OKI semiconductor

MSM6351

CMOS 4BIT HIGH PERFORMANCE AND VERY LOW POWER SINGLE CHIP MICROCONTROLLER WITH LCD DRIVER

GENERAL DESCRIPTION

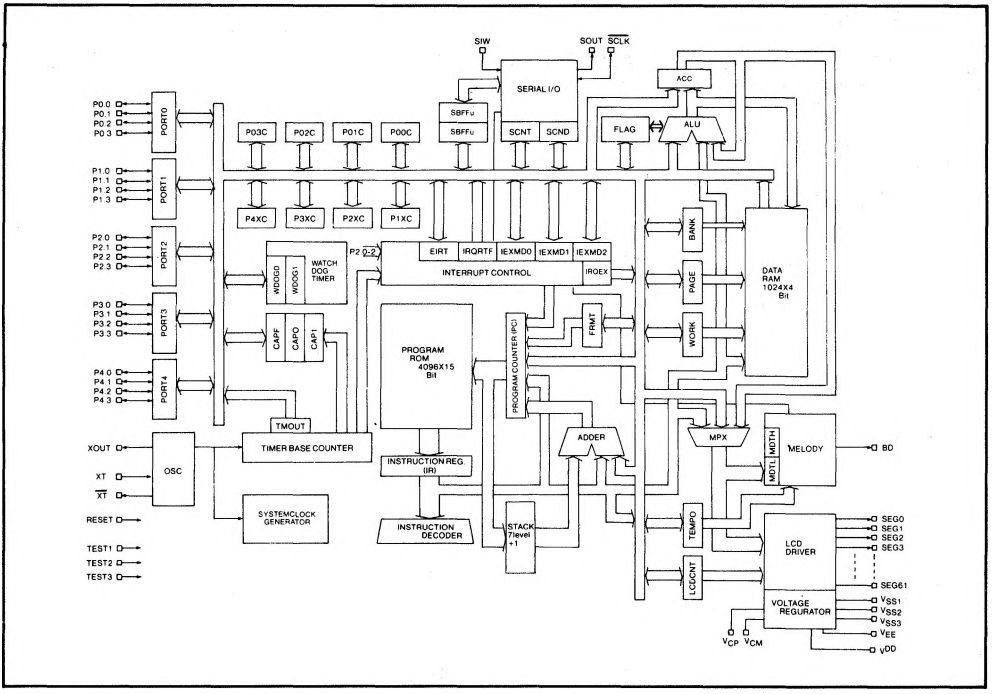
OKI's MSM6351 is a low-power, high-performance single-chip microcontroller employing silicon gate CMOS technology. Integrated onto a single chip are 4-bit ALU, 61K bits of mask programmable ROM, 4096 bits of RAM, 20 bits of I/O port, serial I/O port, time-base counter, LCD driver, 3 interrupts, crystal oscillator and voltage tripler.

The MSM6351 is widely used in electronic products requiring low power consumption, a large number of LCD drivers and a large size of memory.

