Visual Memory System Hardware Manual
Revision 1.00

Introduction

Thank you for your cooperation in developing applications for Sega hardware. This document contains the technical hardware specifications for the “Visual Memory System” (VMS) memory card designed for use with the Dreamcast game machine.

Contents of this manual

Overview

Contains a general description of the Visual Memory System. Game planners and designers as well as graphics developers should consult this section.

Detailed Hardware Specifications

Contains detailed descriptions of Visual Memory System devices and interfaces. Programmers should consult this section.

BIOS

The Visual Memory System comprises a BIOS. The BIOS programs serve for reading and writing to the flash memory area where game data are stored and for getting date and time information from the internal clock. This section describes the use of the BIOS programs and provides reference information.

Note

This document is based on the following documents: “VMS Specifications Revision 0.90”, “VMS Hardware Manual Version 1.01”, “VMS Sound Development Specifications Revision 0.60” (published by Sega Enterprises. Inc.).

VMS specifications and other items contained in this manual may be changed in future without notice.

Sega Enterprises. Inc. is not responsible for any consequences arising out of the use of VMS and of this manual.

Trademarks

Dreamcast and Visual Memory System are trademarks of Sega Enterprises. Inc. Other products mentioned in the manual are normally trademarks of their respective manufacturers, but the trademark indications TM and (R) are omitted in this manual.

Revision History

1998/08/31 Initial release

1998/11/30 Second printing

Copyright (C) 1998 Sega Enterprises. Inc.

Production: Ascii Corporation AAP Editing Department

Other Related Documents

VMS Manuals

VMS Hardware Manual

The current manual. Contains detailed VMS hardware specifications and system BIOS specifications. The first section gives a general overview of VMS specifications.

Game designers should refer to the overview and programmers to the entire manual.

VMS Programmer's Guide

VMS incorporates the LC86700 from Sanyo Electric. This manual describes how to use the assembler, linker, and library manager for this chip and how to install and use E2H86K.EXE and H2BIN that serve for creating VMS executable files. The manual also describes the instruction set of the LC86700.

VMS Tutorial

This document describes how to develop VMS applications and how to transfer these from the development system to the VMS. It also contains sample programs with detailed annotations.

VMS Simulator Guide

Describes the VMS simulator that emulates VMS operation in software.

Limitations

RC oscillator tolerance

The frequency of the RC oscillator circuit is subject to tolerances in the range of ± 30%. The reference value for the circuit set at the factory is 879.236 kHz, but in actual use the frequency can range from a minimum of about 600 kHz to a maximum of 1200 kHz.

For applications where timing is critical, a quartz oscillator should be used.

Technical Support

For assistance with problems arising during development or with points that are not adequately covered in the manual, please contact the Technical Support Center.

Sega Enterprises. Inc.

Technical Support Center

Haneda 1-2-12, Ota-ku, Tokyo, Japan 144-8531

Tel: 81-3-5736-7355

Fax: 81-3-5736-5357

E-mail: katana@sft.sega.co.jp

contents

Introduction.....2

Other Related Documents 3

Limitations 4

Technical Support 5

Part 1
VMS Overview **17**

Chapter 1
Visual Memory System Overview **19**

1.1 VMS Specifications 20

1.2 VMS Functions 22

1.3 Mode Setting.....25

1.4 File Management 27

1.5 LCD Display 28

1.6 Starting an Executable File 30

1.7 Communication Functions.....33

1.8 Clock Function 33

1.9 Alarm Function.....34

1.10 Sleep Function.....34

1.11 Buttons....34
1.12 Batteries....35

Part 2
HardwareManual 37

Chapter 2
CPU Features 39

2.1 Differences to Conventional CPUs....39
2.2 Specifications....40
2.3 System block diagram....44

Chapter 3
Internal System Configuration 45

3.1 Memory Space....45
3.2 Program Counter (PC)....46
3.3 ROM Space....47
3.4 RAM Space....48
3.4.1 Indirect Address Registers....48
3.4.2 Special function registers (SFR)....49
3.5 Flash Memory....52
3.6 Accumulator....52
3.7 B Register, C Register....53
3.8 Program Status Word (PSW)....54

Contents

3.9 Stack Pointer 55

3.10 Table Reference Register (TRR) 56

3.11 CHANGE Instruction 57

Chapter 4 Peripheral System Configuration 59

4.1 I/O Ports 59

4.1.1 Port 1 60
4.1.2 Port 3 64
4.1.3 Port 7 66

4.2 Timer/Counter 0 (T0) 67

4.2.1 Functions 67
4.2.2 Circuit Configuration 68
4.2.3 Related Registers 69
4.2.4 Circuit Configuration and Operation Principles 77

4.3 Timer 1 (T1) 83

4.3.1 Functions 83
4.3.2 Circuit Configuration 84
4.3.3 Related Registers 85
4.3.4 Circuit Configuration and Operation Principles 89

4.4 Base Timer 101

4.4.1 Functions 101
4.4.2 Circuit Configuration 102
4.4.3 Related Registers 103
4.4.4 Using the Base Timer 106

4.5 Serial Interface 107

4.5.1 Functions and Features 108
4.5.2 Circuit Configuration 110
4.5.3 Related Registers 111
4.5.4 Serial Interface Operation 117
4.5.5 Operation Mode Settings 118
4.5.6 Serial transfer clock 119
4.5.7 Serial Transfer Timing 122
4.5.8 LSB/MSB Switchable Start Sequence 122

4.5.9 Overrun Detection....124
4.5.10 Transfer Bit Length Control....125
4.5.11 Sample Program....126

4.6 Dot Matrix LCD Controller....129
4.6.1 Functions....129
4.6.2 Display RAM (XRAM)....129
4.6.3 Display Control Registers....130

4.7 External Interrupt Function....136
4.7.1 Circuit Configuration....137
4.7.2 Related Registers....138

4.8 Port Interrupt Functions....142
4.8.1 Function....142
4.8.2 Circuit Configuration....143
4.8.3 Related Registers....143
4.8.4 Operation Description....144
4.8.5 State Transition....145

4.9 VMS Work RAM....146
4.9.1 Work RAM Control Registers....146
4.9.2 Accessing Work RAM....147
4.9.3 Precautions for Using Work RAM Address Register....148

4.10 Flash Memory....149
4.10.1 Features and Functions....149
4.10.2 Accessing Program/Data Area of Flash Memory....149

Chapter 5
Control Functions 151

5.1 Interrupt Functions....151
5.1.1 Interrupt Types....152
5.1.2 Interrupt Function Operation....153
5.1.3 Circuit Configuration....154
5.1.4 Related Registers....154
5.1.5 Interrupt Priority Ranking....156

5.2 System Clock Generation.....158
5.2.1 Features and Functions 158
5.2.2 Circuit Configuration.....159
5.2.3 Related Registers.....160
5.2.4 System Clock Operation Mode 162

5.3 Sleep Function.....164
5.3.1 Related Registers.....164
5.3.2 Standby Operation Status.....165
5.3.3 HALT Mode.....165

5.4 Hardware Reset Function.....167
5.4.1 External Reset Pin Function.....167
5.4.2 Hardware Status During a Reset.....167

Part 3
System BIOS 171

Chapter 6
Programs in ROM 173

6.1 System Programs....173

6.2 OS Programs 174

6.3 Headers 174

Chapter 7
Memory Space 175

Chapter 8
System BIOS Functions 177

Chapter 9
Subroutine Call Procedure 179

9.1 Processing Contents of Labels....180

9.2 Interaction Between System BIOS and Application 181

Contents

Chapter 10
Application Shutdown Procedure When MODE Button is Pressed 183

10.1 Processing Contents of Labels....184

10.2 Interaction Between System BIOS and Application 185

Chapter 11
VMS Initialization 187

Subroutine Reference 189

12.1 Flash Memory Access Functions 189

12.2 Subroutine Use Precautions 189

Flash memory routines 191

fm_prd_ex(ORG 0120H) Flash memory page data read 191

fm_wrt_ex(ORG 0100H) Flash memory data write 193

fm_vrf_ex(ORG 0110H) Flash memory page data verify....195

Clock Function....197

timer_ex Clock count-up timer 197

Chapter 13
Low Battery Voltage Auto Detection 199

Chapter 14
List of Defined Variables 201

Part 4
PWM Sound Source 203

Chapter 15
Sound Output Method 205

15.1 Timer 1 Outline 205
15.1.1 Timer 1 Block Configuration 205
15.1.2 Related Registers 206
15.1.3 Mode Setting 207

15.2 8 Bit Counter Mode 208
15.2.1 Output Waveform and Parameter Setting 208
15.2.2 8 Bit Counter Mode Setting 208
15.2.3 Frequency Characteristics 210
15.2.4 Output Frequency Table 210

Chapter 16
Sample Program 213

Contents

Part 5
Appendix 215

Chapter 17
Variable Bit Length Pulse Generator 217

Chapter 18
Symbol Table 221

Chapter 19
VMS Mode Selection 223

Chapter 20
Calculation of Battery Life 225

20.1 Methods for Enhancing Battery Life 225

20.2 Oscillator Circuit and Current Consumption 226

20.3 Oscillation Control Register 226
20.3.1 System Clock Division Ratio Setting 226
20.3.2 Oscillator Circuit Selection 227
20.3.3 Oscillator Circuit Start/Stop 227

20.4 Calculating Battery Life 227
20.4.1 Calculating Continuous Operating Time 228
20.4.2 Calculating Battery Life in Days 229

Chapter 21
Serial Communication Precautions 231

21.1 Serial Communication Timing Chart 231

21.2 Measures to Ensure Problem-Free Serial Transfer 232

21.2.1 Mask All Interrupts 232

21.2.2 Set Maximum Send Wait Time 233

PART 1

VMS

Overview

This section contains a general overview of the Visual Memory specifications. It should be read by programmers, game planners and designers, as well as graphics developers.

Chapter 1 Visual Memory System Overview

The Visual Memory System is a memory cartridge that serves not only for storing data but also for visually displaying information on an integrated LCD. It is connected to the Dreamcast controller (hereafter referred to simply as the "controller") and is used as a memory card that can store game data and display secondary screens during a game. It can be connected and removed also while the Dreamcast is turned ON.

In the standalone condition (while not connected to the controller), it is possible to display a directory of data files and to perform housekeeping (deleting files). Two VMS units can be connected for operations such as copying files.

By downloading an application from the Dreamcast to the VMS, the VMS can be used as a miniature game machine. Connecting two units for two-player battle-type games is also possible.

Fig. 1-1 VMS allows two-player battle-type games

1.1 VMS Specifications

The VMS hardware configuration is shown below.

Table 1-1 VMS Specifications

VMS custom chip (Sanyo LC8670)	CPU	8 bit	Instruction cycle time Connected to Dreamcast: 6 MHz (1 μs) Standalone operation: 32 kHz (183 μs) Note: In standalone operation, operation is deliberately slowed down to reduce power consumption.
	Memory	ROM	16 KB system BIOS, system programs
		Flash memory EEPROM	64 KB program/data area 64 KB data area (28 KB reserved for system)
		RAM	256 bytes for applications 256 bytes reserved for system
		Work RAM	512 bytes work RAM. When connected to Dreamcast, reserved by system for use as transfer buffer. In standalone operation, read/write in single byte units possible.
		XRAM (for LCD)	96 bytes for LCD upper half 96 bytes for LCD lower half 6 bytes for icons (reserved by system)
	Serial interface	Used exclusively as follows. (1) Dedicated Dreamcast interface (2) Synchronous 8-but serial interface with 2 transfer channels	
	Timer	16-bit clock timer 16-bit (or 8-bit × 2), use as PWM sound source possible	
	I/O ports	Input/output 16 lines (buttons, serial interface) 4 lines (control connector)	
	LCD controller	33 common lines, 48 segment lines	
LCD	Reflective type liquid crystal	48 (horizontal) × 32 (vertical) dots, 2-value B/W 4 mode icons (file, game, clock, alert), reserved by system	
	Piezoelectric buzzer	Alarm (PWM sound source output)	
Power supply	Button type batter External input External output	CR2032 × 2 +5V, +3.3V input 3.3V output	
Buttons	8 operation buttons + reset button	Direction buttons, A button, B button, MODE button, SLEEP button (reset button)	
Connector	14 pins	Serial interface, power supply	

