

OKI semiconductor

MSM66207/66P207

OKI'S ORIGINAL nX HIGH PERFORMANCE CMOS SINGLE CHIP 8/16 BIT MICROCONTROLLER

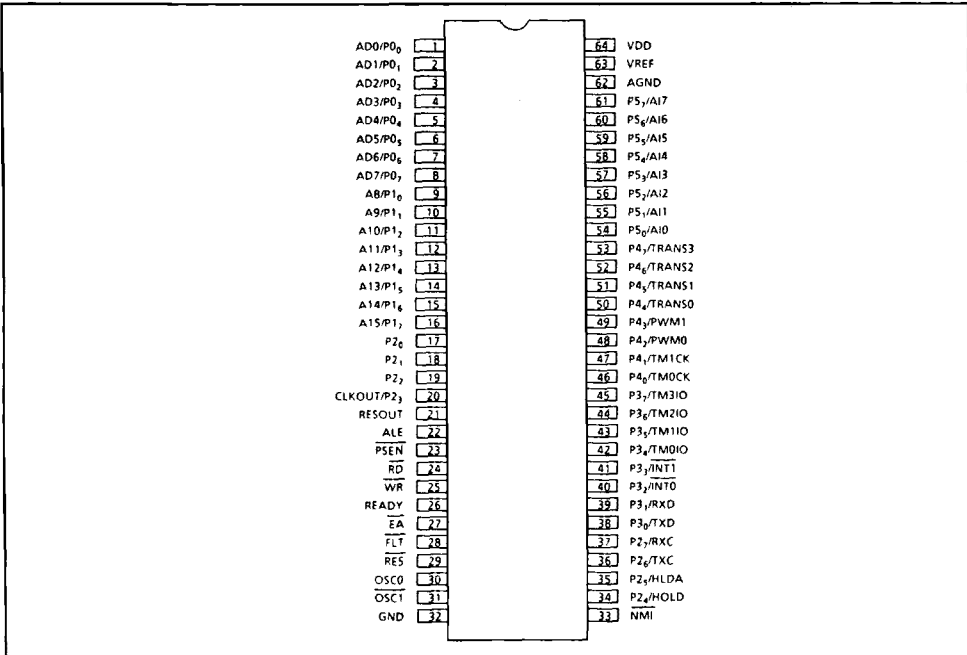
GENERAL DESCRIPTION

The OKI MSM66207 is a new generation, high performance single chip microcontroller implemented in silicon gate complementary metal oxide semiconductor technology (CMOS). Integrated within this chip are a 16-bit ALU, 32K bytes of mask program ROM, 1K bytes of data RAM, 48 I/O lines, built-in 16-bit timers, 10-bit A/D converter, serial I/O port, pulse width modulator (PWM), and an oscillator. Also available is the MSM66P207, which replaces the on-chip program memory with one-time PROM.