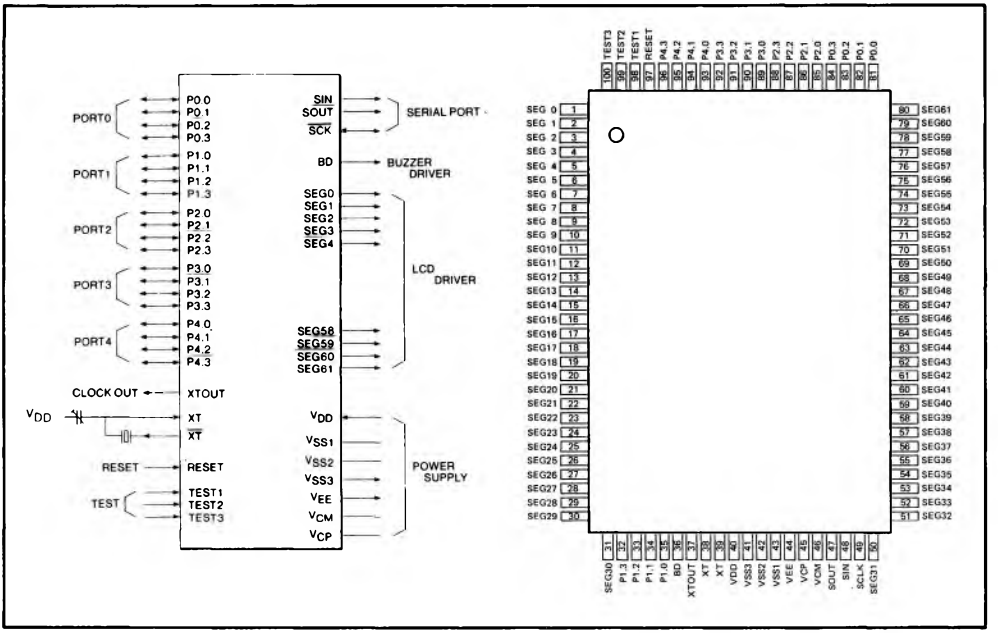
FEATURES

- Low Power Consumption 3 μA Typical
- 4096 × 15 Internal ROM
- 1024 × 4 Internal RAM
- 20 Input/Output Ports
- 62 LCD Drivers
  1/3 Duty, 1/3 Bias or 1/4 Duty 1/3 Bias
  (Selectable by software)
- Serial I/O Port
  8 bits or 5 bits data frame mode
  Asynchronous
  receiver/transmitter mode
  Internal or external clock mode
- 15 stages Time Base Counter
- Watch Dog Timer
- Capture function by external trigger signal
- 3 Interrupt Sources
  Real time interrupt
  External interrupt
  Serial I/O port interrupt
- Melody Circuit
- 65 Instructions
- Sub-routine Nesting: 7 levels
- 32.768 kHz Crystal Oscilator
- Machine Cycle: 61.0 μsec.
- Power Supply: 1.5V or 3.0V (selectable by mask option)
- 100 pad die or 100 Pin Flat Package
- Silicon gate CMOS Process

FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



PIN DESCRIPTION

<table>
<tr><th>Designation</th><th colspan="4">Function</th></tr>
<tr><td>P0.0</td><td rowspan="4">PORT 0</td><td colspan="2" rowspan="4">• 4-bit I/O port 0
The input (*)/output, the existence (*)/ absence of pull-down resistance, and the HALT function release enable/disable condition can be selected for each bit.</td><td>Capture trigger signal</td></tr>
<tr><td>P0.1</td><td></td></tr>
<tr><td>P0.2</td><td rowspan="2"></td></tr>
<tr><td>P0.3</td></tr>
<tr><td>P1.0 ~ P1.3</td><td>PORT 1</td><td>• 4-bit I/O port 1</td><td rowspan="5">The input (*)/output, the existence (*)/absence of pull-down resistance, and the HALT function release enable/disable condition can be selected for each bit.</td><td></td></tr>
<tr><td>P2.0</td><td rowspan="4">PORT 2</td><td rowspan="4">• 4-bit I/O port 2</td><td rowspan="4">External interrupt signal</td></tr>
<tr><td>P2.1</td></tr>
<tr><td>P2.2</td></tr>
<tr><td>P2.3</td></tr>
<tr><td>P3.0 ~ P3.3</td><td>PORT 3</td><td>• 4-bit I/O port 3</td><td></td></tr>
<tr><td>P4.0 ~ P4.3</td><td>PORT 4</td><td>• 4-bit I/O port 4</td><td></td></tr>
<tr><td>XTOUT</td><td colspan="4">• Oscillator clock output
The oscillator clock is output when XTF (forth bit of port P00C) is set to "1".</td></tr>
<tr><td>XT</td><td colspan="4" rowspan="2">• Oscillator connection terminal</td></tr>
<tr><td>XT</td></tr>
<tr><td>RESET</td><td colspan="4">• Reset input
Input with a pull-down resistance, the system is reset when "1" is input.</td></tr>
<tr><td>TEST 1</td><td colspan="4" rowspan="3">• Test input
Input with a pull-down resistance. Generally used when the resistance is open.</td></tr>
<tr><td>TEST 2</td></tr>
<tr><td>TEST 3</td></tr>
<tr><td>SIN</td><td colspan="4">• Circuit configuration</td></tr>
<tr><td>SOUT</td><td colspan="4">• Serial port data input
the circuit is set at high impedance level when "1" is set to HZOUT (the forth bit of port P4XC) or when no data is transmitted.</td></tr>
<tr><td>SCK</td><td colspan="4">• Serial port clock input and output
the input and output (*) are switched from each other the serial port control register SCNT. In the output mode, the serial clock frequency can be selected from the demultiplied signal (1/1, 1/2 or 1/4 of the original base clock) (see note 1).</td></tr>
<tr><td>BD</td><td colspan="4">• Melody output (buzzer drive output) (not available in the MSM6353.)</td></tr>
<tr><td>SEG0 ~ SEG61</td><td colspan="4">• LCD drive output
LCD drive output with the 1/3 bias, or the 1/3 or 1/4 dusty system.
The duty can be switched by LCD control register LCDCNT.
A maximum of 177 segments can be displayed with the 1/3 duty and 232 with the 1/4 duty.</td></tr>
</table>