Fig. 1-2 External view

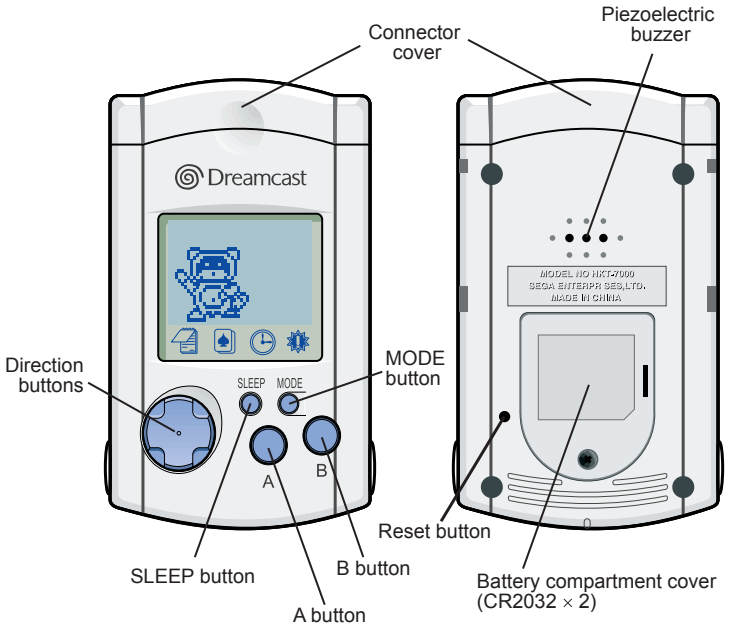


Fig. 1-3 VMS Front View and Rear View

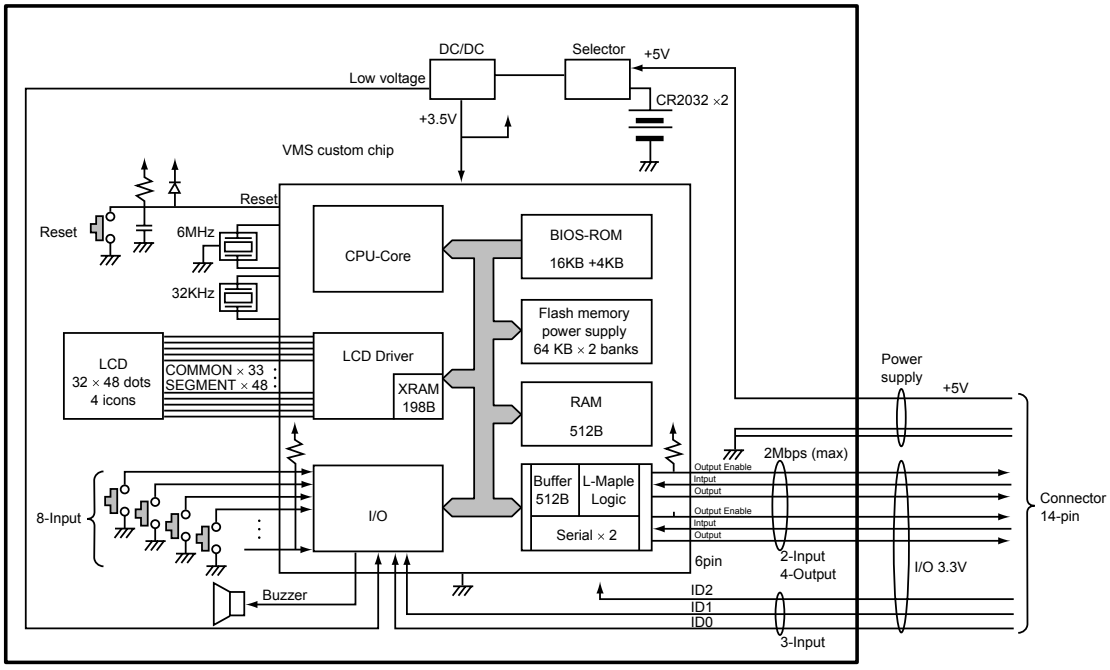


Fig. 1-4 System Block Diagram

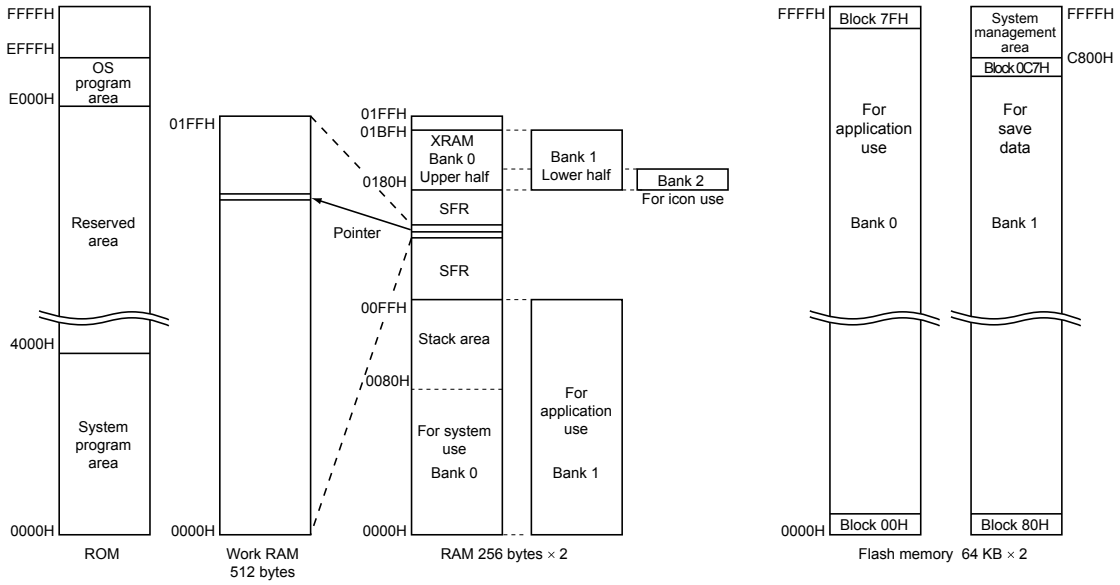


Fig. 1-5 VMS Memory Map

1.2 VMS Functions

When connected to the Dreamcast via its dedicated interface, the following functions of the VMS are controlled by the Dreamcast.

Note	The control port of the Dreamcast accepts the standard controller and other devices such as a steering controller etc. Devices which have an expansion device connector can accommodate the VMS or other add-on devices. These peripherals communicate with the Dreamcast via a dedicated bus called the Maple bus.

1. Game data storage medium
2. Controller-specific LCD display
3. VMS clock read and set

These functions are controlled using special programs stored in ROM on the VMS. These programs are collectively referred to as the system BIOS.

The system BIOS consists of system programs, OS programs, and headers. The system programs perform functions such as copying and deleting files, controlling the clock display, and communicating with the Dreamcast. OS programs control basic functions such as flash memory read/write, internal clock setting, battery voltage checking, etc. Some OS programs can be called by applications. For this purpose, a part of the program must be placed in a dedicated location in the flash memory. These parts are called headers.

The VMS contains the following system programs.

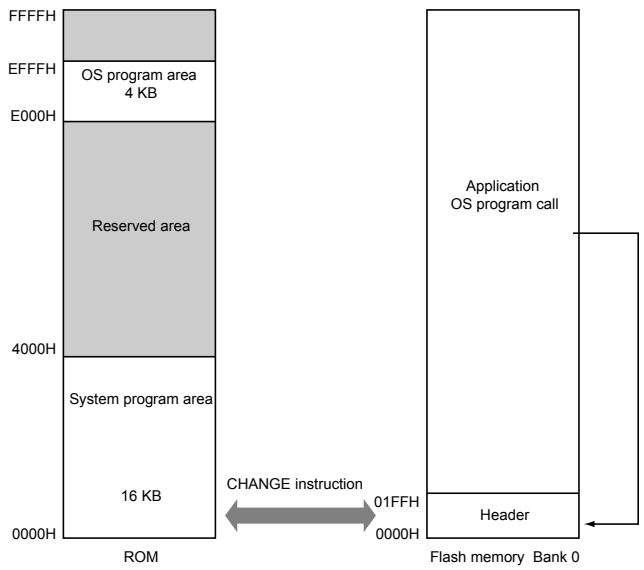


Fig. 1-6 Memory Map of Programs in ROM

File management

File management refers to the handling of game data stored in the Dreamcast and executable files for VMS applications.

Files are managed in units of one block (128 bytes). Reading and writing in block units is possible.

All FAT operations, file name information etc. are handled by system programs.

Liquid-Crystal Display

When connected to the Dreamcast, the display of the VMS shows only graphics transferred as image data from the Dreamcast. System programs receive the data and handle them for display on the LCD.

In standalone mode, the CPU of the VMS directly controls graphics display. The dot matrix section of the LCD uses a grid of 32 (vertical) × 48 (horizontal) dots. The data amount for one image is 192 bytes. In addition, the LCD also contains four types of icons to indicate operation modes.

Icon	Operation mode	Function
	File mode	VMS file management
	Game mode	Executing game from flash memory
	Clock mode	Date and time display

Because these icons show the operation mode of the VMS, their status may not be changed by applications.

Starting VMS applications

A VMS application can be transferred from the Dreamcast and started by the VMS. OS programs also comprise various subroutines that can be used by applications. For details, refer to the “System BIOS” section.

Data transfer

When connected to the Dreamcast, control of the VMS is performed via a dedicated interface.

In standalone operation, an 8-bit synchronous serial interface is available for communication with another VMS unit.

Clock

The VMS incorporates a clock which operates at all times, whether connected to the Dreamcast, running an application, or in sleep mode.

Application programs can obtain date and time information using an OS program.

Buzzer

The piezoelectric buzzer incorporated in the VMS is driven by a pulse generator (PWM) allowing for variable frequencies. In theory, the available frequency range is 21 Hz to 5.5 kHz, with 170 Hz to 2.7 kHz being recommended.

While connected to the Dreamcast, control of the buzzer from the Dreamcast is possible.

During standalone operation, the frequency can be changed by controlling the PWM, and buzzer on/off control is also possible.

Operation mode switching

The VMS operation mode is determined by the connection method to the Dreamcast and by the MODE button. The current operation mode is indicated by an icon on the LCD.

In standalone operation other than game mode, if no button was pressed or no communication has occurred for more than two minutes, the auto power-off function sets the unit to sleep mode to conserve power.

The VMS has the following operation modes:

- File mode (management of stored game data)
- Game mode (playing a VMS internal game)
- Clock mode (clock display and setting)
- System mode (flash memory access)

Table 1-2 Dreamcast Connection Status and Operation Mode

Connection Status	MODE button Icon display	Status	Operation mode
Connected to Dreamcast	Game File Clock	Always out	System mode
	Alert	On	Flash memory access
Standalone operation	Game	On	Application running
	File	On	File operation
	Clock	On	Clock display
	Alert	On	Flash memory access

Integrated character font

VMS incorporates an ANK font using a grid of 8 (vertical) × 6 (horizontal) dots. The font comprises alphanumeric characters, Japanese katakana, and symbols. The font is only for internal use by the system. It cannot be used by applications, either when connected to the Dreamcast or in standalone mode.

To display characters, image data must be placed in XRAM.

1.3 Mode Setting

The VMS operation mode is determined by the connection method to the Dreamcast and by the MODE button + A button.

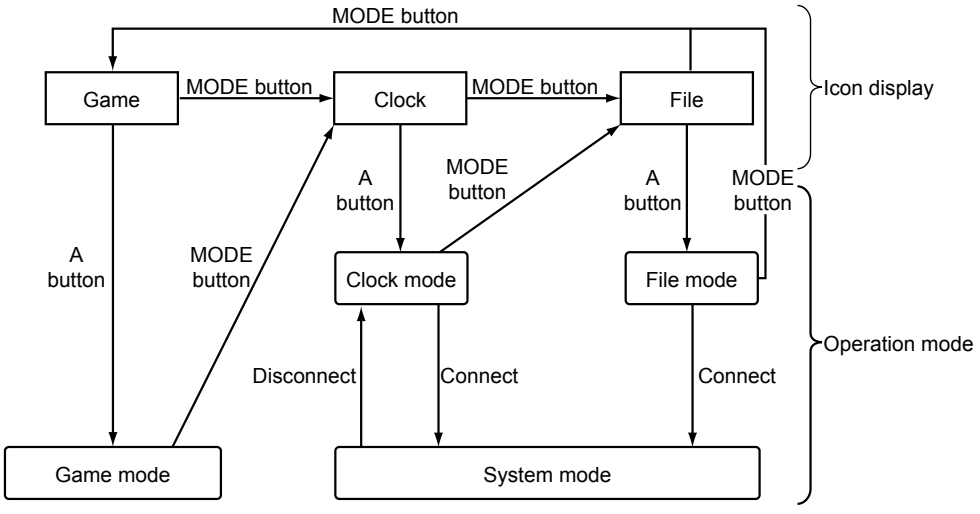


Fig. 1-7 Mode Transition

Details of the various operation modes are as follows.

System mode

VMS is controlled by external control program (Dreamcast). VMS carries out Maple bus compliant communications and memory management, LCD display, and clock management.

When leaving system mode, such as when the VMS is disconnected from the Dreamcast controller, the VMS title screen is shown on the LCD.

Game mode

In this mode, an application read into the flash memory is executed.

Caution	Applications should be designed to always check for a MODE button press. When the button is depressed, the application must terminate immediately and control must be handed to the system program. This applies also when the VMS is connected to the Dreamcast controller while an application is running. When a work area in RAM is used, the application should move its contents to flash memory or similar before terminating. For information on how to terminate applications and hand control to the system program, refer to Chapter 10 "Application Shutdown When MODE Button Is Pressed".

File mode

This mode serves for managing game data stored on the VMS. File management is performed by system programs. The buttons on the VMS are used to display, copy, or delete files written to the flash memory.

Clock mode

In this mode, the time is displayed on the LCD of the VMS. Time can be displayed using hours, minutes, and seconds, and the user can set the time as desired. When connected to the Dreamcast, time setting can also be performed from the Dreamcast side.

Clock functions are performed using system programs and OS programs.

1.4 File Management

The total capacity of the flash memory on the VMS is 128 KB (64 KB × 2 banks). 28 KB are reserved for the system.

Flash memory is managed by the system using 128-byte blocks. The smallest read/write unit for a file therefore is 1 block (= 128 bytes), and up to 200 blocks of data can be stored.

One executable application file can be transferred to the VMS. The executable file must be placed be in contiguous blocks starting at block 00H. The maximum size for the executable file is 64 KB (= 128 blocks).

It is not possible to transfer and execute multiple executable files or an executable file larger than 64 KB.