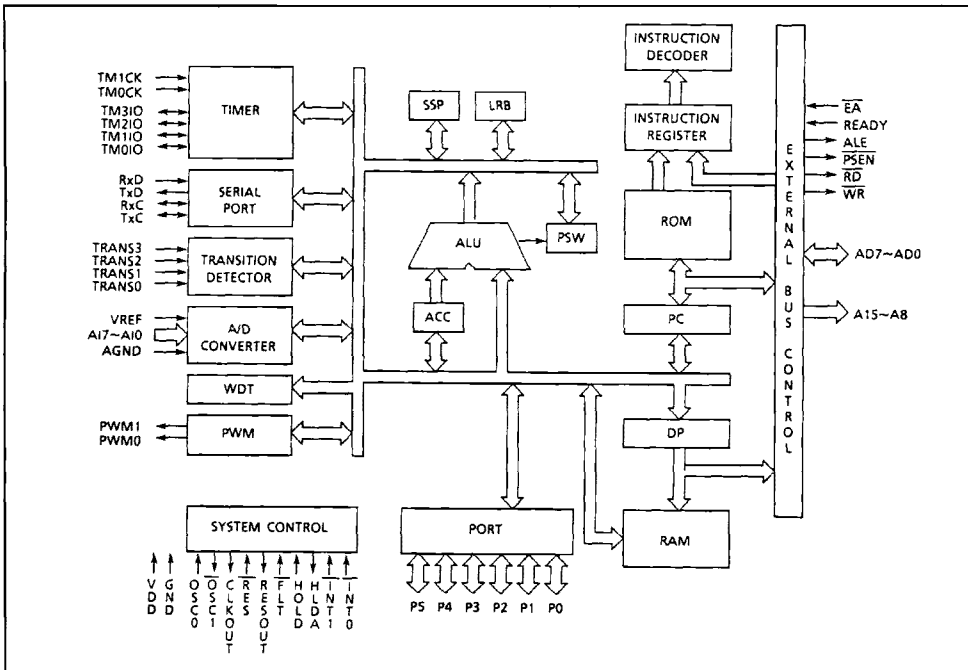
FEATURES

- 8-Bit External Data Bus Interface
- 16-Bit Internal Architecture
- 64K address space for program memory (including 32K bytes on-chip ROM)
- 64K address space for data memory (including 1K bytes on-chip RAM)
- High speed execution
Minimum Cycle for Instruction: 400ns (10MHz)
- Powerful Instruction set
8/16 data transfer operation
8/16 bit arithmetic operation
16(8) bit x 16(8) bit → 32(16) bit
32(16) bit ÷ 16(8) bit → 32(16) bit
16(8) bit ± 16(8) bit → 16(8) bit
8/16 logic operation
Bit operation
ROM table access operation
- Numerous Addressing Modes
- 8 Input only lines: 40 Input/Output lines
- Built-in 16 bit timer x 4
Each timer has the following 4 modes.
Auto reload timer mode
Clock output mode
Capture register mode
Real time output mode
- Serial Port x 1 channel (variable bit length, baud rate generators for transmitter & receiver)
Asynchronous normal mode
Asynchronous multi processor communication mode
Synchronous normal mode
Synchronous multi processor communication mode
- 16 bit Pulse Width Modulator x 2
- Transition Detector x 4
- 10 bit A/D converter (8 channel)
- 1 non-maskable interrupt, 16 maskable interrupts
- Stand-by Function
Software Clock stop
Software CPU stop
Hardware CPU stop
- Packages:
64 pin plastic shrink DIP (SDIP64-P-750)
68 pin PLCC (QFJ68-P-5950)

PIN CONFIGURATION



FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTION

Designation	Input/Output	Function
P0 ₀ – P0 ₇ / AD0 – AD7	I/O	<p>P0: 8-bit I/O port. Each bit can be assigned to input or output.</p> <p>AD: Outputs the lower 8 bits of program counter during external program memory fetch, and receives the addressed instruction under the control of PSEN. Also outputs the address, Outputs or inputs data during an external data memory access instruction, under the control of ALE, RD, and WR.</p>
P1 ₀ – P1 ₇ / A8 – A15	I/O	<p>P1: 8-bit I/O port. Each bit can be assigned to input or output.</p> <p>A: Outputs the upper 8 bits of program counter (PC₈₋₁₅) during external program memory fetch. Also outputs the upper 8 bits of address during an external data memory access instructions.</p>
P2 ₀ – P2 ₂ P2 ₃ /CLKOUT P2 ₄ /HOLD P2 ₅ /HLDA P2 ₆ /TxC P2 ₇ /RxC	I/O	<p>P2: 8-bit I/O port. Each bit can be assigned to input or output.</p> <p>CLKOUT: Clock output pin. Output frequency range is equal to or twice the system clock.</p> <p>HOLD: Input pin to request the CPU to enter the hardware power-down state.</p> <p>HLDA: HOLD ACKNOWLEDGE: the HLDA signal appears in response to the HOLD signal and indicates that the CPU has entered the power-down state.</p> <p>TxC: Transmitter clock input/output pin.</p> <p>RxC: Receiver clock input/output pin.</p>
P3 ₀ /TxD P3 ₁ /RxD P3 ₂ /INT0 P3 ₃ /INT1 P3 ₄ /TM0IO P3 ₅ /TM1IO P3 ₆ /TM2IO P3 ₇ /TM3IO	I/O	<p>P3: 8-bit I/O port. Each bit can be assigned to input or output.</p> <p>TxD: Transmitter data output pin.</p> <p>RxD: Receiver data input pin.</p> <p>INT: Interrupt Request Input pin. Falling edge trigger or level trigger is selectable.</p> <p>TM0IO~TM3IO: One of the following signals is output or input.</p> <ul style="list-style-type: none"> ● clock twice the frequency range of the 16 bit timer overflow ● load trigger signal to the capture register input ● setting value output <p>Whether the signal is input or output depends on the mode.</p>

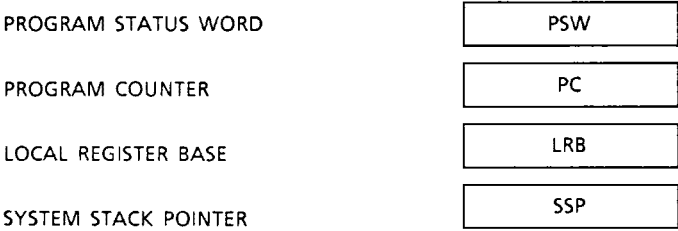
PIN DESCRIPTION (Continued)

Designation	Input/Output	Function
P4 ₀ /TM0CK	I/O	P4: 8-bit I/O port. Each bit can be assigned to input or output
P4 ₁ /TM1CK		TM0CK, TM1CK: Clock input pins of timer 0, timer 1.
P4 ₂ /PWM0		TRANS: Transition Detector. The input pins which sense the falling edge and set the flag.
P4 ₃ /PWM1 P4 ₄ – P4 ₇ / TRANS0 – TRANS3		PWM: Pulse Wide Modulator output pin.
P5 ₀ – P5 ₇ / AI0 – AI7	INPUT	P5: 8-bit input port. AI: Analog signal input pin for A/D converter.
RESOUT	OUTPUT	Output 'H' level when the CPU is in RESET cycle. Reset to 'L' level by program.
ALE	OUTPUT	Address Latch Enable: The timing pulse to latch the lower 8 bit of the address output from port 0 when the CPU accesses the external memory.
$\overline{\text{PSEN}}$	OUTPUT	Program Store Enable: The strobe pulse to fetch to external program memory.
$\overline{\text{RD}}$	OUTPUT	Output strobe activated during a bus read cycle. Used to enable data on to the bus from the external data memory.
$\overline{\text{WR}}$	OUTPUT	Output strobe during a bus write cycle. Used as write strobe to external data memory.
READY	INPUT	Used when the CPU accesses low speed peripherals.
$\overline{\text{EA}}$	INPUT	Normally set to 'H' level. If set to 'L' level, the CPU fetches the code from external program memory.
$\overline{\text{FLT}}$	INPUT	If $\overline{\text{FLT}}$ is 'H' level, ALE, $\overline{\text{WR}}$, $\overline{\text{RD}}$, $\overline{\text{PSEN}}$ are set 'H' level when reset. If $\overline{\text{FLT}}$ is set to 'L', ALE, $\overline{\text{WR}}$, $\overline{\text{RD}}$, $\overline{\text{PSEN}}$ are set to floating level when reset.
$\overline{\text{RES}}$	INPUT	RESET input pin.
OSC ₀ , $\overline{\text{OSC}}_1$		Oscillation circuit input and output.
$\overline{\text{NMI}}$	INPUT	Non maskable interrupt input pin (falling edge)
VREF	INPUT	Reference voltage input pin for A/D converter
AGND	INPUT	Ground for A/D converter
VDD		System power supply
GND		Ground