Designation	Function
V_{DD}	• 0V power supply terminal
V_{SS1}	• –1.5V power terminal (for the 1.5V specification)
V_{SS2}	• –3.0V power terminal (for the 3.0V specification)
V_{SS3}	• –4.5V power terminal (not available in the MSM6353)
V_{EE}	• Internal logic power terminal
V_{CM}	• Internal power generator capacitor connecting terminal
V_{CP}	

Note 1: The base clock refers to an oscillator output signal demultiplied into 1/1, 1/2, 1/4, or 1/8 of the output frequency (with the masking option).

FUNCTIONAL DESCRIPTION

A block diagram of MSM6351 is given on page 128. Each block of logic will be briefly discussed. For further information, please refer to MSM6351/53 user's manual.

Programmable ROM

The programmable ROM has a capacity of 4096 words, each of which is 15 bits long. It is provided with the address space of 000 to FFFH.

In the MSM6351, the programmable ROM is not only used for programming but also used to save the following items:

1) LCD indicator segment conversion table
2) Melody tone data

The program instructions are all mde up of one word, thus the ROM can save up to 4096 instructions in it.

Data RAM

The data RAM has a capacity of 1024 words, each consisting of four bits. It is provided with the space for address between 000H and 3FFH. Data is organized in 4 bit nibble.

Page Register (PAGE)

The page register specifies one of 16 pages in each bank of the data RAM. It is used in the addressing bank mode or in the paging mode.

Bank Register (BANK)

The bank register specifies one of four banks in the data RAM. It is used together with the page register in the bank paging mode.

Working Specification Register

The working specification register specifies one of 16 pages in bank 0 of the data RAM. It is used in the addressed working specification mode.

Operational Section

The operational section consists of the ALU, accumulator (ACC), and conditional flags C, Z and G.

This operational section performs four-bit computation of the contents of the data RAM with the contents of ACC or the immediate data fetched into the instruction words.

This computation is mainly performed with the data RAM which functions as a register. The resultant data of the computation are input to the data RAM or to ACC (for operation other than comparison).

Program Counter (PC)

The program counter (PC) generates addresses for the programmable ROM.

The addresses for programming are changed according to the instructions executed. These addresses are incremented by one each time the instruction is executed.

When an interrupt is generated, the current address is stored in the STACK. The address is set to 400H, 401H or 402H depending on the type of the interrupt (see Fig. 3.5.1).