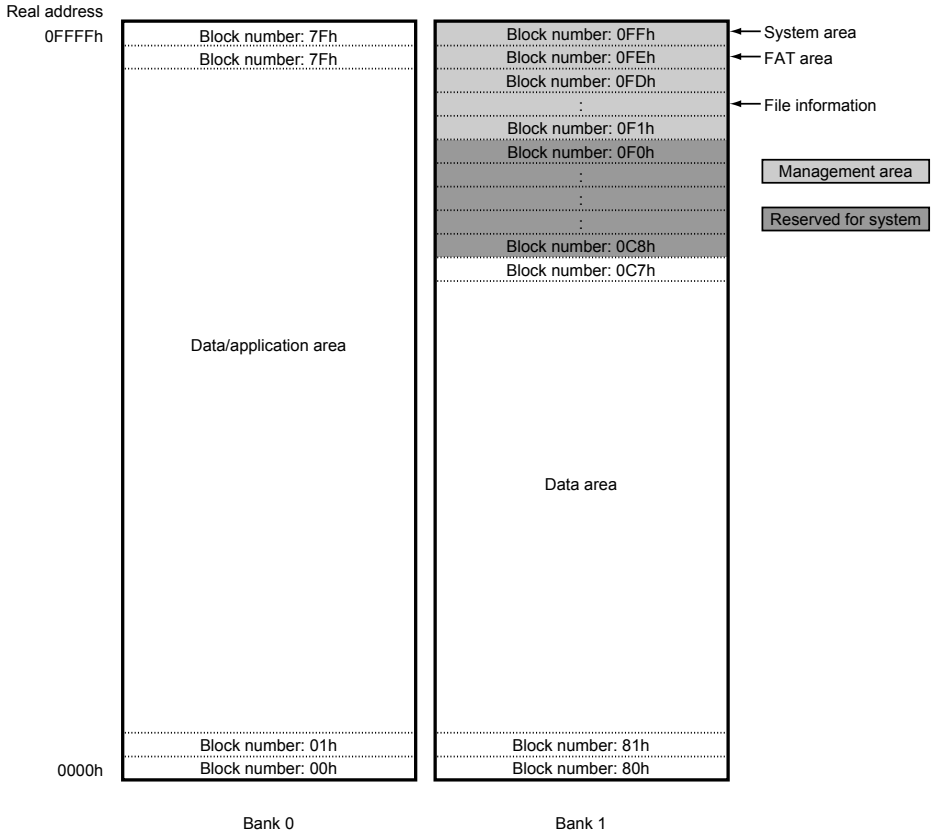


Fig. 1-8 Flash Memory Memory Map

Flash memory management area

15 blocks starting from the top of the memory range (block 0FFH) are used as memory management area. The management area is divided into the system area (1 block), FAT area (1 block), and file information area (13 blocks).

The system area is write-protected except for VMS formatting performed by the Dreamcast.

The FAT area manages one block using 2 bytes (16 bits), to maintain the block chain configuration.

The file information area holds 32 bytes of information per file and can manage up to 200 files. Out of the 32 bytes, 12 bytes (equivalent to 12 ASCII codes) are used for the file name. Because a hierarchical structure is not supported, subdirectories cannot be created.

Data area

The data area which can hold files consists of 200 blocks extending from block 00H to block 0C7H. Files are placed in this area starting from block 0C7H and going towards block 00H. The application starts from block 00H.

Blocks 00H to 7FH and 80H to 0FFH are managed by bank switching, performed automatically by an OS program.

For reading and writing to the flash memory, always call the OS program.

Reserved area

This area is used by system programs and system modes. Writing to this area is prohibited.

1.5 LCD Display

The LCD of the VMS consists of a dot matrix section with 32 (vertical) × 48 (horizontal) dots and an operation mode icon section with 4 icons.

To display images on the LCD, the image data must be stored in the dedicated XRAM.

XRAM

The dedicated RAM used for LCD display is called XRAM. This corresponds to the video RAM in a conventional computer.

The XRAM has 3 banks. Banks 0 and 1 can be written to by applications. Bank 2 serves for operation mode display and cannot be used by applications.

Bank 0 of the XRAM corresponds to the upper half of the LCD (48 × 16 dots), and bank 1 to the lower half (48 × 16 dots).

1 LCD dot corresponds to 1 bit in the XRAM. 1 byte of XRAM controls 8 horizontal dots, with 6 bytes forming one horizontal line.

Image mode

When connected to the Dreamcast, image data received from the Dreamcast are normally written to the XRAM by a system program. However, for display of a secondary game screen, image data are written directly to the VMS. When transferring image data, pay attention to the top/bottom orientation of the VMS. Vertical image reversal can be performed using the Ninja library.

The VMS also incorporates an ANK character font, but this is for exclusive use by system programs. It cannot be used by applications.

To draw an image on the LCD, XRAM bits for black dots should be set.

Icon

Because the icons show the operation mode of the VMS, their status may not be changed by applications.

Image configuration

The LCD of the VMS is configured as follows.

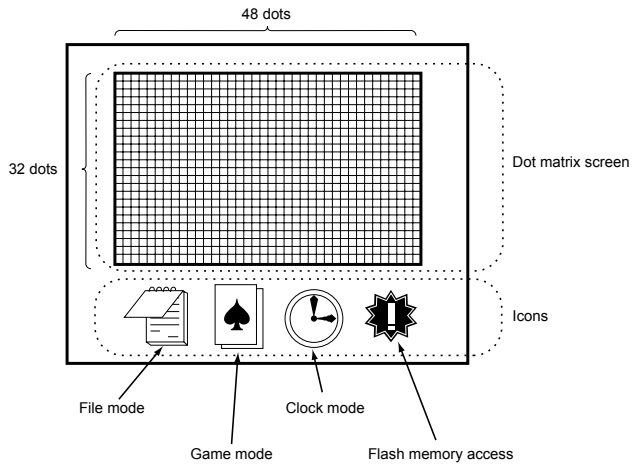


Fig. 1-9 LCD Screen

LCD characteristics

The screen refresh principle for an LCD differs from that for a CRT display. After data have been transferred to the XRAM, they are displayed immediately on the LCD, but there is a certain delay due to the response characteristics of the LCD. If this delay is not handled properly, trailing images and flicker will severely impair display quality.

The clock differs in standalone operation and when connected to the Dreamcast. In standalone mode, LCD display speed is slower.

Recommended refresh rate for the LCD of the VMS is 200 μs or more.

Other important points

Also consider the points listed below when developing applications.

- There is no provision for contrast adjustment (only LCD on/off control)
- There is no provision for brightness adjustment.
- There is no backlight.
- The reflective polarizer plate (rear sheet) of the LCD cannot have a pattern (picture or similar).

1.6 Starting an Executable File

An application can be transferred from the Dreamcast or a conventional computer to the VMS, for execution in standalone mode.

Only one executable file can be transferred to one VMS. It is not possible to use multiple applications simultaneously.

Several OS programs are being made available for use by applications.

Writing applications for the VMS

Applications for the VMS should be written using an MS-DOS assembler and linker. The conventional executable file created by the linker is converted into an executable file for the VMS by the program E2H86K.EXE.

A VMS application can be debugged using the VMS simulator designed to run under Windows 95 and later. This simulator emulates all aspects of VMS hardware operation in software. For details, refer to the VMS Simulator Guide.

Transferring an executable file

The executable file is to be stored in blocks 00H to 7fH of the flash memory, starting at block 00H.

Before sending an executable file from the Dreamcast or a conventional computer to the VMS, a contiguous area starting at block 00H must be obtained (defragmented). If the amount of available memory is smaller than the application or if no contiguous area can be obtained, the application cannot be transferred.

For transfer, use the Ninja library and transfer utilities. These allow automatic checking of available space and defragmentation.

Executable file size

The maximum executable file size is 64K. Larger applications cannot be transferred to the VMS. When an area for storing data in the flash memory is required, this area must be provided for within the application.

Caution	The executable file comprises the OS program and a program header area containing interrupt vector information. In GHEAD.ASM supplied with the SDK, the program header area is 0000H - 01FFH.

OS programs usable by applications

The following OS programs can be used by applications. When an OS program is called, a part of RAM can be used as work area.

1. Automatic low-battery check	Allows enabling an automatic low battery warning.
2. Clock read	Gets the date and time from the internal clock in the VMS.
3. Flash memory write	Writes data to flash memory in block units.
4. Flash memory read	Reads data from flash memory in block units.
5. Flash memory verify	Checks data read from flash memory for validity.

RAM

The following RAM areas are available to applications.

```
RAM         00H to 0FFH (bank 1)
            RAM bank 0 is reserved for the system.  Except for the
            stack area, it cannot be used by applications.
Work RAM    00H to 1FFH (read in 1-byte units by specifying address)
XRAM        Bank 0, bank 1
```

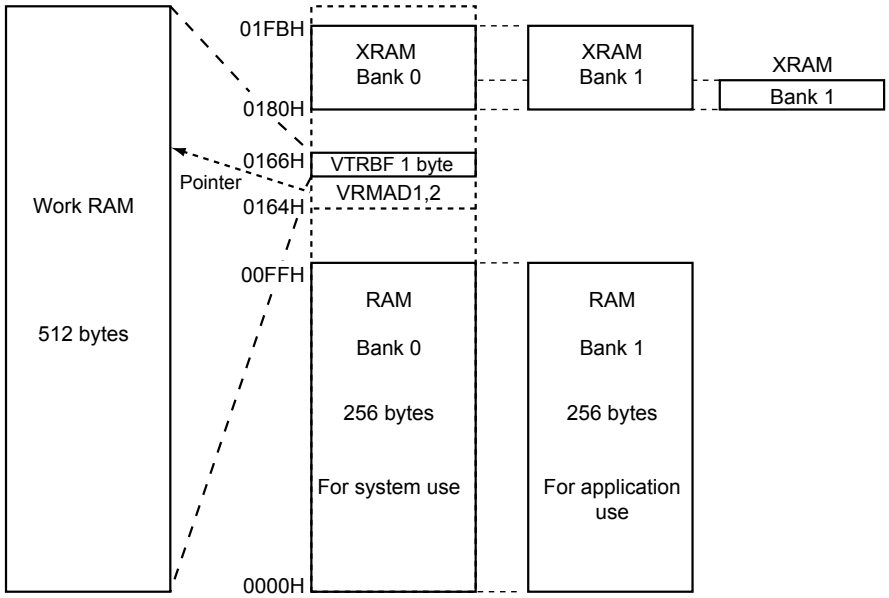


Fig. 1-10 RAM Memory Map

Saving application data

If a VMS application needs to save progress data or parameters, a data area must be provided within the executable file. Because the executable file is read into flash memory, the data area also will be stored in flash memory. It is not possible to create files of the same format as for Dreamcast save data.

When data saved in a VMS application are to be used as Dreamcast applications or links, read the entire VMS application and perform a lookup on the data addresses in it.

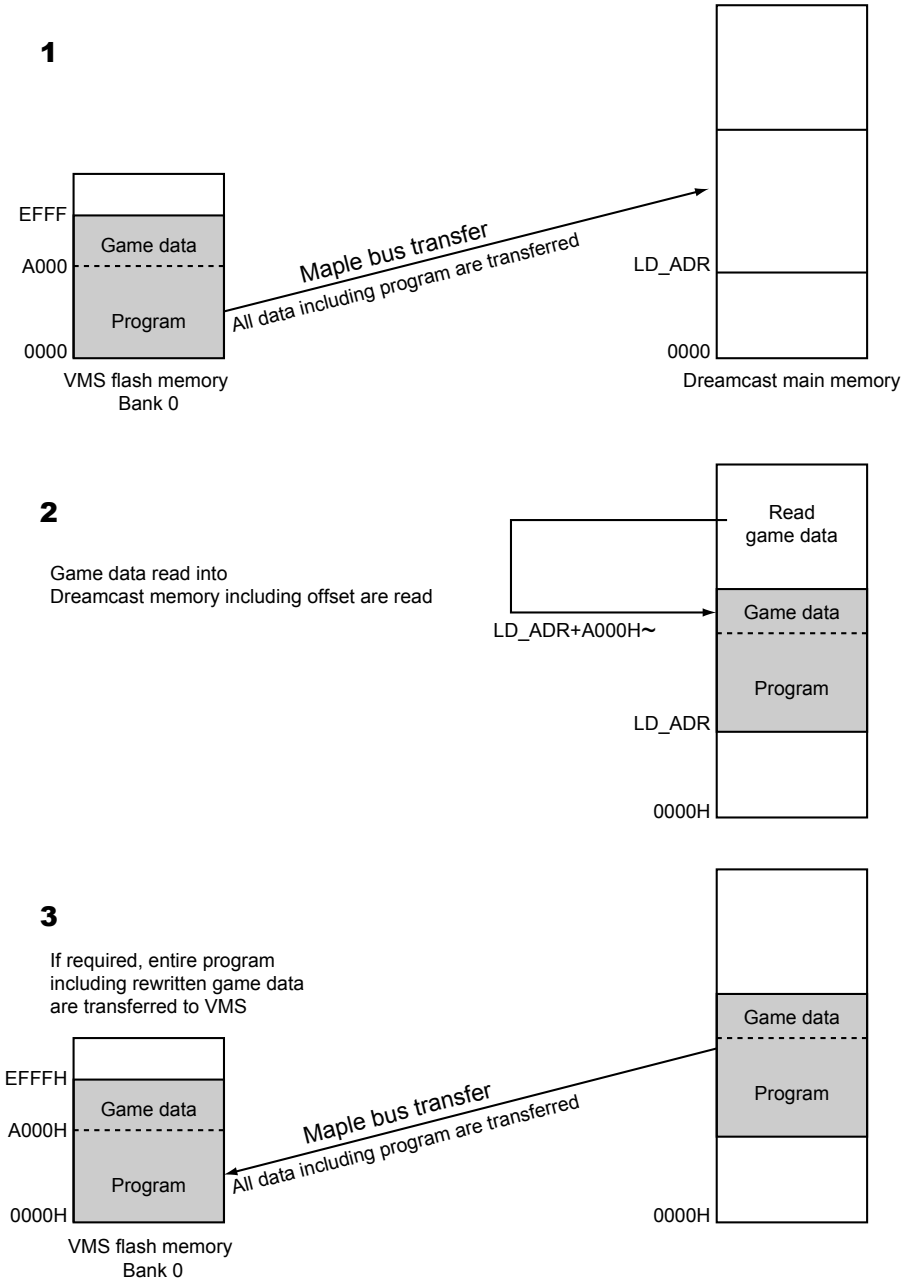


Fig. 1-11 Linking of Save Data in VMS and Dreamcast

Auto power-off

The VMS incorporates an auto power-off function that automatically sets the unit to sleep mode if no button was pressed or no communication has occurred for more than two minutes. For details on the sleep mode of the VMS, refer to section “1.10 Sleep Mode”.

When game mode is active, auto power-off is disabled. Applications must provide their own sleep mode. For details, refer to section “5.3 Sleep Mode”.