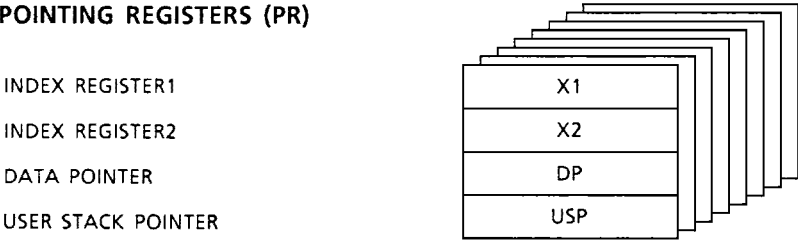
REGISTERS



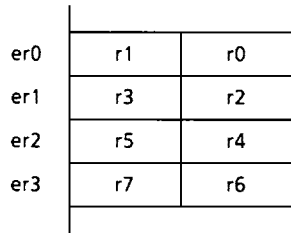
- **CONTROL REGISTERS (CR)**



- **POINTING REGISTERS (PR)**



- **LOCAL REGISTERS**



- **SPECIAL FUNCTION REGISTERS (SFR)**

All of the I/O functions are controlled by SFRs. Also, some of the internal functions (Timer, WDT, etc, ...) are controlled by SFRs. SFRs are located in the top of RAM space (000H~007FH).

MSM66207 SPECIAL FUNCTION REGISTERS

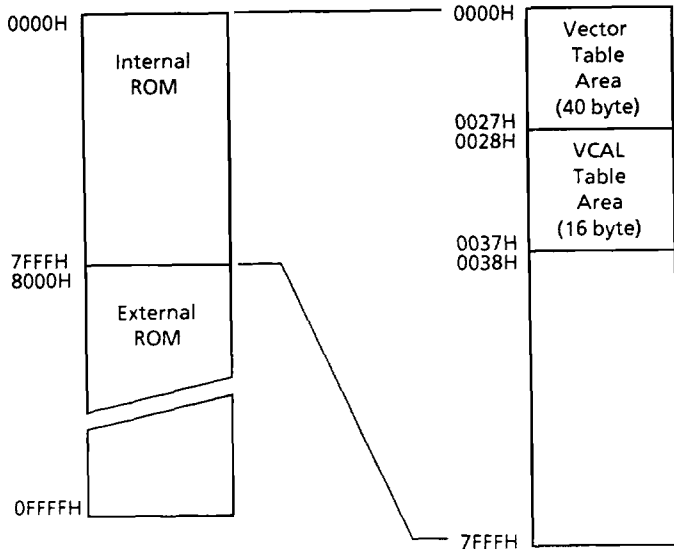
Address (HEX)	SFR Name	Abbreviated Name	R/W	8/16-bit operation	Reset Value (HEX)				
0000	System Stack Printer	SSP (ASSP)	R/W	8/16	FFH				
0001					FFH				
0002	Local Register Base	LRB (ALRB)			R/W	8/16	undefined		
0003							undefined		
0004	Program Status Word	PSWL (APSW)					R/W	8/16	C8H
0005		PSWH							0CH
0006		Accumulator							ACC
0007	00H								
0010	Standby control register	SBYCON	W	8					F8H or F0H
0011	Watchdog timer	WDT			00H/WDT is stopped				
0012	Peripheral control register	PRPHF	R/W	8	FDH				
0013	Stop code acceptor	STPACP	W		"0"				
0018	Interrupt request flag	IRQ	R/W	8/16	00H				
0019					00H				
001A	Interrupt Enable flag	IE			R/W	8/16	00H		
001B							00H		
001C	External interrupt control register	EXICON					R/W	8	FCH
0020	Port 0 data register	P0							undefined
0021	Port 0 mode register	P0IO							00H
0022	Port 1 data register	P1	undefined						
0023	Port 1 mode register	P1IO	00H						
0024	Port 2 data register	P2	undefined						
0025	Port 2 mode register	P2IO	00H						
0026	Port 2 secondary function control register	P2SF	R/W	8	07H				
0028	Port 3 data register	P3			undefined				
0029	Port 3 mode register	P3IO			00H				
002A	Port 3 secondary function control register	P3SF			00H				
002C	Port 4 data register	P4			undefined				
002D	Port 4 mode register	P4IO			00H				
002E	Port 4 secondary function control register	P4SF			00H				