These addresses are set to the start addresses of each interrupt routine.

In the MSM6351, the PC also gives LCD indicator "segment conversion table" or melody "tone data" addresses to the programmable ROM.

The output data of the programmable ROM whose address is specified by the PC is fetched into the instruction register (IR). If the output data is an instruction, it is decoded by the instruction decoder. Then, control signals to each section are generated.

Ports

The MSM6351/6353 handle the I/O ports, flags and registers collectively as ports. Therefore, each of the I/O ports, flags and registers are selected by specifying their own addresses.

All of these ports are accessed by the INP and OUT instructions.

PORT NAMES, ADDRESSES AND THEIR CONTENTS

Port name	Address	Bit 3	Bit 2	Bit 1	Bit 0	Access mode (*)
PORT0	00	P0.3	P0.2	P0.1	P0.0	R/W
PORT1	01	P1.3	P1.2	P1.1	P1.0	R/W
PORT2	02	P2.3	P2.2	P2.1	P2.0	R/W
PORT3	03	P3.3	P3.2	P3.1	P3.0	R/W
PORT4	04	P4.3	P4.2	P4.1	P4.0	R/W
P00C	05	XTF	HRE00	HZ00	DIR00	R/W
P01C	06	BUF	HRE01	HZ01	DIR01	R/W
P02C	07	–	HRE02	HZ02	DIR02	R/W
P03C	08	–	HRE03	HZ03	DIR03	R/W
P1XC	09	–	HRE1X	HZ1X	DIR1X	R/W
P2XC	0A	HZSOUT	HRE2X	HZ2X	DIR2X	R/W
P3XC	0B	EISIO	HRE3X	HZ3X	DIR3X	R/W
P4XC	0C	5/8	HRE4X	HZ4X	DIR4X	R/W
SBF	0D	(L) d_3	d_2	d_1	d_0	R/W
		(u) d_7	d_6	d_5	d_4	
SCNT	0E	CLKSL1	CLKSL0	MODE	LSB/MSB	R/W
SCND	0F	SIOEND	STPErr	ENRC	ENTR	BIT: 3 and 2; R 1 and 0; R/W
IRQEX	10	–	IRQP22	IRQP21	IRQP20	R
EIRT	11	EI256Hz	EI32Hz	EI16Hz	EI1Hz	R/W
IRQRT	12	IRQ256Hz	IRQ32Hz	IRQ16Hz	IRQ1Hz	R
IEXM0	13	–	EIP20	L/E0	P/N0	R/W
IEXM1	14	–	EIP21	L/E1	P/N1	R/W
IEXM2	15	–	EIP22	L/E2	P/N2	R/W
TMOUT	16	15th	14th	13th	12th	R
CAPRT	17	128Hz / 32Hz	256Hz / 64Hz	512Hz / 128Hz	1KHz / 256Hz	R
		–	CAPMD	CAP1F	CAP0F	Bity 2: R/W Other bits: R
FLAG	18	MSTART	G	Z	C	Bity 3: R Other bits: R/W
WDOG	19	W_{03}	W_{02}	W_{01}	W_{00}	R/W
		W_{13}	W_{12}	W_{11}	W_{10}	
FRAMT	1A	F_3	F_2	F_1	F_0	R/W
LCDCT	1B	AIION	Drty 3/4	FLM1	FLM0	R/W

PORT NAMES, ADDRESSES AND THEIR CONTENTS (Continued)

Port name	Address	Bit 3	Bit 2	Bit 1	Bit 0	Access mode (*)
TEMPO	1C	P_3	P_2	P_1	P_0	R/W
BANK	1D	b_3	b_2	b_1	b_0	R/W
PAGE	1E	P_3	P_2	P_1	P_0	R/W
WORK	1F	w_2	w_2	w_1	w_0	R/W

Note: • In access mode (*), R denotes "readable" bits and W "writable" bits.
• Bits marked with denote bits which are not present.