1.7 Communication Functions

VMS can communicate with other devices via a serial interface. Two protocols are available. When connected to the Dreamcast, the Maple bus protocol is used. In standalone operation, full-duplex synchronous serial transfer is used. Protocol switching is performed automatically by a system program detecting the Dreamcast connection status.

Maple bus protocol

When connected to the Dreamcast, the communication connector of the VMS becomes a 2 Mbps Maple bus connector. The entire work RAM is used as send/receive buffer. If an application was using the work RAM as work area, the entire contents will be destroyed. Take this into account when designing applications.

The Maple bus cannot be used by applications.

Synchronous serial transfer

When the VMS is operating in standalone mode and data transfer is carried out between two VMS units or between one VMS and a computer, synchronous serial transfer is used. There are two serial communication lines, allowing full-duplex operation. Data can be transferred in 1 byte units, and the maximum transfer rate is about 2.4 kbps.

1.8 Clock Function

A 32.768 kHz quartz oscillator and dedicated counter keep time in 500-millisecond units. Date and time data are managed by an OS program. These data can be read by an application, but not written to.

A setting screen is used to set the year, month, day, and time. When connected to the Dreamcast, the clock in the VMS can be set from the Dreamcast.

1.9 Alarm Function

The piezoelectric buzzer incorporated in the VMS can be used for an alarm. The buzzer can emit a single tone at a time. In theory, the available frequency range is 21 Hz to 5.5 kHz, with 170 Hz to 2.7 kHz being recommended. The alarm function can be implemented by setting the timer (pulse generator) connected to the buzzer. The buzzer can be switched on and off, but volume control is not possible.

When connected to the Dreamcast, the Dreamcast can control the buzzer of the VMS, including the frequency setting.

1.10 Sleep Function

The VMS incorporates a sleep function designed to conserve power when operating in standalone mode. In the sleep condition, the state of the I/O ports and the contents of RAM are maintained, but the CPU and LCD are turned off.

In game mode, transition to the sleep state is controlled by the application. In clock mode and file mode, the following conditions cause transition to sleep mode.

- SLEEP button was pressed
- Auto power-off function was activated
- No button press or communication for about 2 minutes

To cancel the sleep mode, the SLEEP button must be pressed. Other buttons are disregarded.

The RAM and register memory contents are preserved, except for the clock register.

1.11 Buttons

The VMS has the following buttons.

Applications should be designed so as to maintain the interface described below.

Button name	Main function
Direction button (up)	Cursor movement and display scrolling
Direction button (down)	Cursor movement and display scrolling
Direction button (left)	Cursor movement and display scrolling
Direction button (right)	Cursor movement and display scrolling
A button	Mainly “confirm”
B button	Mainly “cancel”
MODE button	Mode switching during standalone operation Each push cycles through “File” → “Game” → “Clock” → “File” ...
SLEEP button	Activating and canceling of sleep mode in standalone operation
Reset button	Reset of unit contents except flash memory and clock

1.12 Batteries

The VMS incorporates two button-size batteries (CR2032) which act as a power source in standalone operation. While connected to the Dreamcast, power is supplied by the Dreamcast.

Battery life will depend on the usage conditions of applications. Under the conditions outlined below, the batteries will last about two weeks.

- VMS standalone operation
- LCD display on (refresh rate 1 kHz)
- No alarm output
- Communication functions not used
- No write to flash memory

For specific information on how to calculate expected battery life for an application, refer to section 20 “Calculating Battery Life”.

Battery life

Battery life depends on the operation condition of the VMS. Refer to the table below to calculate battery life for applications.

Operation	Battery power consumption	Comments
Program running	Standard	Reference for flash memory read and CPU battery power consumption
LCD screen update	Standard ×5	Frequent XRAM rewriting (screen update) consumes battery power
Alarm output	Slightly more than standard	Slight increase in battery power consumption
Flash memory write	Standard ×25	Flash memory writes should be limited to minimum because of extremely high power consumption
Data transfer		Very high battery consumption, especially by applications which also write to flash memory. When transferring large files, take battery life into consideration.

Battery status monitoring

An OS program continuously monitors the battery voltage. When the batteries near the end of their service life, the program will trigger auto power-off, even if an application is running.

Battery replacement

When the batteries are replaced, the clock will be initialized, but the contents of flash memory are not affected.

PART 2

Hardware Manual

.

This section contains detailed specifications for the VMS. The information in this section is intended for program designers and developers. Special terms used in this section are explained below.

Term	Explanation
ACC (A_	Accumulator
B	B register
C	C register
CY	Carry flag (PSW bit 7)
AC	Auxiliary carry flag (PSW bit 6)
OV	Overflow flag (PSW bit 2)
PC	Program counter
ROM	Program memory
RAM	RAM
SFR	Special function register
PSW	Program status word
TRL	Table reference register lower byte
TRH	Table reference register upper byte
TRR	Table reference register (16 bit expressed as TRH, TRL)
SP	Stack pointer
Pn	Port n
Pmn	Port m, bit n
Rj	Indirect address register for current bank (part of internal RAM)
D9	Direct addressing data
#i8	Immediate data
B3	Bit addressing data
R8	Relative addressing data [signed 8-bit: -128 to +127]
R16	Relative addressing data [unsigned 16-bit: 0 to +65535]
A12	Relative addressing data
A16	Relative addressing data
←	Transfer direction
∧	Logical product
∨	Logical AND
∀	Exclusive logical AND
Mod	Residual
Bit	Binary digit (0 or 1)
Clear	Set all to “0”
Carry	Carry over to next higher digit
Borrow	Borrow from next higher digit
Cycle clock	1 cycle signal at instruction execution
Bytes	8-bit unit (1 byte); indicates the unit handled by an instruction
Cycles	Indicates in how many basic cycles the instruction executes
??H	Hexadecimal number
????????B	Binary number
MSB	Most significant bit (bit 7)
LSB	Least significant bit (bit 0)
R	Read enable (used in tables)
W	Write enable (used in tables)
X	Undetermined (used in tables)
H	Not used ("1" when read) (used in tables)

PART 2

Hardware Manual

.

This section contains detailed specifications for the VMS. The information in this section is intended for program designers and developers. Special terms used in this section are explained below.

Term	Explanation
ACC (A_	Accumulator
B	B register
C	C register
CY	Carry flag (PSW bit 7)
AC	Auxiliary carry flag (PSW bit 6)
OV	Overflow flag (PSW bit 2)
PC	Program counter
ROM	Program memory
RAM	RAM
SFR	Special function register
PSW	Program status word
TRL	Table reference register lower byte
TRH	Table reference register upper byte
TRR	Table reference register (16 bit expressed as TRH, TRL)
SP	Stack pointer
Pn	Port n
Pmn	Port m, bit n
Rj	Indirect address register for current bank (part of internal RAM)
D9	Direct addressing data
#i8	Immediate data
B3	Bit addressing data
R8	Relative addressing data [signed 8-bit: -128 to +127]
R16	Relative addressing data [unsigned 16-bit: 0 to +65535]
A12	Relative addressing data
A16	Relative addressing data
←	Transfer direction
∧	Logical product
∨	Logical AND
∀	Exclusive logical AND
Mod	Residual
Bit	Binary digit (0 or 1)
Clear	Set all to “0”
Carry	Carry over to next higher digit
Borrow	Borrow from next higher digit
Cycle clock	1 cycle signal at instruction execution
Bytes	8-bit unit (1 byte); indicates the unit handled by an instruction
Cycles	Indicates in how many basic cycles the instruction executes
??H	Hexadecimal number
????????B	Binary number
MSB	Most significant bit (bit 7)
LSB	Least significant bit (bit 0)
R	Read enable (used in tables)
W	Write enable (used in tables)
X	Undetermined (used in tables)
H	Not used ("1" when read) (used in tables)

Chapter 2 CPU Features

The VMS is a memory system for the Dreamcast game machine. The CPU in the custom LSI chip has a minimum cycle time of 0.5 µs. Other functions integrated on this chip are a 128- KB flash memory, 20-KB ROM, 710-byte RAM, LCD controller/driver, 16-bit timer/counter/pulse generator, 16-bit (or 2-channel × 8-bit) timer, 2-channel × 8-bit synchronous serial interface, dedicated Dreamcast interface, and 13-source, 10-vector interrupt architecture.

2.1 Differences to Conventional CPUs

Normally, a CPU will have an internal accumulator as well as general registers and flag registers. The control registers and data registers for the serial port and other peripheral devices are mapped onto the I/O ports.

In the VMS custom chip, all CPU and peripheral device registers are mapped onto memory. These registers are referred to as "special function registers" (SFR) and are treated separately from RAM.

Keep in mind that these "special function registers" are not internal registers of the CPU.

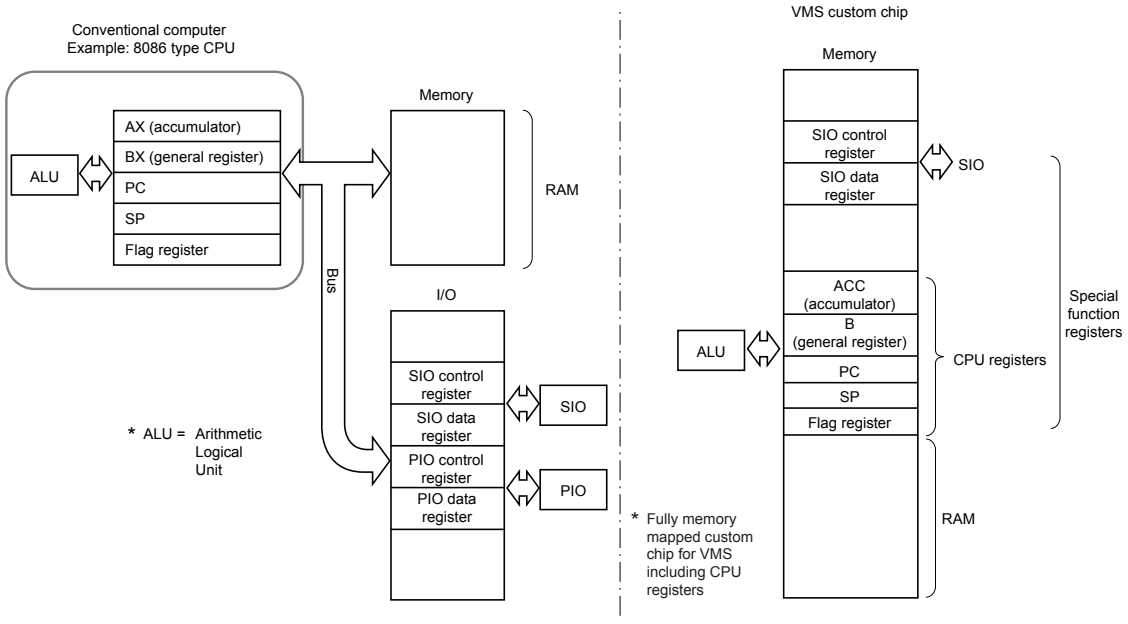


Fig. 2-1 Differences to Conventional CPU

2.2 Specifications

This section gives an overview of VMS specifications.

Memory specifications

Flash memory

65536 bytes: Program/data area

65536 bytes: Data area

ROM

16384 bytes: Program area

4096 bytes: System BIOS program area

RAM

Arithmetic area: 256 bytes × 2 banks

Display area: 198 bytes (LCD video XRAM)

Work area: 256 bytes × 2 banks (work RAM)

Note	The work RAM in the work area is used as a send/receive buffer when connected to the Dreamcast.

Bus cycle time and instruction cycle time

The bus cycle time refers to the ROM read time.

Bus cycle time	Instruction cycle time	System clock oscillator	Oscillation frequency	Power supply voltage	Others
3.412µs	6.824µs	RC oscillator	879.236kHz	3.15 to 3.8V	OCR7=1*[1]
91.553µs	183.105µs	Quartz oscillator	32.768kHz	3.15 to 3.8V	OCR7=1*[1]

Caution	OCR7 (bit 7 of the oscillation control register OCR) controls the system clock generator operation and cycle time. For details, refer to section 5.2 "System Clock Generator". OCR7 = 1: 1/6 of system clock is used as cycle time The frequency of the RC oscillator circuit is subject to tolerances. The reference value is 879.236 kHz, but the frequency can range from about 600 kHz to 1200 kHz.

Ports

I/O ports: 2 (P1, P3)

Input port: 1 (P7)

LCD segment drive output ports: 48

LCD drive common output ports: 33

LCD controller

Display duty cycle: 1/33

Display bias: 1/5

LCD instruction: on/off

Graphics display: 32 vertical x 48 horizontal dots + 4 icons

Serial interface

8-bit serial interface × 2 channels (synchronous)

Integrated 8-bit baud rate generator (also used for 2- channel serial interface)

Dedicated Dreamcast interface (automatic start patter/end pattern detection)

Caution	Synchronous serial interface and dedicated Dreamcast interface cannot be used simultaneously.

Timer

Timer 0

16-bit timer/counter

with 8-bit programmable prescaler

Timer 1

16-bit timer/pulse generator

Base timer: clock selector function

Selects between 32.768 kHz quartz oscillator, system clock, timer 0 programmable prescaler output

500-ms overflow signal generator for clock (when 32.768 kHz quartz oscillator is selected)

Overflow signal generator for 976 μs, 3.9 ms, 15.6 ms, or 62.5 ms cycle (when 32.768 kHz quartz oscillator is selected)

Interrupts

The interrupt architecture comprises 13 sources and 10 vectors

1. External interrupt INT0: connection detection for dedicated Dreamcast interface
2. External interrupt INT1: low power supply voltage interrupt
3. External interrupt INT2: timer/counter T0L (timer 0, lower 8 bits)
4. External interrupt INT3: base timer
5. Timer/counter T0H (timer 0, upper 8 bits)
6. Timer T1L (lower 8 bits), timer T1H (upper 8 bits)
7. Serial interface 0 (SIO0)
8. Serial interface 1 (SIO1)
9. Dedicated Dreamcast interface
10. Port 3

Caution	The clock function of the VMS is implemented by counting the interrupts generated in 0.5 second intervals by the base timer. The port 3 interrupt is a level interrupt which is maintained for as long as the user presses a button. If the timer is used to frequently generate interrupts or to accept the port 3 level interrupt, the internal clock may run slow. When using the base timer interrupt, call the user-side handler immediately after the label timer_ex_exit in GHEAD.ASM. The user-side handler must be designed to keep processing time at a minimum, so that the interrupt can be properly processed every 0.5 seconds. Care must be taken to prevent clock slow-down already when designing an application.