Address (HEX)	SFR Name	Abbreviated Name	R/W	8/16-bit operation	Reset Value (HEX)	
002F	Port 5	P5	R	8	–	
0030	Timer 0 counter	TM0	R/W	16	00H	
0031					00H	
0032	Timer 0 register	TMR0			00H	
0033					00H	
0034	Timer 1 counter	TM1			00H	
0035					00H	
0036	Timer 1 register	TMR1			00H	
0037					00H	
0038	Timer 2 counter	TM2			00H	
0039					00H	
003A	Timer 2 register	TMR2			00H	
003B					00H	
003C	Timer 3 counter	TM3			00H	
003D					00H	
003E	Timer 3 register	TMR3			00H	
003F					00H	
0040	Timer 0 Control register	TC0N0	R/W	8	00H	
0041	Timer 1 Control register	TC0N1			00H	
0042	Timer 2 Control register	TC0N2			00H	
0043	Timer 3 Control register	TC0N3			00H	
0046	Transition detector register	TRNSIT			undefined	
0048	Serial port transmission baud rate generator counter	STTM			00H	
0049	Serial port transmission baud rate generator register	STTMR			00H	
004A	Serial port transmission control register	STTMC			0CH	
004C	Serial port receiving baud rate generator counter	SRTM			00H	
004D	Serial port receiving baud rate generator register	SRTMR			00H	
004E	Serial port receiving control register	SRTMC			0EH	
0050	Serial port transmission mode control register	STCON			80H	
0051	Serial port transmission data buffer register	STBUF			W	undefined
0054	Serial port receiving mode control register	SRCON			R/W	00H
0055	Serial port receiving data buffer register	SRBUF			R	undefined
0056	Serial port receiving error register	SRSTAT			R/W	FOH
0058	A/D scan mode register	ADSCAN	80H			
0059	A/D select mode register	ADSEL	A0H			

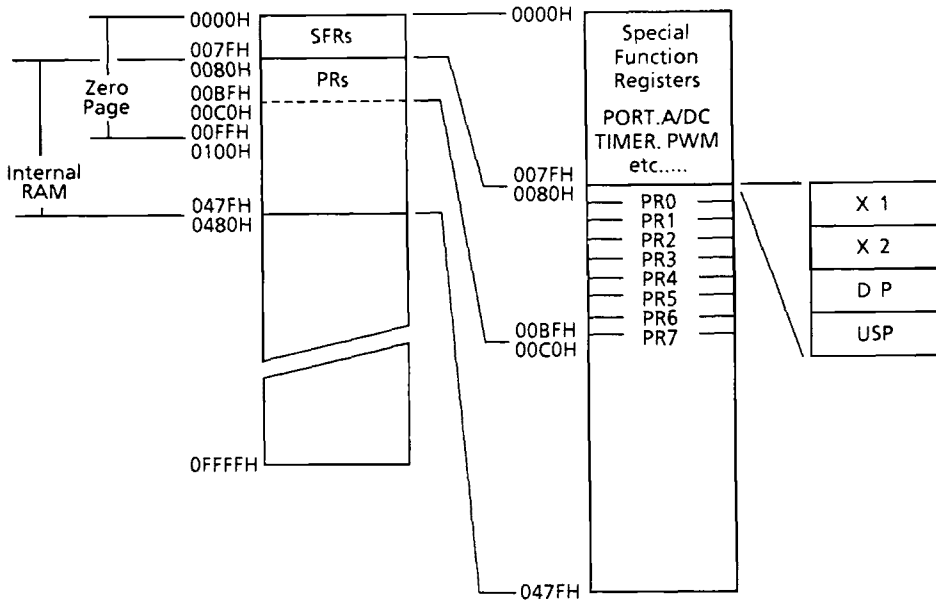
Address (HEX)	SFR Name	Abbreviated Name	R/W	8/16-bit operation	Reset Value (HEX)
0060	A/D conversion result register 0	ADCR0	R	8/16	undefined
0061					
0062	A/D conversion result register 1	ADCR1			
0063					
0064	A/D conversion result register 2	ADCR2			
0065					
0066	A/D conversion result register 3	ADCR3			
0067					
0068	A/D conversion result register 4	ADCR4			
0069					
006A	A/D conversion result register 5	ADCR5			
006B					
006C	A/D conversion result register 6	ADCR6			
006D					
006E	A/D conversion result register 7	ADCR7			
006F					
0070	PWM 0 counter	PWMC0	R/W	8	00H
0071					
0072	PWM 0 register	PWMR0			00H
0073					
0074	PWM 1 counter	PWMC1			00H
0075					
0076	PWM 1 register	PWMR1			00H
0077					
0078	PWM 0 control register	PWC0N0			00H
007A	PWM 1 control register	PWC0N1			00H

MEMORY MAP

● Program Memory Space



● Data Memory Space

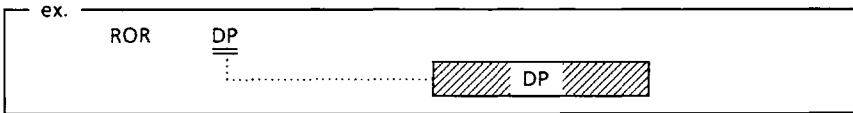


ADDRESSING MODE

The MSM66207 supports 64KB of data space and 64KB of program space with various types of addressing modes. These modes divide into the following types.