I/O Ports

The MSM6351/6353 are provided with five ports; PORT0 to PORT4. Each port consists of four bits.

Ports are controlled by the I/O port control register. The register controls PORT1 to PORT4 port by port and controls PORT0 in bit units.

Each port is accessed by the OUT and INP instruction.

Time Base Counter

The MSM6351/6353 have their built-in time base counters consisting of a 15-stage binary counter. System base clock $\phi 0$ is input to the time base counter clock.

Capture Circuit

The MSM6351/6353 are provided with a capture function that fetches the 1KHz to 128Hz outputs at stages 5 to 8 of the time base counter or the 256 Hz to 32Hz outputs at stages 7 to 10 when P0.0 or P0.1 of I/O port 0 is set at the "H" level.

Watchdog Timer

The MSM6351/6353 have their built-in watchdog timer to prevent any program runway occurrence. The time may be set with two types of setting time: 250ms and 2s.

Serial Port (SERIAL I/O)

The MSM6351/6353 have their built-in serial port. It is used for asynchronous data communications. A data length of five or eight bits can be selected. Either internal or external clock can be selected as the driving clock. At the end of data transfer, a serial port inerrupt can be generated.

The serial port registers and their functions are described in the following sections.

Melody Output Circuit (built in the MSM6351) (MEMODY)

The melody output circuit automatically outputs melody or buzzer sound. It is built in the MSM6351.

The melody circuit initiates its operation by the MSA instruction. Automatically fetching the musical note data defined in the program ROM, the MSA instruction outputs the melody from buzzer driving output terminal BD.

Liquid Crystal Display Circuit (LCD DRIVER)

The MSM6351 has its built-in liquid crystal display circuit that can drive the liquid crystal (LCD).

The liquid crystal display consists of the display data register for writing the data to indicate and the display driver. After data is written in the display data register with an display instruction, the display driver automatically fetches data from the display data register to output the driving waveform.

Interruption Controller (INTERRUPT CONTROL)

There are four types of interruptions as follows:

1) Real-time interruption — Interruption with the time base counter output
2) External interruption — Interruption from PORT2
3) Serial port interruption — Interruption by terminating serial port data transfer
4) Melody interruption — Interruption by requesting melody data (not available in the MSM6353)

ABSOLUTE MAXIMUM RATING (Target Specification)

V_{DD} = 0V (V_{SS2} = Battery Voltage)

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−6.0 ~ +0.3	V
Input Voltage	V_{IN}		V_{SS2}−0.3 ~ +0.3	V
Output Voltage*1	V_{01}		V_{SS2}−0.3 ~ +0.3	V
Output Voltage*2	V_{02}		−6.0 ~ +0.3	V
Storage Temperature	T_{STG}		−55 ~ +125	°C

*1 Normal Output
*2 When LCD Driver in use

OPERATING CONDITION (Target Specification)

Parameter	Symbol	Limits	Units
Operating Voltage	$-V_{SS2}$	2.6 ~ 3.3	V
Operating Temperature	Topr	−20 to 60	°C

DC CHARACTERISTICS 3V Li Battery (Target Specification)