Priority can be assigned to the interrupts using three interrupt levels (low, high, top). The interrupt priority register can be used to specify high or low priority for the 11 interrupt sources of port 3 for external interrupt INT2 and timer/counter T0L (timer 0, lower 8 bits). High or low priority can also be specified for external interrupt INT0 and INT1.

Stack area

128 bytes in RAM bank 0, from 80H to 0FFH. The internal clock uses 20 bytes. The stack is used up from the top (80H).

High-speed arithmetic instructions

16 bit × 8 bit (execution time: 7 command cycles)

16 bit ÷ 8 bit (execution time: 7 command cycles)

3 oscillator circuits

RC oscillator: system clock (reference: 879.236 kHz; tolerance range: 600 to 1200 kHz)

Quartz oscillator: clock, system clock, LCD driver clock (32.768 kHz)

Standby function

Sets CPU to HALT mode. In this mode, instructions are not executed, but the internal clock continues to operate. The mode can be canceled by a reset or interrupt.

The mode is identical to the sleep mode of the VMS, which can be canceled by pressing the SLEEP button.

Flash memory specifications

Memory type: EEPROM (Electrically Erasable Programmable ROM)

Capacity: 128 KB

Write method: using OS program

Write block size: 128 KB

Erase/write voltage: 3.15 to 3.8 V

Maximum number rewrite cycles: 50,000 (each cycle consisting of one FFH write and 00H write operation)

Ta = 25 °C, memory managed by program

Program memory space: 64 KB

System BIOS (ROM)/Application (flash memory) switching: by CHANGE instruction. At reset, BIOS is activated.

2.3 System block diagram

A block diagram of the VMS is shown below.

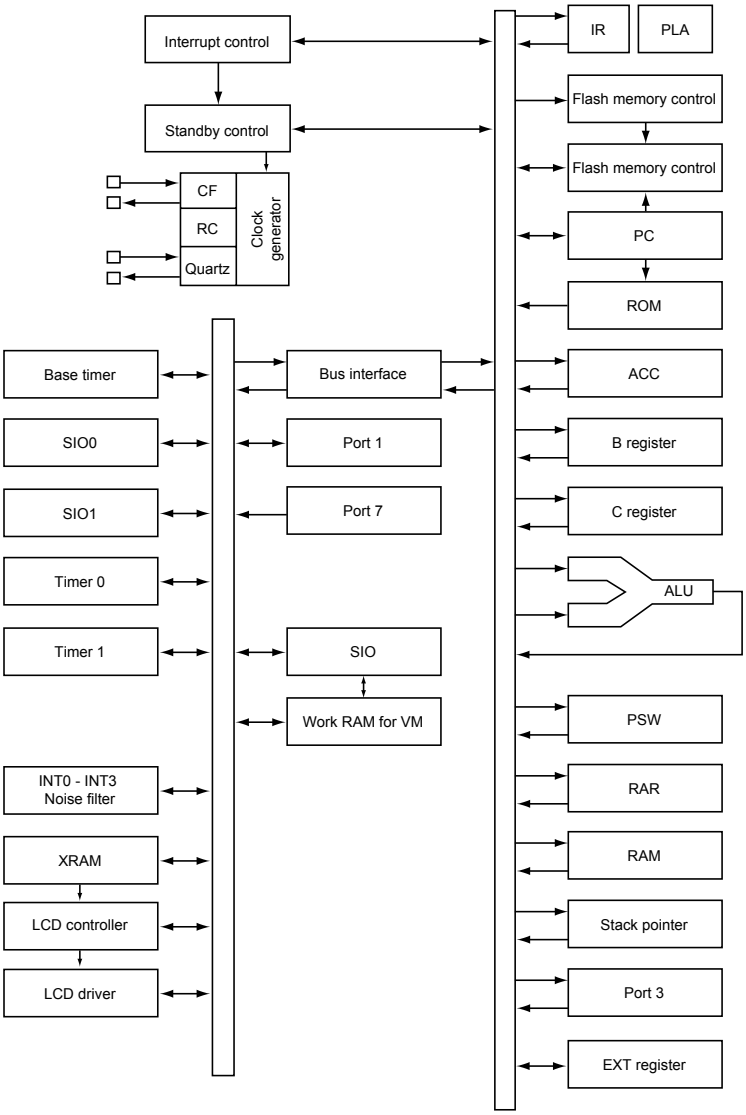


Fig. 2-2 VMS System Block Diagram

Chapter 3 Internal System Configuration

Unlike in a conventional CPU, the accumulator and all registers are mapped to RAM. The relationship between CPU functions and special function registers is described in this section.

3.1 Memory Space

The VMS custom chip comprises internal memory space and flash memory space. The internal memory space is divided into ROM (64 KB) and RAM (512 bytes). In ROM, sequential addresses are incremented with each normal instruction execution, allowing linear access to 64 KB.

In RAM, the 256 bytes formed by address range 000 to 0FFH are assigned as general-purpose RAM. The 256 bytes formed by address range 100 to 1FFH are assigned to the special function registers (SFR). General-purpose RAM consists of 2 banks. The bank can be specified by bit 1 (RAMBK0) of the program status word (PSW) of the special function registers (SFR). Bank 0 is also used as stack area. The SFR comprises accumulator (ACC), PSW, timer, I/O ports etc., forming a completely memory-mapped I/O configuration.

The flash memory space has a capacity of 128 KB, divided into 2 banks of 64 KB each. Bank 0 only is available for execution of application programs. Switching between the ROM system BIOS and a program in flash memory is performed by a dedicated macro instruction (CHANGE). Data writing to flash memory must be performed by calling the appropriate OS program.

Caution	When accessing the flash memory, inhibit all interrupts including the base timer. Because the base timer is used by the internal clock, the inhibit interval should be kept as short as possible. For writing to the flash memory, set the system clock to RC oscillator and the division ratio to 1/6. For write and verify, set the system clock to RC oscillator and the division ratio to 1/12.

OS program routines in ROM are provided for flash memory write, data verify, and read operations.

A VMS application always is stored in bank 0 of the flash memory.

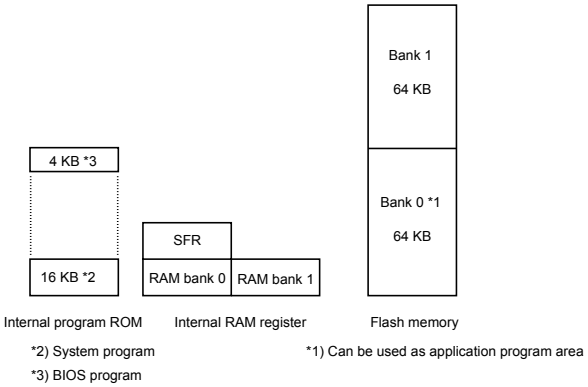


Fig. 3-1 Three Memory Space Types

3.2 Program Counter (PC)

The program counter (PC) uses a 16-bit configuration for storing the address of the program memory (ROM) where then next instruction to be executed is stored. The CPU refers to the PC value to execute a series of program instructions. The PC is normally incremented in steps of one instruction. When divider instructions and subroutines are executed and when interrupt or reset requests are processed, values for the respective operation states are set in the PC. These values are shown in the table below.

Table 3-1 Program Counter Setting Values

Operation		Program counter value
Reset		0000H (internal program space)
External interrupt 0		0003H
External interrupt 1		000BH
External interrupt 2, timer/counter T0L interrupt		0013H
External interrupt 3, base timer interrupt		001BH
Timer/counter T0H interrupt		0023H
Timer T1L, timer T1H interrupt		002BH
SIO0 interrupt		0033H
SIO1 interrupt		003BH
VMS SIO interrupt		0043H
Port 3 interrupt		004BH
Unconditional branch instruction	JMP a12	PC15 to PC12 = current page PC11 to 00=a12
	JMPF a16	C15 to 00=a16
	BR r16	(PC+2)+r8 [128 to +127]
	BRF r16	(PC+2)+r16 [0 to +65535]
Conditional branch instruction	BZ_ BNZ_ BP_ BNE BPC_ BN_ DBNZ_ BE	(PC+2 or +3) +r8 [-128 to +127]
CALL instruction	CALL a12	C15 to C12 = current page PC11 to 00=a12
	CALLF a12	C15 to 00=a16
	CALLR 16	(PC+2)+r16 [0 to +65535]
Macro instruction	CHANGE label name (or address)	Value specified by other program mode label or address

Caution	For convenience, 4 KB of ROM space are referred to as a page. The "current page" refers to the page which contains the instruction that is to be executed after the currently running instruction. When an interrupt is generated during ROM program execution, the interrupt vector in ROM (address in above table) is called. When an interrupt is generated while an application in flash memory is executing, the interrupt vector of bank 0 in flash memory (address in above table) is called. Applications cannot arbitrarily specify interrupt vectors. Rather, the specified program must be included in the application. For details, refer to section 5.1 "Interrupt Functions".

3.3 ROM Space

The 64 KB ROM space comprises 16 KB for system programs and 4 KB for OS programs.

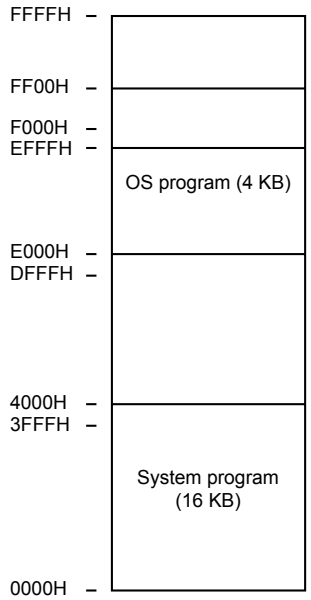


Fig. 3-2 ROM Space

3.4 RAM Space

1222 bytes of RAM are included, comprising 198 bytes of LCD video XRAM and 512 bytes VTRBF work RAM. The special function registers (SFR) are located in the top address range (100H to 1FFH) of RAM.

Table 3-2 RAM configuration

Memory	Capacity
RAM size	1222 bytes
XRAM	Bank 0 180H - 1FBH (96 bytes)
	Bank 1 180H - 1FBH (96 bytes)
	Bank 2 180H - 185H (6 bytes)
Main RAM	Bank 0 000H - 0FFH (256 bytes)
	Bank 1 000H - 0FFH (256 bytes)
VTRBF	166H (256 bytes × 2 banks)

3.4.1 Indirect Address Registers

The 16-byte address range 00H to 0FH in RAM contains 4 banks of indirect address registers. Starting from the lowest address, these consist of @R0, @R1 (for RAM), @R2, @R3 (for SFR). For addressing, the indirect address register banks are specified by bits 3 and 4 of the program status word (PSW) (indirect address register bank flag: IRBK0, 1). This 16-byte area can also be used as regular RAM.

The relationship between indirect address registers and RAM is shown in the table below.

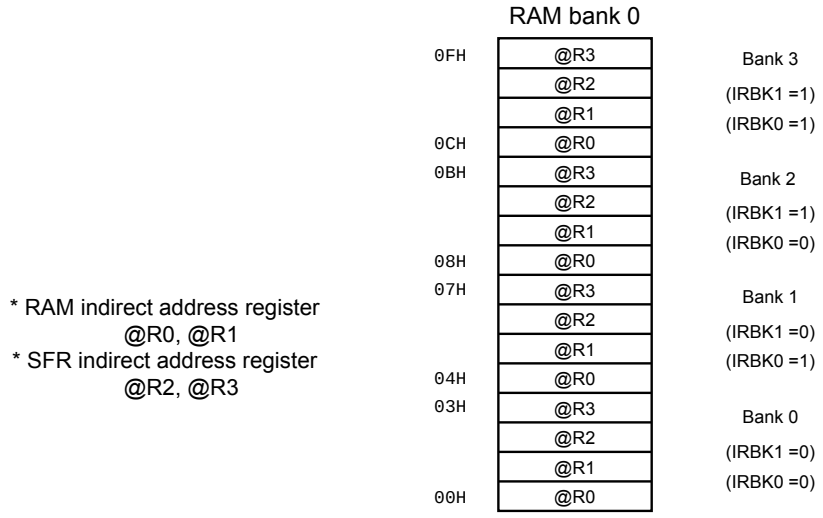


Fig. 3-3 Indirect Address Register Arrangement

Table 3-3 Indirect Address Register Map

Indirect address register name	Function	Bank 0 (IRBK1=0) (IRBK0=0)	Bank 1 (IRBK1=0) (IRBK0=1)	Bank 2 (IRBK1=1) (IRBK0=0)	Bank 3 (IRBK1=1) (IRBK0=1)
_R0	RAM access	RAM 00H	RAM 04H	RAM 08H	RAM 0CH
_R1	RAM access	RAM 01H	RAM 05H	RAM 09H	RAM 0DH
_R2	SFR access	RAM 02H	RAM 06H	RAM 0AH	RAM 0EH
_R3	SFR access	RAM 03H	RAM 07H	RAM 0BH	RAM 0FH

(1) Direct addressing mode

When executing instructions such as MOV #i8, d9

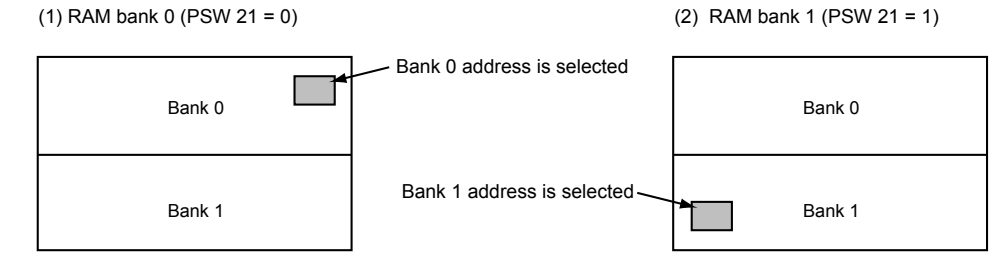


Fig. 3-4 Direct Addressing Mode

(2) Indirect addressing mode

When executing instructions such as MOV #i8, @Rj

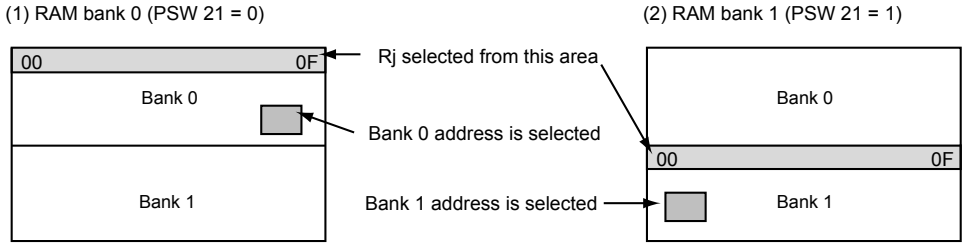


Fig. 3-5 Indirect Addressing Mode

3.4.2 Special function registers (SFR)

A table of RAM and SFR is shown in Table 3-4. For information on the various registers in the SFR range, refer to the sections on the various items.