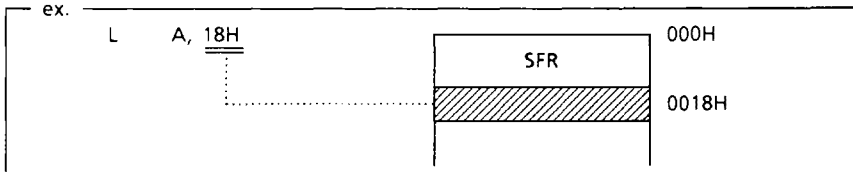
1. RAM ADDRESSING (FOR DATA SPACE)

1.1 Register Direct Addressing

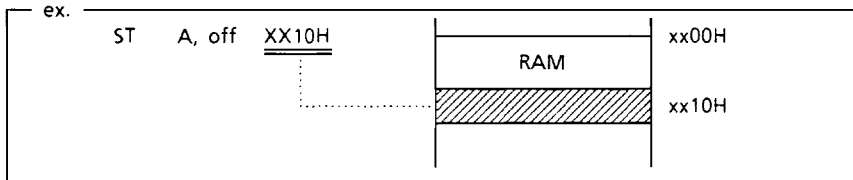


1.2 Displacement Addressing

a) Zero Page

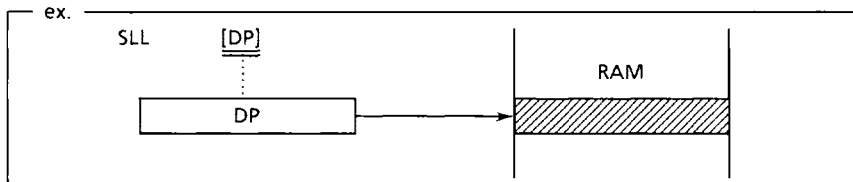


b) Direct Page

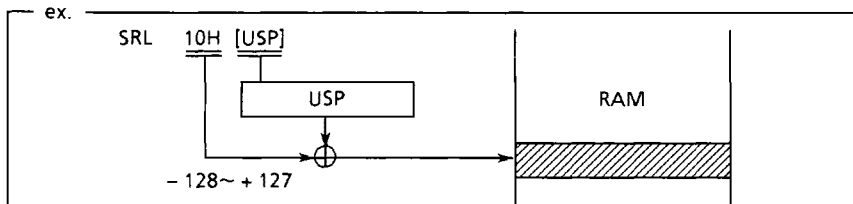


1.3 Pointing Register Indirect Addressing

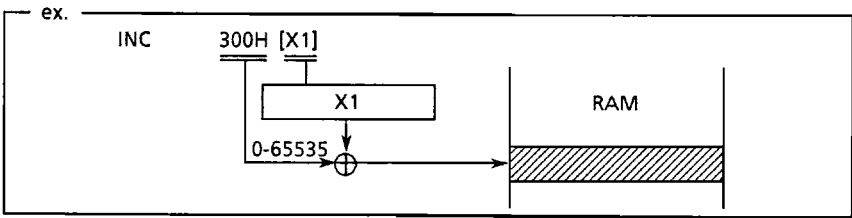
a) Data Pointer (DP) Indirect



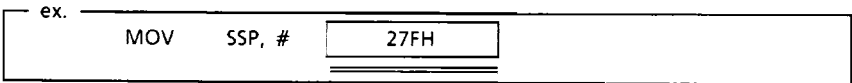
b) User Stack Pointer (USP) Indirect



c) Index register (X1, X2) Indirect

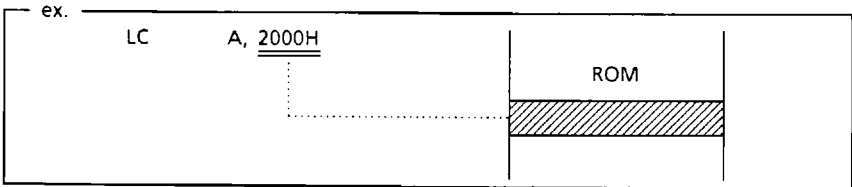


1.4 Immediate Addressing



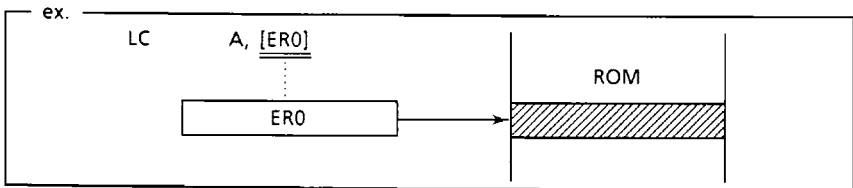
2. ROM ADDRESSING (FOR PROGRAM SPACE)