V_{DD} = 0V, V_{SS1} = −1.5V, V_{SS2} = −3.0V, V_{SS3} = −4.5V, f = 32,768Hz, Ta = 25°C*1

Parameter	Symbol	Donditions	Rating			Unit	Terminal Applied
			Min.	Typ.	Max.		
Current Consumption	I_{DD}	*3	–	3.0	–	μA	
Voltage for Oscillation Start	$-V_{OSC}$	Within 2 sec.	–	–	2.4	V	
Output Current 1 (Common Segment)	$-I_{OH1}$	V_{OH1} = −0.2V	4	–	–	μA	SEG0 ~ SEG61
	$\|I_{OMH1}\|$	V_{OMH1} = V_{SS1}±0.2V	4	–	–		
	$\|I_{OML1}\|$	V_{OML1} = V_{SS2}±0.2V	4	–	–		
	I_{OL1}	V_{OL1} = −4.3V	4	–	–		
Output Current 2	$-I_{OH2}$	V_{OH2} = −0.5V	500	–	–	μA	PORT0~PORT4*2 S_{OUT}, $S\overline{CLK}$ X_{TOUT}
	I_{OL2}	V_{OL2} = −2.5V	500	–	–		
Output Current 3	$-I_{ON3}$	V_{ON4} = −0.5V V_{SS2} = 3.0V	7	–	–	μA	BD
	I_{OL3}	V_{ON4} = −2.5V V_{SS2} = 3.0V	20	–	–		
Input Current 1	$-I_{IN1}$	V_{IN1} = 0V I/O input, with pull down	150	300	600	μA	PORT0~PORT4
Input Current 2	$\|I_{I2}\|$	V_{IN2} = 0V, −3V I/O input, without pull down	–	–	1	μA	PORT0~PORT4 S_{IN}, $S\overline{CLK}$
Input Current 3	$-I_{IH3}$	V_{IH3} = 0V with pull down	–	25	–	μA	RESET
Input Voltage	$-V_{IH}$		–	–	0.5	V	All input terminals
	$-V_{IH}$		2.5	–	–		

*1. When 3V battery with halver is used.

*2. PORT0 = P0.0 ~ P0.3, PORT1 = P1.0 ~ P1.3, PORT2 = P2.0 ~ P2.3, PORT 3 = P3.0 ~ P3.3, PORT4 = P4.0 ~ P4.3

*3. This value changes depending on the soft duty (HALT to HALT)

INSTRUCTION LIST

	Mnemonic	Instruction code																Description	Machine cycle	Page
		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Operation instruction	ADD ACC, REG1	0	0	0	0	0	0	P	0	1	0	0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← (rP) + (ACC)	1	
	ADD #i, REG1	0	0	1	1	0	0	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← (rP) + i	1	
	ADC REG1	0	0	0	0	0	0	P	0	1	0	1	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← decimal adj [(rP) + (ACC) + (C)]	1	
	ADCN REG1	0	1	1	0	0	0	P	N_3	N_2	N_1	N_0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← N adjust [(rP) + (C)]	1	
	SUB ACC, REG1	0	0	0	0	0	1	P	0	1	0	0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← (rP) − (ACC)	1	
	SUB #i, REG1	0	0	1	1	0	1	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← (rP) − i	1	
	SBC REG1	0	0	0	0	0	1	P	0	1	0	1	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← decimal adj [(rP) − (ACC) − (C)]	1	
	SBCN REG1	0	1	1	0	0	1	P	N_3	N_2	N_1	N_0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← N adjust [(rP) − (C)]	1	
	CMP ACC, REG1	0	0	0	0	0	1	P	1	1	1	0	r_3	r_2	r_1	r_0		(Z), (G) ← (rP) − (ACC)	1	
	CMP #i, REG1	0	0	1	0	1	1	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0		(Z), (G) ← (rP) − i	1	
	OMC REG1	0	0	0	0	0	0	P	0	0	0	1	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← (rP) + 1	1	
	INCD REG2	1	0	0	0	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0		(rPb), (ACC), (Z), (C) ← (rPb) + 1	1	
	DEC REG1	0	0	0	0	0	1	P	0	0	0	1	r_3	r_2	r_1	r_0		(rP), (ACC), (Z), (C) ← (rP) − 1	1	
	DECD REG2	1	0	0	1	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0		(rPb), (ACC), (Z), (C) ← (rPb) − 1	1	
Bit operation instruction	BIT ACC, REG1	0	0	0	0	0	0	P	1	1	1	0	r_3^3	r_2	r_1^1	r_0		(Z) ← $\overline{(rP_3)} \wedge (ACC_3) \vee \overline{(rP_2)} \wedge (ACC_2) \vee \overline{(rP_1)} \wedge (ACC_1) \vee \overline{(rP_0)} \wedge (ACC_0)$	1	
	BIT #i, REG1	0	0	1	0	1	0	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0		(Z) ← $\overline{(rP_3)} \wedge i_3 \vee \overline{(rP_2)} \wedge i_2 \vee \overline{(rP_1)} \wedge i_1 \vee \overline{(rP_0)} \wedge i_0$	1	
	BIS ACC, REG1	0	0	0	0	0	0	P	0	1	1	0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z) ← (rP) ∨ (ACC)	1	
	BIS #i, REG1	0	0	1	0	0	0	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z) ← (rP) ∨ i	1	
	BIC ACC, REG1	0	0	0	0	0	1	P	0	1	1	0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z) ← (rP) ∧ $\overline{(ACC)}$	1	
	BIC #1, REG1	0	0	1	0	0	1	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z) ← (rP) ∧ $\overline{i}$	1	
	XOR ACC, REG1	0	0	0	0	0	0	P	0	1	1	1	r_3	r_2	r_1	r_0		(rP), (ACC), (Z) ← (rP) ∨ (ACC)	1	
	XOR #i, REG1	0	0	1	1	1	1	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0		(rP), (ACC), (Z) ← (rP) ∀ i	1	