Caution	The initial values are the values established by the BIOS after a reset.

R = READ X = Undetermined
W = WRITE H = Does not exist

Table 3-4 RAM Memory Map

Symbol	Address	R/W	Designation	Default value	See page
RAM (bank 0)	000H-0FFH	R/W	Data memory	XXXXXXXX (stored at reset)	43
RAM (bank 1)	000H-0FFH	R/W	Data memory	XXXXXXXX (stored at reset)	43
ACC	100H	R/W	Accumulator	00000000	50
PSW	101H	R/W	Program status word	00H00000	52
B	102H	R/W	B register	00000000	51
C	103H	R/W	C register	00000000	51
TRL	104H	R/W	Table reference register lower byte	00000000	54
TRH	105H	R/W	Table reference register upper byte	00000000	54
SP	106H	R/W	Stack pointer	XXXXXXXX	53
PCON	107H	R/W	Power control register	HHHHHH00	158
IE	108H	R/W	Master interrupt enable control register	0HHHHH00	138
IP	109H	R/W	Interrupt priority control register	00000000	151
EXT	10DH	R/W	External memory control register	HHHH0000	_
OCR	10EH	R/W	Oscillation control register	0H00HH00	156
T0CNT	110H	R/W	Timer 0 control register	00000000	67
T0PRR	111H	R/W	Timer 0 prescaler data	00000000	71
T0L	112H	R	Timer 0 low	00000000	71
T0LR	113H	R/W	Timer 0 low reload data	00000000	71
T0H	114H	R	Timer 0 high	00000000	72
T0HR	115H	R/W	Timer 0 high reload data	00000000	72
T1CNT	118H	R/W	Timer 1 control register	00000000	83
T1LC	11AH	R/W	Timer 1 low comparison data	00000000	86
T1L	11BH	R	Timer 1 low	00000000	85
T1LR		W	Timer 1 low reload data	00000000	85
T1HC	11CH	R/W	Timer 1 high comparison data	00000000	87
T1H	11DH	R	Timer 1 high	00000000	86
T1HR		W	Timer 1 high reload data	00000000	86
MCR	120H	W	Mode control register	00000000	127
STAD	122H	R/W	Start address register	00000000	129
CNR	123H	W	Character count register	H0000000	130
TDR	124H	W	Time division register	HH000000	130
XBNK	125H	R/W	Bank address register	HHHHHH00	130
VCCR	127H	W	LCD contrast control register	00000000	131
SCON0	130H	R/W	SIO0 control register	00H00000	108
SBUF0	131H	R/W	SIO0 buffer	00000000	113
SBR	132H	R/W	SIO0 baud rate generator	00000000	113
SCON1	134H	R/W	SIO1 control register	00000000	111
SBUF1	135H	R/W	SIO1 buffer	00000000	113
P1	144H	R/W	Port 1 latch	00000000	58
P1DDR	145H	W	Port 1 data direction register	00000000	58

Symbol	Address	R/W	Designation	Default value	See page
P1FCR	146H	W	Port 1 function control register	10111111	59
P3DDR	14DH	W	Port 3 data direction register	00000000	62
P3INT	14EH	R/W	Port 3 interrupt function control register	11111101	62
P7	15CH	R	Port 7 latch	HHHHXXXX	64
I01CR	15DH	R/W	External interrupt 0, 1 control	00000000	135
I23CR	15EH	R/W	External interrupt 2, 3 control	00000000	137
ISL	15FH	R/W	Input signal select	11000000	138
VSEL	163H	R/W	Control register	11111100	143
VRMAD1	164H	R/W	System address register 1	00000000	144
VRMAD2	165H	R/W	System address register 2	HHHHHHH0	144
VTRBF	166H	R/W	Send/receive buffer	XXXXXXXX	144
BTCR	17FH	R/W	Base timer control	01000001	101
RAM (XRAM) (Bank 0)	180H-1FBH	R/W	LCD memory	XXXXXXXX (stored at reset)	126
RAM (XRAM) (Bank 1)	180H-1FBH	R/W			
RAM (XRAM) (Bank 2)	180H-185H	R/W			

3.5 Flash Memory

The VMS custom chip comprises a 128 KB flash memory space which consists of two 64-KB banks. Reading and writing data from and to the flash memory is performed by calling the appropriate OS program. By using the ROM table lookup instruction (LDC), ROM space data can be accessed. Applications are always placed in the 64 KB memory space of bank 0. Switching between the system BIOS (ROM) and an application (flash memory) is performed by a dedicated macro instruction (CHANGE).

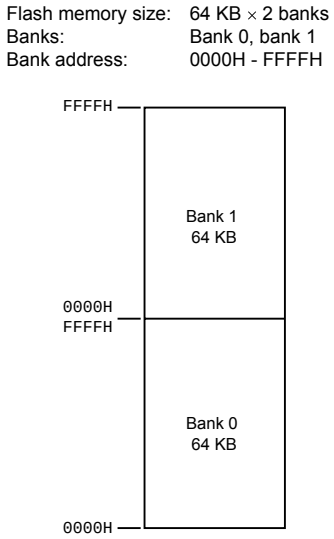


Fig. 3-6 Flash Memory Map

Data read/write for the flash memory is performed by calling an OS program. For details, refer to chapter 12 “Subroutine Reference” in the System BIOS manual.

3.6 Accumulator

The accumulator (ACC) is an 8-bit register used for data arithmetic processing, transfer, I/O operations etc. It is assigned to address 100H of SFR, and initialized to 00H after a reset.

Unlike in a conventional CPU, a part of the memory is used to serve as accumulator.

Accumulator (ACC)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ACC	100H	R/W	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0
Reset			0	0	0	0	0	0	0	0

3.7 B Register, C Register

The B register and C register are 8-bit registers used in combination with the accumulator for arithmetic operations. They are assigned to address 102H (B register) and address 103H (C register) of SFR, and initialized to 00H after a reset.

B register

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B	102H	R/W	B7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
Reset			0	0	0	0	0	0	0	0

C register

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C	103H	R/W	C7	C6	C5	C4	C3	C2	C1	C0
Reset			0	0	0	0	0	0	0	0

Multiplication is performed using 16 bits × 8 bits. For the multiplicand (16 bits), the upper 8 bits are stored in the accumulator and the lower 8 bits in the C register. The multiplier (8 bits) is stored in the B register. The processing result (product) has 24 bits. The top 8 bits are stored in the B register, middle 8 bits in the accumulator, and lower 8 bits in the C register. Therefore, the following applies:

(ACC) (C) × (B) = (B) (ACC) (C)

Division is performed using 16 bits ÷ 8 bits. For the dividend (16 bits), the upper 8 bits are stored in the accumulator and the lower 8 bits in the C register. The divisor (8 bits) is stored in the B register. The processing result (quotient) has 16 bits. The upper 8 bits are stored in the accumulator, and the lower 8 bits in the C register. The surplus is stored in the B register. Therefore, the following applies:

(ACC) (C) ÷ (B) = (B) (ACC) (C) mod (B)

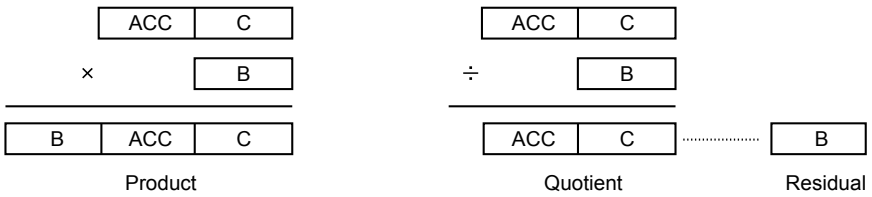


Fig. 3-7 Arithmetic Register Contents

3.8 Program Status Word (PSW)

The program status word (PSW) consists of flags indicating the arithmetic processing result status and flags specifying the RAM banks and indirect address registers. It is assigned to address 101H of SFR, and initialized to 0 after a reset.

Program status word (PSW)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PSW	101H	R/W	CY	AC	_	IRBK1	IRBK0	OV	RAMBK 0	P
Reset			0	0	0	0	0	0	0	0

CY (bit 7): carry flag

CY is set (1) when the processing result carries over to the next higher digit (carry) or to the next lower digit for subtraction and comparison (borrow). Otherwise the flag is reset (0). The flag is influenced by rotating instructions that include CY, and is reset (0) when an arithmetic instruction is executed.

AC (bit 6): auxiliary carry flag

AC is set (1) when the ACC bit 3 carries over to the next higher digit (carry) or to the next lower digit (borrow). Otherwise the flag is reset (0).

IRBKx: indirect address register bank flag

Consists of IRBK1 (bit 4) and IRBK0 (bit 3) which specify indirect address register bank flag 1 and indirect address register bank flag 0.

Serve for specifying the 4 register banks used as indirect address registers for indirect addressing within each RAM bank.

Bank	IRBK1	IRBK0
0	0	0
1	0	1
2	1	0
3	1	1

OV (bit 2): overflow flag

When overflow occurs, the OV bit is set (1). Otherwise it is reset (0). This means that the bit is set when the result of an arithmetic operation involving “negative number” + “negative number” or “negative number” - “positive number” is positive, or when the result of an arithmetic operation involving “positive number” + “positive number” or “positive number” - “negative number” is negative. For multiplication and division, the bit is set when the contents of the B register are not 0, and reset when the contents of the B register are 0.

RAMBK0 (bit 1): RAM bank flag

Serves for specifying the RAM bank. When an instruction performs RAM access, the RAM address within the specified bank is accessed.

Bank	RAMBK0
0	0
1	1

P (bit 0): accumulator (ACC) parity flag

When the total number of bits set in the accumulator is odd, this bit is set (1). When the number is even, the bit is reset (0). This bit is read-only.

3.9 Stack Pointer

RAM bank 0 is used as stack memory. The 8-bit SP register is used to specify addresses in the stack area.

SP is assigned to address 106H of SFR. It is incremented before data are moved into stack memory and decremented after data are fetched from stack memory.

After a reset, SP is undetermined, but system programs initialize it to 7FH. After SP is initialized, the application is called.

Caution	The stack is used from RAM bank 0 address 80H upwards (towards 0ffH). The clock function uses up to 20 bytes of the stack, leaving 108 bytes for the application. When the PUSH instruction is executed, data are stored only after SP was incremented. Also when PUSH or POP are used during access of RAM bank 1, the data will be stored in the RAM bank 0 stack area.

Stack pointer (SP)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SP	106H	R/W	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
Reset			X	X	X	X	X	X	X	X

When the PUSH instruction is executed, SP is incremented and the SFR and RAM data specified by the operand are moved out. When the POP instruction is executed, the data are moved back into the SFR and RAM specified by the operand, and SP is decremented.

Also when RAM bank 1 was specified for a PUSH or POP operation, data are stacked in RAM bank 0. When the RAM address is used as operand, bank 0 (not bank 1) is accessed.

When a CALL instruction is executed, SP is incremented, and the lower 8 bits of the program counter (PC) are moved to the stack. Then SP is incremented and the upper 8 bits of the PC are moved to the stack. When a RET instruction is executed, data specified by SP are stored as the upper 8 bits of the PC, SP is decremented, and the data specified by the SP are stored as the lower 8 bits of the PC. SP is then decremented further.

When an interrupt is received, SP is incremented, and the lower 8 bits of the PC are moved to the stack. Then SP is incremented again, and the upper 8 bits of the PC are moved to the stack. When a RET1 instruction for returning from interrupt processing is executed, the upper 8 bits of the PC are stored, SP is decremented, and the data specified by the SP are stored as the lower 8 bits of the PC. SP is then decremented further.

3.10 Table Reference Register (TRR)

The table reference register (TRR) is a 16-bit register that serves for ROM and flash memory addressing. The lower byte (TRL) is assigned to address 104H of SFR and the upper byte (TRH) to address 105H of SFR. During reset, the register is initialized to 00H.

The table lookup instruction (LDC) adds the data stored in the TRR to the data stored in the accumulator and uses the result as address for reading data and transferring them to the accumulator. During flash memory read/write (using OS programs), the data stored in the TRR are used as address for the specified bank.

Table reference register (lower byte) (TRL)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRL	104H	R/W	TRL7	TRL6	TRL5	TRL4	TRL3	TRL2	TRL1	TRL0
Reset			0	0	0	0	0	0	0	0

Table reference register (upper byte) (TRH)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRH	105H	R/W	TRH7	TRH6	TRH5	TRH4	TRH3	TRH2	TRH1	TRH0
Reset			0	0	0	0	0	0	0	0

3.11 CHANGE Instruction

The CHANGE instruction serves for switching between the system BIOS and the application. When a system program is running, the instruction causes a change to the application mode. The program counter is reset to the address specified by label or address.