2.1 Direct Addressing



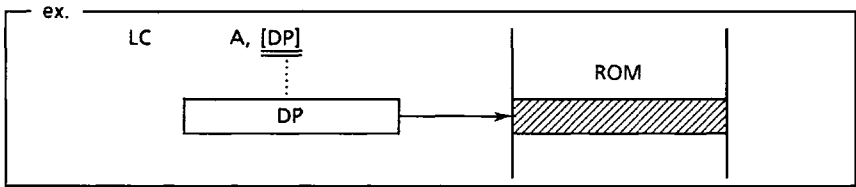
2.2 Simple Indirect Addressing

a) Local Register Indirect

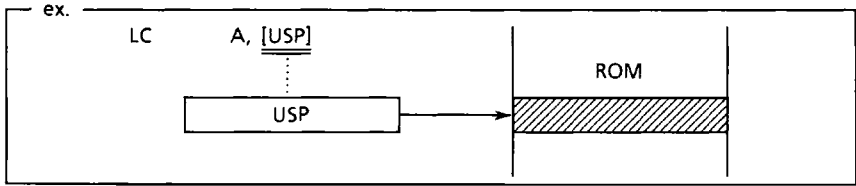


b) Pointing Register Indirect

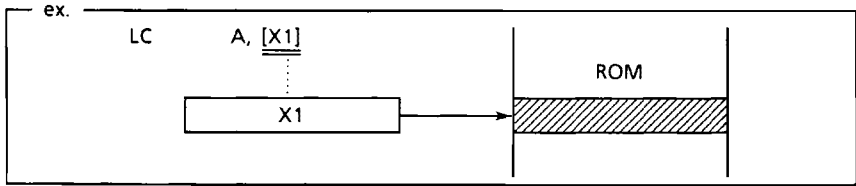
① Data Pointer (DP) Indirect



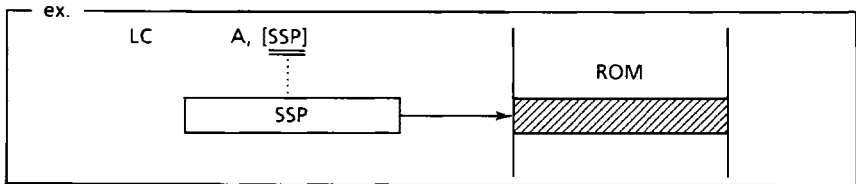
② User Stack Pointer (USP) Indirect



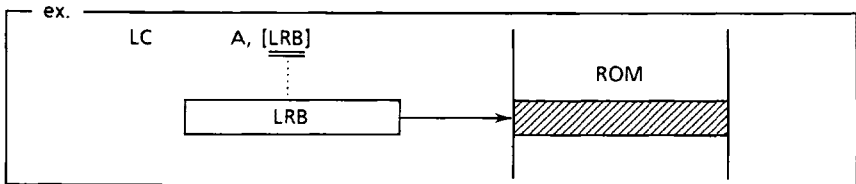
③ Index Register (X1, X2) Indirect



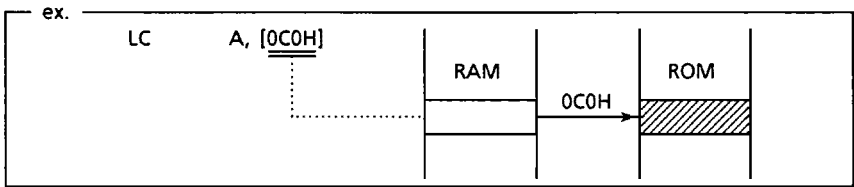
c) System Stack Pointer (SSP) Indirect



d) Local Register Base (LRB) Indirect

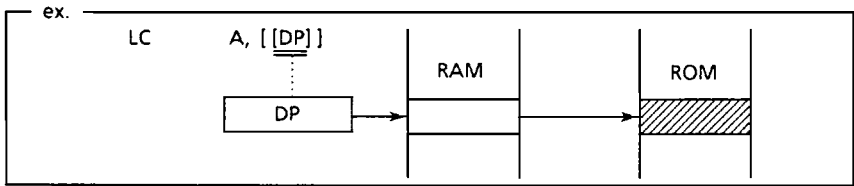


e) RAM Indirect

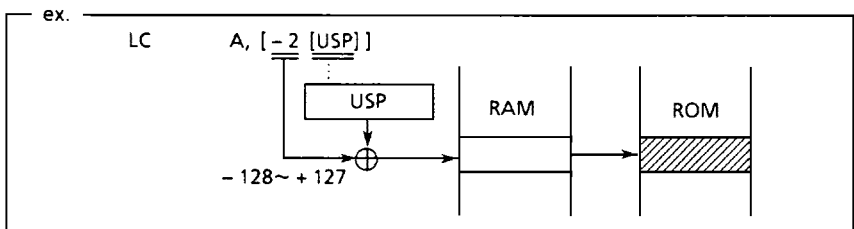


2.3 Double Indirect Addressing

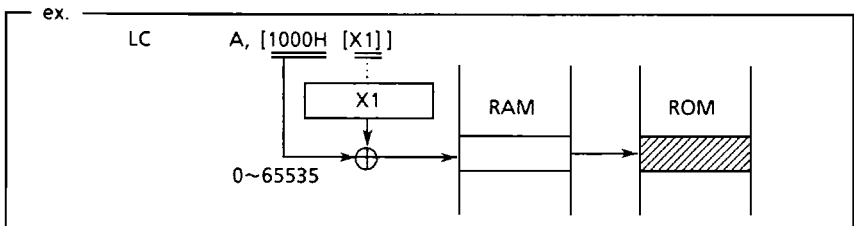
a) Data Pointer (DP) Double Indirect



b) User Stack Pointer (USP) Double Indirect



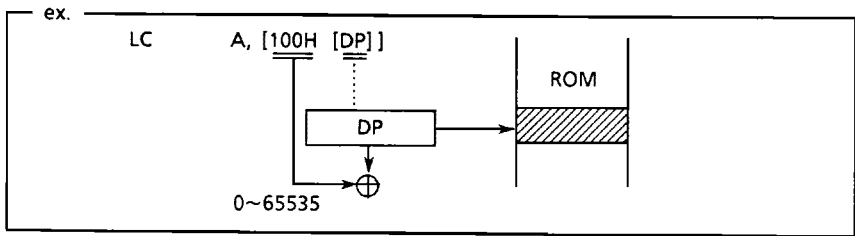
c) Index Register (X1, X2) Double Indirect



