed by the register r contents. The result is placed in the accumulator, and the flag is updated.

Example XRL A, @R1

Instruction code : 7 0

0	1	1	0	0	1	1	1

Byte 1

Before execution

Accumulator

0	1	0	0	1	0	0	1

7 0

Register 1

0	0	1	1	0	1	1	0

7 0

36H

1	0	0	1	0	0	0	1

7 0

After execution

Accumulator

1	1	0	1	1	0	0	0

7 0

Register 1

0	0	1	1	0	1	1	0

7 0

36H

1	0	0	1	0	0	0	1

7 0

108. XRL A, Rr (Logical exclusive OR register to accumulator)

Instruction code	:	7 0 0 1 1 0 1 r2 r1 r0 Byte 1
Operation	:	(A)←(A) XOR (Rr) r=0 thru 7
Number of bytes	:	1
Number of cycles	:	1
Flags	:	C AC F0 RS1 RS0 OV F1 P
(PSW)		•
Description	:	The exclusive OR between the accumulator contents and the register r contents is determined. The result is stored in the accumulator and the flag is updated.

Example XRL A, R3

Instruction code	:	7 0 0 1 1 0 1 0 1 1 Byte 1

Before execution

Accumulator

1 0 1 0 0 0 1 0

7 0

Register 3

0 0 1 0 0 1 0 1

7 0

After execution

Accumulator

1 0 0 0 0 1 1 1

7 0

Register 3

0 0 1 0 0 1 0 1

7 0

109. XRL A, data address (Logical exclusive OR memory to accumulator)

Instruction code : 7 0 — 0 1 1 0 0 1 0 1 — Byte 1

Data address — 7 0 — a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 — Byte 2

Operation : (A)←(A) XOR (data address)

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • (P)

Description : The exclusive OR between the accumulator contents and the specified data address contents is determined. The result is placed in the accumulator and the flag is updated.

Example XRL A, 70H

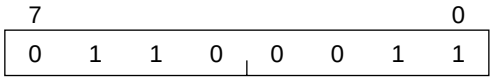
Instruction code :

7 0 — 0 1 1 0 0 1 0 1 — Byte 1

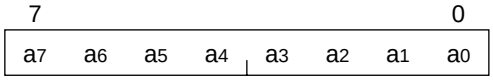
7 0 — 0 1 1 1 0 0 0 0 — Byte 2

Before execution		After execution	
Accumulator		Accumulator	
1 1 1 1 0 1 1 0		1 1 0 1 1 0 0 1	
7 0		7 0	
70H		70H	
0 0 1 0 1 1 1 1		0 0 1 0 1 1 1 1	
7 0		7 0	

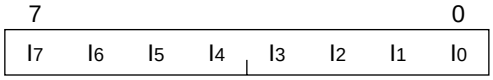
110. XRL data address, #data (Logical exclusive OR immediate data to memory)

Instruction code : 

Byte 1

Data address 

Byte 2

#data 

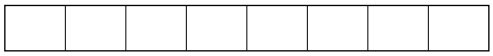
Byte 3

Operation : (data address)←(data address) XOR #data

Number of bytes : 3

Number of cycles : 2

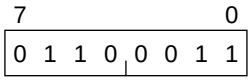
Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) 

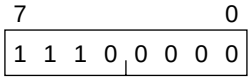
Description : The exclusive OR between an immediate data value and the specified data address contents is determined. The result is placed in the specified data address.

Example XRL ACC, #5AH

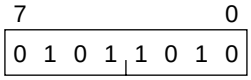
Instruction code :



Byte 1

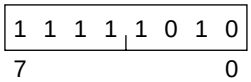


Byte 2

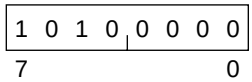


Byte 3

Before execution
Accumulator



After execution
Accumulator



111. XRL data address, A (Logical exclusive OR accumulator to memory)

Instruction code : 7 0 — 0 1 1 0 0 0 1 0 — Byte 1

Data address — 7 0 — a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 — Byte 2

Operation : (data address)←(data address) XOR (A)

Number of bytes : 2

Number of cycles : 1

Flags : C AC F0 RS1 RS0 OV F1 P

(PSW) : • (P)

Description : The exclusive OR between the accumulator and the specified data address contents is determined. The result is placed in the specified data address.

Example XRL 20H, A

Instruction code :

7 0 — 0 1 1 0 0 0 1 0 — Byte 1

7 0 — 0 0 1 0 0 0 0 0 — Byte 2

Before execution		After execution	
Accumulator		Accumulator	
0 1 1 1 0 1 0 1		0 1 1 1 0 1 0 1	
7 0		7 0	
20H		20H	
1 1 0 1 0 0 1 1		1 0 1 0 0 1 1 0	
7 0		7 0	