Format

CHANGE <label name or address>

Operation

As described below, operation of the CHANGE instruction differs, depending on whether it is executed while a system program or the application is running. The actual mode shift occurs after the dedicated macro instruction was executed.

(1) System program running

The system switches from the system program to the application (game mode). The program counter is reset to the application address specified by label or address.

(2) Application running

The system switches from the application (game mode) to the system program.

However, if bit 1 (LDCEXT) of the external memory control register is set, the CHANGE instruction will not cause a change to the system program. The application continues to run.

The program counter is reset to the system program address specified by label or address.

Sample program

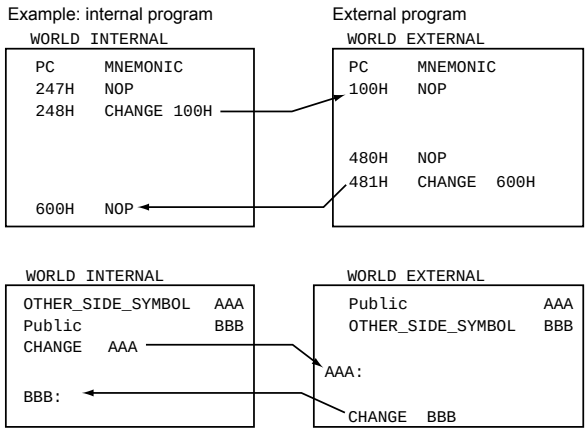


Fig. 3-8 System Program ⟷ Application Transition

Chapter 4 Peripheral System Configuration

This section gives details about peripheral devices including I/O ports, timer, serial communication, etc.

4.1 I/O Ports

The VMS custom chip has three I/O ports which are all mapped to memory using Special Function Registers (SFR). For ports 1 and 3, data direction register (PnDDR) determine the I or O assignment. Port 1 is used only for the serial interface and dedicated Dreamcast interface. Port 7 is a dedicated input port for the VMS buttons.

After a reset, all ports are set as input ports, and the port latch is "0".

To use the I/O ports, the following Special Function Registers must be operated.

Port 1 (P1)	• P1	• P1DDR	• P1FCR	
Port 3 (P3)	• P3	• P3DDR	• P3INT	• EXT
Port 7 (P7)	• P7	(dedicated input port)		

Caution	When reading an I/O port, depending on the instruction, data may be either latched (Fig. 4-1 left) or read directly from the port (Fig. 4-1 right). This must be taken into consideration when reading I/O port data. When reading an I/O port, some instructions read port latched data. BPC, DBNZ, INC, DEC, SET1, CLR1, NOT1

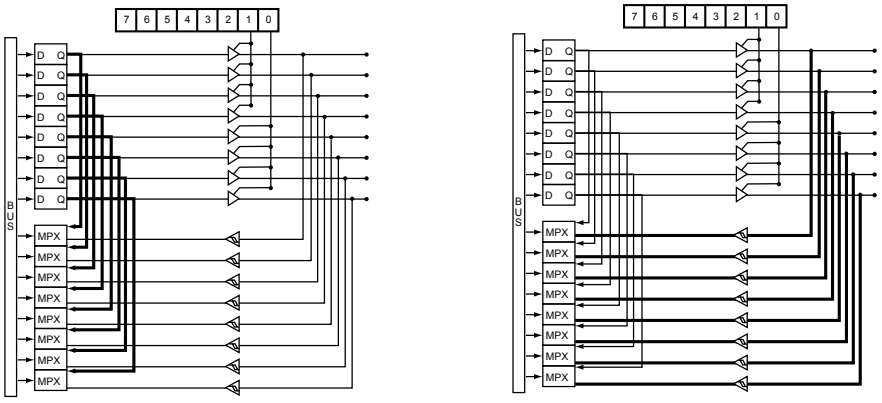


Fig. 4-1 Instruction and Data Path

4.1.1 Port 1

Port 1 can be used as I/O port for the serial interface of the VMS, or for the dedicated Dreamcast interface.

Applications can use only SIO (P10 - P15). To operate these registers, be sure to use bit-level instructions. For details on SIO output, refer to the section 4.5 “Serial Interface”.

Caution	When coding VMS applications, the following operations must be included. **Standalone operation (SIO not used)** 1. Monitor port 7 to detect 5V. 2. Store values of bits 2 and 5 of port 1. 3. When 5V is detected, change bits 2 and 5 of port 1 to port data output mode and output “0” for these bits. 4. Reset stored values of bits 2 and 5 of port 1. If these operations are not performed, the VMS may not be recognized correctly when connected to the Dreamcast. Except for the above operations and for serial data transfer, port 1 registers should not be operated by an application.

Port 1 latch (P1): 144H

Port 1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	P17	P16	P15	P14	P13	P12	P11	P10
Function	Pulse output	TEST	SCK1	SB1	SO1	SCK0	SB0	SO0

Port 1 data direction register (P1DDR): 145H

Stmbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIDDR	145H	W	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Reset			0	0	0	0	0	0	0	0

Caution	The data direction register for port 1 is a write-only register corresponding to each data latch bit. When a bit operation instruction or an instruction such as INC, DEC, or DBNZ is used for a write-only register, bits other than the specified bit become “1”. For the P1DDR, use the following instructions. **MOV, MOV @, ST, ST @, POP**

Bit name	Function
P17DDR (bit 7)	Input control
$\overline{P10}$DDR (bit 7)	0: Input mode 1: Output mode

P1nDDR (bit 7 to 0): P17 - P10 I/O control

Specifies whether bits 7 to 0 of port 1 are used for input or output.

When set to "1", P1n is in output mode.

When reset to "0", P1n is in input mode.

Port 1 function control register (P1FCR): 146H

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIFCR	146H	W	P17FCR	P16FCR	P15FCR	P14FCR	P13FCR	P12FCR	P11FCR	P10FCR
____			0	0	0	0	0	0	0	0

Caution	The data direction register for port 1 is a write-only register. When a bit operation instruction or an instruction such as INC, DEC, or DBNZ is used for a write- only register, bits other than the specified bit become "1". For the P1FCR, use the following instructions. **MOV, MOV @, ST, ST @, POP**

Bit name	Function
P17CR (bit 7)	P17 control function
	0: Port data (P17) output 1: PWM output
P16CR (bit 6)	Use prohibited
	0: Port data (P16) output (fixed) 1: Not allowed
P15CR (bit 5)	P15 control function
	0: Port data (P15) output 1: Serial interface data (SCK1) output
P14CR (bit 4)	P14 control function
	0: Port data (P14) output 1: Serial interface data (SB1) output
P13CR (bit 3)	P13 control function
	0: Port data (P13) output 1: Serial interface data (SO1) output
P12CR (bit 2)	P12 control function
	0: Port data (P12) output 1: Serial interface data (SCK0) output
P11CR (bit 1)	P11 control function
	0: Port data (P11) output 1: Serial interface data (SB0) output
P10CR (bit 0)	P10 control function
	0: Port data (P10) output 1: Serial interface data (SO0) output

P17FCR (bit 7): Select P17 function

Controls the PWM assigned to P17. When set to “1”, the logical sum of the PWM signal and the port latch data is output. When reset to “0”, the port latch data are output.

P16FCR (bit 6): Select P16 function

This bit is fixed to “0”. It may not be manipulated by an application.

P15FCR (bit 5): Select P15 function

Controls the clock assigned to P15 for serial transfer 1. When set to “1”, the logical sum of the serial interface clock (SCK1) and port latch data is output. When reset to “0”, port latch data are output.

P14FCR (bit 4): Select P14 function

Controls the data assigned to P14 for serial transfer 1. When set to “1”, the logical sum of the serial interface data (SB1) and port latch data is output. When reset to “0”, port latch data are output.

P13FCR (bit 3): Select P13 function

Controls the data assigned to P13 for serial transfer 1. When set to “1”, the logical sum of the serial interface data (S01) and port latch data is output. When reset to “0”, port latch data are output.

P12FCR (bit 2): Select P12 function

Controls the clock assigned to P12 for serial transfer 0. When set to “1”, the logical sum of the serial interface clock (SCK0) and port latch data is output. When reset to “0”, port latch data are output.

P11FCR (bit 1): Select P11 function

Controls the data assigned to P11 for serial transfer 0 When set to “1”, the logical sum of the serial interface data (SB0) and port latch data is output. Serial interface data can always be input.

P10FCR (bit 0): Select P10 function

Controls the data assigned to P10 for serial transfer 0. When set to “1”, the logical sum of the serial interface data (S00) and port latch data is output. When reset to “0”, port latch data are output.

Caution	• To use the function assigned to port 1, the corresponding port latch must be reset to "0". For example, to use PWM, set P17FCR to "1" and reset P17 to "0". • The instructions BPC, DBNZ, INC, DEC, SET1, CLR1, NOT1 read port latch data. Other instructions read data assigned to the port.

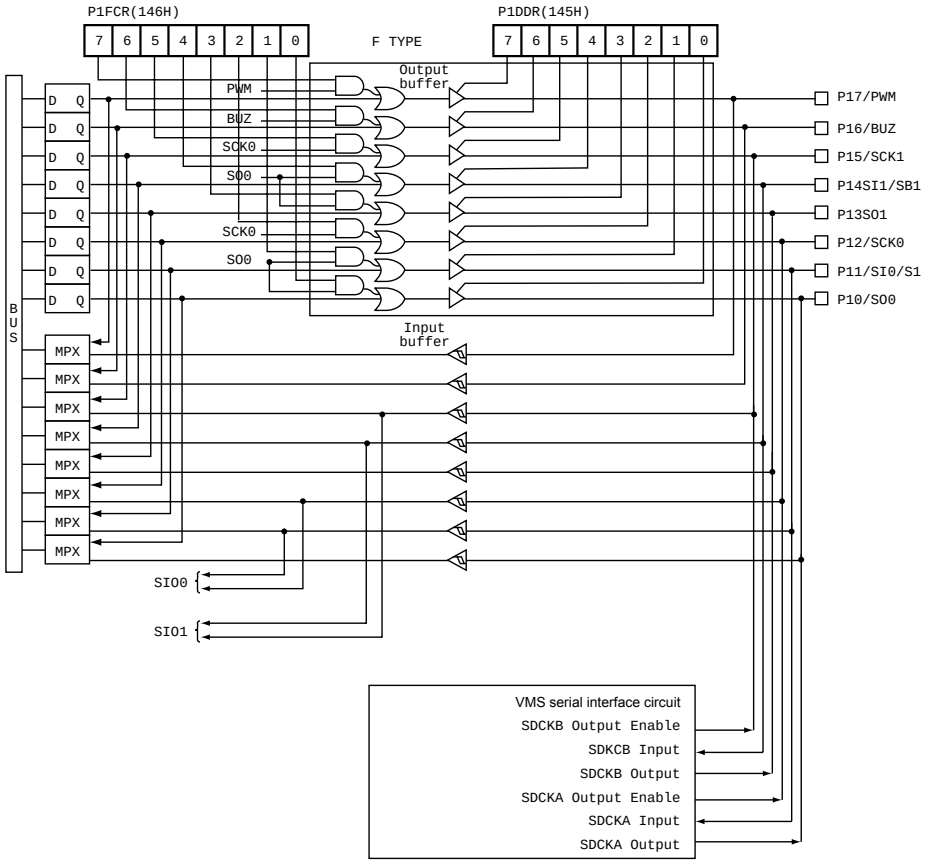


Fig. 4-2 Port 1 Block Diagram

4.1.2 Port 3

Port 3 is an input-only port dedicated to the VMS direction buttons, A button, B button, MODE button, and SLEEP button.

Port 3 latch (P3)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3	14CH	R/W	P37	P36	P35	P34	P33	P32	P31	P30
Function			SLEEP	MODE	B button	A button	RIGHT	LEFT	DOWN	UP
Reset			0	0	0	0	0	0	0	0

Bits 0 to 7 of port are programmable pull-up bits. The application must set the bit corresponding to the button to be detected to "1". When the button is pressed, the bit is reset to "0".

Caution	Measures against simultaneous presses of different direction buttons must be taken by the application.

Port 3 data direction register (P3DDR): 14DH

This register may not be manipulated by an application.

Port 3 interrupt control register (P3INT)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3INT	14EH	R/W	–	–	–	–	–	P32INT	P31INT	P30INT
Reset			H	H	H	H	H	0	0	0

Bit name	Function
P32INT (bit 2)	Port 3 interrupt control flag
	0: Port interrupt disabled 1: Port interrupt enabled
P31INT (bit 1)	Port 3 interrupt source flag
	0: Interrupt source disabled 1: Interrupt source enabled
P32INT (bit 0)	Port 3 interrupt request enable
	0: Interrupt request disabled 1: Interrupt request enabled

P32INT (bit 2): port 3 interrupt generation selector flag

Determines whether an interrupt is generated while a button connected to port 3 is pressed. Whereas the P30INT (bit 0) flag selects whether a generated interrupt is accepted or not, this flag controls interrupt generation itself.

When reset to "0", no interrupt is generated.

When set to "1", an interrupt is generated.

Caution	The port 3 interrupt is a level interrupt which is generated continuously for as long as the button is pressed.

P31INT (bit 1): port 3 interrupt source flag

This flag is relevant if the P32INT flag is set. The port 3 interrupt request status is monitored, and the flag is set to "1" when an interrupt request is generated by port 3. When no interrupt request is generated, the flag does not change. This allows an interrupt processing routine to specify an interrupt source.

Caution	This flag must be reset by the application. Use a suitable interrupt processing routine to do this.

P30INT (bit 0): port 3 interrupt request enable control

Enables (1) or disables (0) interrupt requests from port 3. When reset to "0", interrupt processing is disabled and the interrupt processing routine is not called. When set to "1", the interrupt vector 004BH is called when an interrupt is generated (P31INT = 1).