INSTRUCTION LIST (Continued)

	Mnemonic	Instruction code															Description	Machine cycle	Page
		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Rotate instruction	ROR REG1	0	0	0	0	0	0	P	0	0	1	0	r_3	r_2	r_1	r_0	(Z), (ACC) ← [↳ (C) → (rP) ↵]	1	
	ROL REG1	0	0	0	0	0	1	P	0	0	110	r_3	r_2	r_1	r_0	r	(Z), (ACC) ← [↰ (C) ← (rP) ↵]	1	
	ASR REG1	0	0	0	0	0	0	P	0	0	1	1	r_3	r_2	r_1	r_0	(Z), (ACC) ← [O → (rP) → (C)]	1	
	ASL REG1	0	0	0	0	0	1	P	0	0	1	1	r_3	r_2	r_1	r_0	(Z), (ACC) ← [(C) ← (rP) ← O]	1	
Flag operation instruction	CLG	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	(G) ← O	1	
	CLC	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	(C) ← O	1	
	CLZ	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	(Z) ← O	1	
	CLA	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	(Z), (C), (G) ← O	1	
	SEG	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	(G) ← 1	1	
	SEC	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	(C) ← 1	1	
	SEZ	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	(Z) ← 1	1	
	SEA	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	(Z), (C), (G) ← 1	1	
Date transfer instruction	MOV ACC, REG1	0	0	0	0	0	0	P	1	1	1	1	r_3	r_2	r_1	r_0	(rP) ← (ACC)	1	
	MOVD ACC, REG2	1	0	1	0	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0	(rPb) ← (ACC)	1	
	MOV #i, REG1	0	0	1	1	1	0	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z) ← i	1	
	MOV REG1, ACC	0	0	0	0	0	1	P	1	1	1	1	r_3	r_2	r_1	r_0	(ACC), (Z) ← (rP)	1	
	MOVD REG2, ACC	1	0	1	1	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0	(ACC), (Z) ← (rPb)	1	
	EXG REG1	0	0	0	0	0	1	P	0	0	0	0	r_3	r_2	r_1	r_0	(rP) ↔ (ACC)	1	
	EXGD REG2	0	1	1	1	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0	(rPb) ↔ (ACC)	1	
Sub-routine instruction	CALL adrs	1	1	1	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	(STACK) o (PC), (PC) ← a_{11} ~ a_0 (SP) ← (SP) + 1	1	
	RET	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	(PC) ← (STACK) + 1 (SP) ← (SP) − 1	1	
	RT1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	(PC) ← (STACK) + 1 (SP) ← (SP) − 1 (at INT routine)	1	
Jump instruction	JMP adrs	1	1	0	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	(PC) ← a_{11} ~ a_0	1	
	JMP a REG1	0	0	0	0	0	0	P	1	1	0	1	r_3	r_2	r_1	r_0	(PC) ← (PC) + (rP) + 1	1	
	JMPIO a REG1	0	0	0	0	0	1	P	1	1	0	1	r_3	r_2	r_1	r_0	(PC) ← (PC) + 7 A (rP) + 1	1	