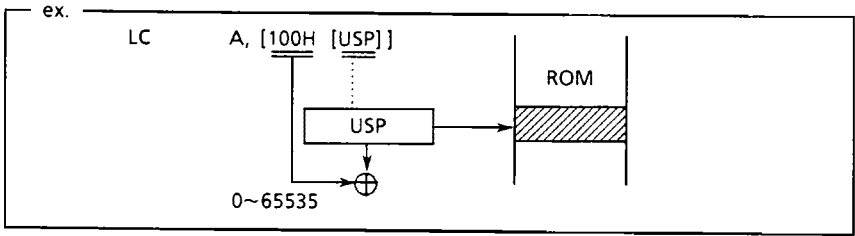
2.4 Indirect Addressing with 16 bit offset

a) Pointing Register Indirect

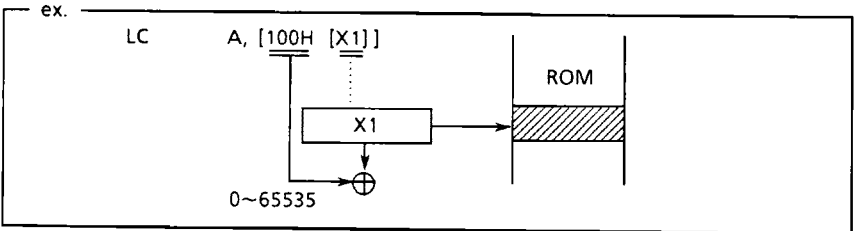
① Data Pointer (DP) Indirect



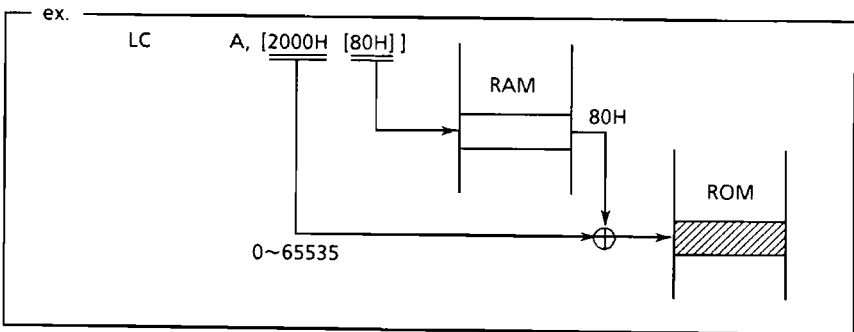
② User Stack Pointer (USP) Indirect



③ Index Register (X1, X2) Indirect



b) RAM Indirect



INSTRUCTIONS

(Note) '*' Addressing mode expression
 '#' and 'imme' Immediate value

Data Transfer

Instruction	Function
L A, *	A ← *
ST A, *	A → *
MOV *, A	A ← *
MOV *, #	* → Imme.
MOV er, *	er ← *
MOV d, *	(d) ← *
MOV DP, *	DP ← *
MOV X1, *	X1 ← *
MOV X2, *	X2 ← *
MOV USP, *	USP ← *
MOV PSW, *	PSW ← *
MOV SSP, *	SSP ← *
MOV LRB, *	LRB ← *
CLR *	* ← 0
SWAP	A ₁₅₋₈ ↔ A ₇₋₀
XCHG A, *	A ↔ *
XNBL A, *	A ₃₋₈ ↔ * ₃₋₀

Instruction	Function
LB A, *	A _L ← *
STB A, *	A _L → *
MOVB A, *	A _L ← *
MOVB *, A	* ← A _L
MOVB *, #	* ← Imme.
MOVB r, *	r ← *
MOVB d, *	(d) ← *
MOVB PSW0, *	PSW0 ← *
MOVB SCB, *	SCB ← *
CLRB *	* ← 0
SWAPB	A ₁₅₋₈ ↔ A ₇₋₀
XCHGB A, *	A ↔ *

Push & Pop

Instruction	Function
PUSHS * , A	* → SYSTEM STACK
POPS * , #	* ← SYSTEM STACK

Rotate & Shift

Instruction	Function
ROL *	Rotate left
ROR *	Rotate right
SLL *	Shift left Logical
SRL *	Shift right Logical
SRA *	Shift right Arithmetic

Instruction	Function
ROLB *	Rotate left
RORB *	Rotate right
SLLB *	Shift left Logical
SRLB *	Shift right Logical
SRAB *	Shift right Arithmetic

Increment & Decrement

Instruction	Function
INC *	* ← * + 1
DEC *	* ← * - 1

Instruction	Function
INCB *	* ← * + 1
DECB *	* ← * - 1

ROM Table Reference

Instruction	Function
LC A , *	A ← * (ROM)
CMPC A , *	* - * (ROM)

Instruction	Function
LCB A , *	A _L ← * (ROM)
CMPCB A , *	A _L - * (ROM)

Arithmetic Operation

Instruction	Function
MUL	$er1: A \leftarrow A \times er0$
DIV	$er0: A \leftarrow er0: A / er2$
ADD A, *	$A \leftarrow A + *$
ADD *, A	$* \leftarrow * + A$
ADD *, d	$* \leftarrow * + (d)$
ADD *, #	$* \leftarrow * + Imme.$
ADC A, *	$A \leftarrow A + * + C$
ADC *, A	$* \leftarrow * + A + C$
ADC *, d	$* \leftarrow * + (d) + C$
ADC *, #	$* \leftarrow * + Imme. + C$
SUB A, *	$A \leftarrow A - *$
SUB *, A	$* \leftarrow * - A$
SUB *, d	$* \leftarrow * - (d)$
SUB *, #	$* \leftarrow * - Imme.$
SBC A, *	$A \leftarrow A - * - C$
SBC *, A	$* \leftarrow * - A - C$
SBC *, d	$* \leftarrow * - (d) - C$
SBC *, #	$* \leftarrow * - Imme. - C$