INSTRUCTION LIST (Continued)

	Mnemonic	Instruction code 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description	Machine cycle	Page
Jump instruction	BGT n	0	0	0	0	1	1	P	0	0	1	n_4	n_3	n_2	n_1	n_0	if (G) = 1 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1	
	BLE n	0	0	0	0	1	1	P	1	0	1	n_4	n_3	n_2	n_1	n_0	if (G) = 0 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1	
	BCS n	0	0	0	0	1	1	P	0	0	0	n_4	n_3	n_2	n_1	n_0	if (C) = 1 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1	
	BCC n	0	0	0	0	1	1	P	1	0	0	n_4	n_3	n_2	n_1	n_0	if (C) = 0 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1	
	BEQ n (BZE n)	0	0	0	0	1	1	P	0	1	0	n_4	n_3	n_2	n_1	n_0	if (Z) = 1 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1	
	BNE n (BNZ n)	0	0	0	0	1	1	P	1	1	0	n_4	n_3	n_2	n_1	n_0	if (Z) = 0 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1	
	BGE n	0	0	0	0	1	1	P	0	1	1	n_4	n_3	n_2	n_1	n_0	if [(G) = 1 or (Z) = 1] then (PC) ← (PC) + N else (PC) → (PC) + 1	1	
	BLT n	0	0	0	0	1	1	P	1	1	1	n_4	n_3	n_2	n_1	n_0	if [(G) = 0 or (Z) = 0] then (PC) ← (PC) + N else (PC) ← (PC) + 1	1	
																	When P = 0 in bit 8, N = n + 1; when P = 1, N = −n		
Melody start	MSA adrs*	0	0	0	0	1	0	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Specifies the first address of note data. ($E00_H$ ~ FFF_H)	2	
Display instruction	DSP dig, REG1*	0	0	0	1	0	0	P	dig_3	dig_2	dig_1	dig_0	r_3	r_2	r_1	r_0	digit (Lower Part) ← (rP), (ACC)	1	
	DSPH dig, REG1*	0	0	0	1	0	1	P	dig_3	dig_2	dig_1	dig_0	r_3	r_2	r_1	r_0	digit (High Part) ← (rP), (ACC)	1	
	DSPF dig, REG1*	0	0	0	1	1	0	P	dig_3	dig_2	dig_1	dig_0	r_3	r_2	r_1	r_0	digit (Low Part) ← (rP) via Table	2	
	DSPFH dig, REG1*	0	0	0	1	1	1	P	dig_3	dig_2	dig_1	dig_0	r_3	r_2	r_1	r_0	digit (High Part) ← (rP) via Table	2	
Input/output instruction	OUT REG1, PORT	0	1	0	1	0	y_4	P	y_3	y_2	y_1	y_0	r_3	r_2	r_1	r_0	(PORT y) ← (rP)	1	
	OUT #i, PORT	0	1	0	1	1	y_4	0	y_3	y_2	y_1	y_0	i_3	i_2	i_1	i_0	(PORT y) ← i	1	
	INP PORT, REG1	0	1	0	0	0	y_4	P	y_3	y_2	y_1	y_0	r_3	r_2	r_1	r_0	(rP), (ACC) ← (PORT y)	1	
CPU control	NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	No Operation	1	
	HALT	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	Halt CPU	1	

Note: Instructions marked with an asterisk (*) are available for MSM6351 only.