Instruction	Function
MULB	$A_L \leftarrow A_L \times r0$
DIVB	$A \leftarrow A / r0$
ADDB A, *	$A_L \leftarrow A_L + *$
ADDB *, A	$* \leftarrow * + A_L$
ADDB *, d	$* \leftarrow * + (d)$
ADDB *, #	$* \leftarrow * + Imme.$
ADCB A, *	$A_L \leftarrow A_L + * + C$
ADCB *, A	$* \leftarrow * + A_L + C$
ADCB *, d	$* \leftarrow * + (d) + C$
ADCB *, #	$* \leftarrow * + Imme. + C$
SUBB A, *	$A_L \leftarrow A_L - *$
SUBB *, A	$* \leftarrow * - A_L$
SUBB *, d	$* \leftarrow * - (d)$
SUBB *, #	$* \leftarrow * - Imme.$
SBCB A, *	$A_L \leftarrow A_L - * - C$
SBCB *, A	$* \leftarrow * - A_L - C$
SBCB *, d	$* \leftarrow * - (d) - C$
SBCB *, #	$* \leftarrow * - Imme. - C$

Logical Operation

Instruction	Function
AND A, *	$A \leftarrow A \text{ and } *$
AND *, A	$* \leftarrow * \text{ and } A$
AND *, d	$* \leftarrow * \text{ and } (d)$
AND *, #	$* \leftarrow * \text{ and Imme.}$
OR A, *	$A \leftarrow A \text{ or } *$
OR *, A	$* \leftarrow * \text{ or } A$
OR *, d	$* \leftarrow * \text{ or } (d)$
OR *, #	$* \leftarrow * \text{ or Imme.}$
XOR A, *	$A \leftarrow A \text{ xor } *$
XOR *, A	$* \leftarrow * \text{ xor } A$
XOR *, d	$* \leftarrow * \text{ xor } (d)$
XOR *, #	$* \leftarrow * \text{ xor Imme.}$

Instruction	Function
ANDB A, *	$A_L \leftarrow A_L \text{ and } *$
ANDB *, A	$* \leftarrow * \text{ and } A_L$
ANDB *, d	$* \leftarrow * \text{ and } (d)$
ANDB *, #	$* \leftarrow * \text{ and Imme.}$
ORB A, *	$A_L \leftarrow A_L \text{ or } *$
ORB *, A	$* \leftarrow * \text{ or } A_L$
ORB *, d	$* \leftarrow * \text{ or } (d)$
ORB *, #	$* \leftarrow * \text{ or Imme.}$
XORB A, *	$A_L \leftarrow A_L \text{ xor } *$
XORB *, A	$* \leftarrow * \text{ xor } A_L$
XORB *, d	$* \leftarrow * \text{ xor } (d)$
XORB *, #	$* \leftarrow * \text{ xor Imme.}$

Comparison

Instruction	Function
CMP A, *	$A - *$
CMP *, A	$* - A$
CMP *, d	$* - (d)$
CMP *, #	$* - \text{Imme.}$

Instruction	Function
CMPB A, *	$A_L - *$
CMPB *, A	$* - A_L$
CMPB *, d	$* - (d)$
CMPB *, #	$* - \text{Imme.}$

Decimal Adjust

Instruction	Function
DAA	Decimal Adjust for Addition
DAS	Decimal Adjust for Substruct

Data Type Conversion

Instruction	Function
EXTND	A15 - 8 ← An

Bit Operation

Instruction	Function
SBR *	bit ← 1
RBR *	bit ← 0
MBR C, *	C ← bit
MBR *, C	bit ← C
TRB *, C	Z ← bit

Instruction	Function
SBR *	bit ← 1
RBR *	bit ← 0
MBR C, *	C ← bit
MBR *, C	bit ← C

Jump & Call

Instruction	Function
SJ adrs	Short Jump
J *	Jump
JC EQ, adrs	Jump if '='
JC LE, adrs	Jump if '<='
JC GE, adrs	Jump if '>='
JBS bit, adrs	Jump if bit-on
JRNZ DP, adrs	Loop Function

Instruction	Function
SCAL adrs	Short Call
CAL *	Call Subroutine
JC NE, adrs	Jump if '<>'
JC LT, adrs	Jump if '<'
JC GT, adrs	Jump if '>'
JBR bit, adrs	Jump if bit-off
VCAL table:adrs	Vector Call

Return

Instruction	Function
RT	Return from Subroutine
RTI	Return from Interrupt

Others

Instruction	Function
SC	$C \leftarrow 1$
SS	STACK FLAG $\leftarrow 1$
NOP	No Operation

Instruction	Function
RC	$C \leftarrow 0$
BRK	Software Reset

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	- 0.3~7.0	V
Input Voltage	V_I		- 0.3~ $V_{DD} + 0.3$	V
Output Voltage	V_O		- 0.3~ $V_{DD} + 0.3$	V
Analog Ref. Voltage	V_R		- 0.3~ $V_{DD} + 0.3$	V
Analog Input Voltage	V_{AI}		- 0.3~ V_R	V
Power Dissipation	P_D	Ta = 25°C (per package)	400 MAX	mW
		Ta = 25°C (per out)	50 MAX	mW
Storage Temperature	T_{STG}	-	- 55~ + 150	°C

OPERATING CONDITIONS

Parameter	Symbol	Conditions	Limits	Unit	
Supply Voltage	V_{DD}	$f_{(OSC)} \leq 10 \text{ MHz}$	4.5~5.5	V	
Memory Hold Voltage	V_{DDH}	$f_{(OSC)} = 0 \text{ Hz}$	2~5.5	V	
Operating Frequency	$f_{(OSC)}$	$V_{DD} = 5V \pm 10\%$	0~10	MHz	
Ambient Temperature	T_a	MSM66207	-40~+85	°C	
		MSM66P207	0~+70		
Fan Out	P_D	MOS load	20		
		TTL load	P0	2	
			P1, P2, P3, P4	1	