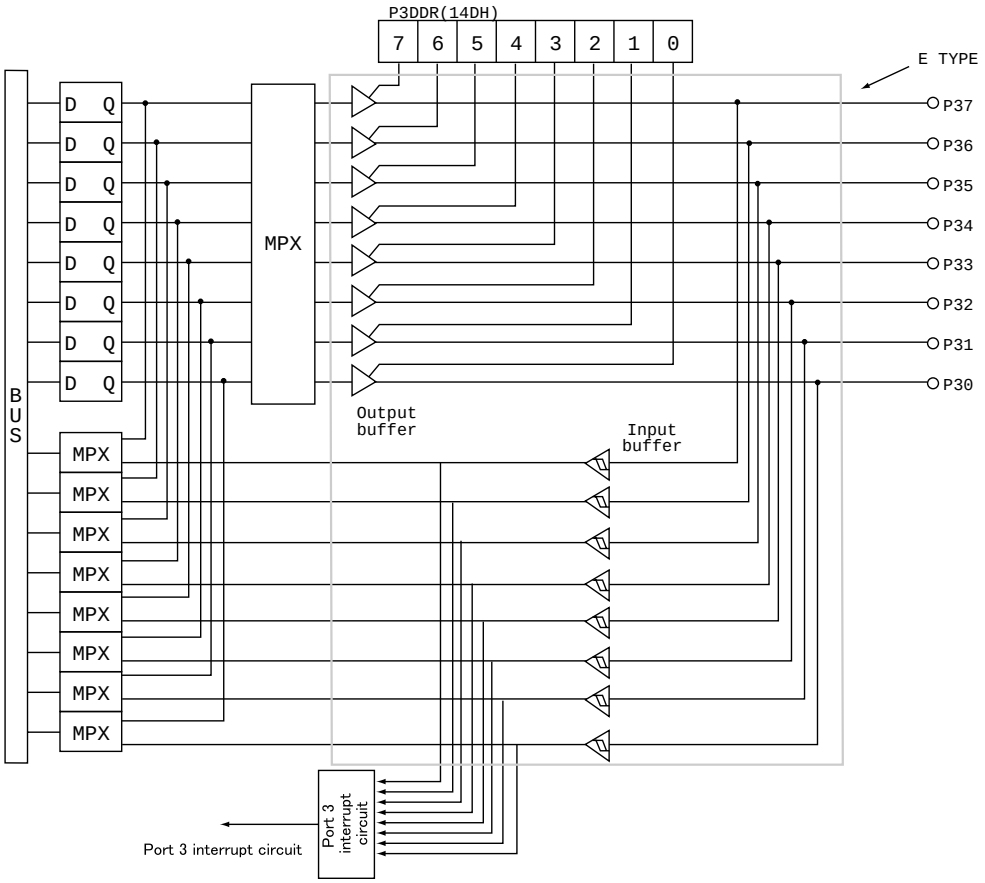


Fig. 4-3 Port 3 block diagram

4.1.3 Port 7

Port 7 is a dedicated input port that serves for low- voltage detection and for checking the connection status to the Dreamcast.

Port 7 (P7)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P7	15CH	R	–	–	–	–	P73	P72	P71	P70
Function			–	–	–	–	ID1	ID0	Low voltage	5V detection
Reset			H	H	H	H	0	0	1	0

Bits 0 to 3 of port 7 are pulled up. Immediately after a reset, bit 1 is set to "1" and all other bits are reset to "0".

Caution	VMS applications should be designed in such a way as to store data in flash memory and then terminate quickly when the VMS is connected to the Dreamcast controller while the application is running. The application should monitor bit 0 of port 7. When detecting that bit was set to "1", the same application termination routine as when the MODE button is pressed should be carried out, and control should be returned to the system BIOS. The interrupt INT0 can be used to detect the connection status.

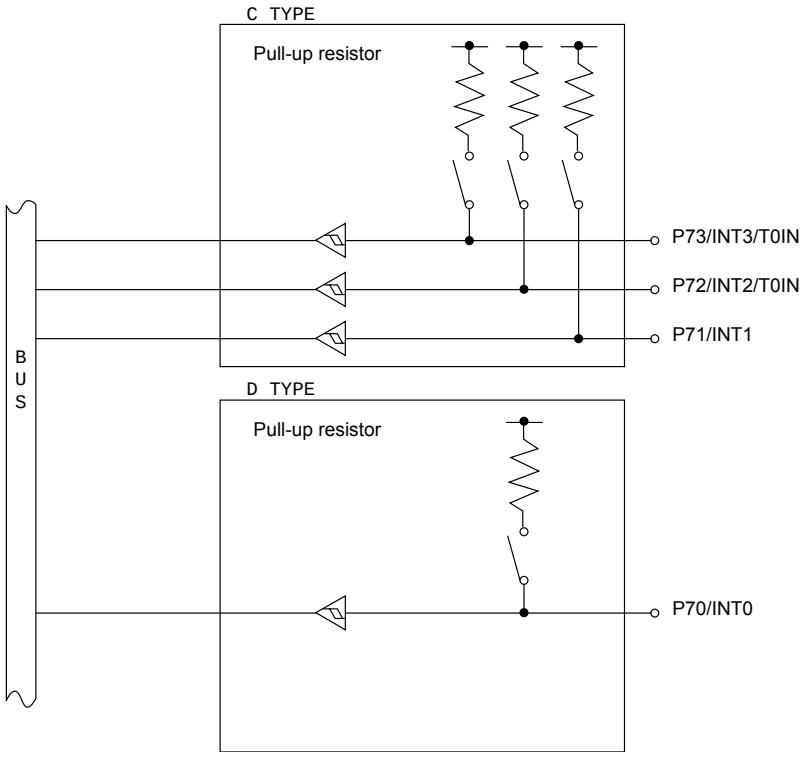


Fig. 4-4 Port 7 Block Diagram

4.2 Timer/Counter 0 (T0)

The timer/counter 0 (T0) in the VMS custom chip is a 16- bit timer/counter with the following 4 functions. The prescaler of timer 0 is an 8-bit type.

The following Special Function Registers are used to control the timer/counter 0.

T0H, T0HR, T0L, T0LR, T0CNT, T0PRR, ISL, I23CR

- Mode 0: 8-bit reload timer × 2 channels
- Mode 1: 8-bit reload timer + 8-bit reload counter
- Mode 2: 16-bit reload timer
- Mode 3: 16-bit reload counter

4.2.1 Functions

8-bit reload timer × 2 channels (mode 0)

The clock from the 8-bit prescaler is used to drive two separate 8-bit reload timers (T0H, T0L).

8-bit reload timer + 8-bit reload counter (mode 1)

T0H operates as an 8-bit reload timer driven by the prescaler clock. T0L performs counting by detecting the input signal at the P72/INT2/T0IN and P73/INT3/T0IN pins.

16-bit reload timer (mode 2)

The clock from the 8-bit prescaler is used to drive the 16-bit reload timer (T0H + T0L).

16-bit reload counter (mode 3)

The overflow of T0L is used as clock for T0H, to drive the 16-bit reload counter. T0L counts the input signal at the P72/INT2/T0IN and P73/INT3/T0IN pins.

Interrupt generation

When the interrupt enable bit is set, overflow of the register T0H or T0L generates a T0H or T0L interrupt.

4.2.2 Circuit Configuration

The timer/counter 0 (T0) configuration is shown in Fig. 4- 5.

Prescaler … ❷

The prescaler is an 8-bit programmable counter that operates constantly while the system is on.

The cycle clock is a signal generated with each cycle when an instruction is executed and at HALT mode.

Timer/counter 0 low (T0L) … ➧

This 8-bit reload timer/counter uses the prescaler output or an external signal (from other VMS) as a clock.

In modes 0 and 1, T0L is the overflow. In modes 2 and 3, T0H is the overflow. The T0LR (reload register) contents are reloaded to the respective counter. Reloading is carried out also when the T0LRUN (T0CNT bit 6) is reset and the counter stops.

Timer/counter 0 high (T0H) … ➥

This 8-bit reload timer/counter uses the prescaler output or the T0L overflow as a clock.

At T0H overflow the contents of the timer 0 high reload register (T0HR) are reloaded. Reloading is carried out also when the T0HRUN (T0CNT bit 7) is reset and the counter stops.

Timer/counter 0 control register (T0CNT) … ❸

Serves for T0 mode 0 to 3 setting and interrupt control.

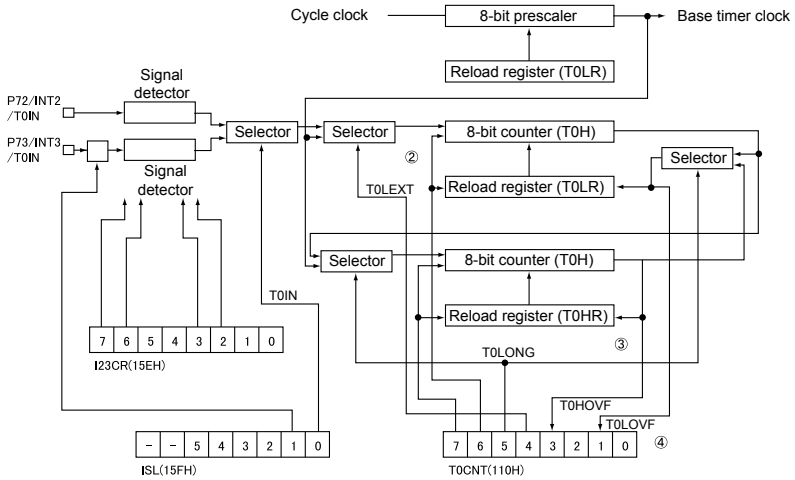


Fig. 4-5 Timer/Counter 0 Block Diagram

4.2.3 Related Registers

Timer/counter 0 control register (T0CNT)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CNT	14EH	R/W	P0HRUN	P0LRUN	P0LONG	P0LEXT	P0HOVF	T0HIE	T0LOVF	T0LIE
Reset			0	0	0	0	0	0	0	0

Bit name	Function
P0HRUN (bit 7)	T0H count control
	0: Count stop/data reload 1: Count start
P0LRUN (bit 6)	T0L count control
	0: Count stop/data reload 1: Count start
P0LONG (bit 5)	Timer/counter 0 bit length selector
	0: 8 bit 1: 16 bit
P0LEXT (bit 4)	T0L input clock select
	0: Prescaler output 1: External pin input signal Pin for external input can be specified by input select register (ISL)
P0HOVF (bit 3)	T0H overflow flag
	0: No overflow flag 1: Overflow flag
T0HIE (bit 2)	T0H interrupt request enabled
	0: Interrupt request disabled 1: Interrupt request enabled
T0LOVF (bit 1)	T0L overflow flag
	0: No overflow flag 1: Overflow flag
T0LIE (bit 0)	T0L interrupt request enabled
	0: Interrupt request disabled 1: Interrupt request enabled

T0HRUN (bit 7): T0H count control

Controls count-up start/stop of timer/counter 0 high (T0H). When set to “1”, the T0H clock is supplied and counting starts. To stop counting, the bit must be reset to “0”. This stops the clock and sends the reload data (T0HR) to T0H.

T0LRUN (bit 6): T0L count control

Controls count-up start/stop of timer/counter 0 low (T0L). When set to “1”, the T0L clock is supplied and counting starts. To stop counting, the bit must be reset to “0”. This stops the clock and sends the reload data (T0LR) to T0L.

T0LONG (bit 5): Timer/counter 0 bit length select

Specifies the bit length of T0. "1" selects a 16-bit counter and "0" an 8-bit counter.

For mode 0 or 1, specify "0", and for mode 2 or 3, specify "1".

Mode	T0LONG	T0LEXT
0	0	0
1	0	1
2	1	0
3	1	1

T0LEXT (bit 4): T0L input clock select

Specifies the clock supplied to T0L. The clock can be either an external signal (from a another connected VMS) or the prescaler output.

When the bit is set to "1", the external input signal is selected. When the bit is reset to "0", the prescaler output is selected.

When the external signal is selected (1), either port P72 (INT2/T0IN pin) or P73 (INT3/T0IN pin) can be used. as T0L clock. Switching between P72 and P73 is performed by the input signal select register (ISL).

T0HOVF (bit 3): T0H overflow flag

This flag is set when T0H overflow has occurred. If there is no overflow, the flag does not change.

This flag must be reset by the T0H interrupt processing routine or another routine of the application.

T0HIE (bit 2): T0H interrupt request enable control

Enables or disables interrupt request generation at T0H overflow.

When set to "1", the interrupt vector 0023H is called when T0H overflow occurs. When reset to "0", no interrupt request is generated.

T0LOVF (bit 1): T0L overflow flag

This flag is set when T0L overflow has occurred. If there is no overflow, the flag does not change.

This flag must be reset by the T0L interrupt processing routine or another routine of the application.

When the 16-bit counter is used, the flag is not set also when overflow occurs. When T0H overflow occurs, it is set together with T0HOVF.

T0LIE (bit 0): T0L interrupt request enable control

Enables or disables interrupt request generation at T0L overflow.

When set to "1", the interrupt vector 0013H is called when T0H overflow occurs. When reset to "0", no interrupt request is generated.

Caution	• The overflow flags (T0HOVF, T0LOVF) must be reset to "0" by the respective interrupt processing routine of the application. • When using the 16-bit counter, set T0HRUN and T0LRUN together to "1". • When using the 16-bit counter, set T0HOVF and T0LOVF together to "1".

Input signal select register (ISL)

This register serves to select the time constant for the noise filter connected to P73 (INT3/T0IN pin) and to select the external signal.

Caution	This register may not be manipulated by an application.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ISL	15FH	R/W	–	–	ISL5	ISL4	ISL3	ISL2	ISL1	ISL0
Reset			H	H	0	0	0	0	0	0

Bit name	Function		
ISL5 (bit 5) ISL4 (bit 4)	Base timer clock select		
	ISL5	ISL4	
	1	1	Timer/counter T0 prescaler
	0	1	Cycle clock
	X	0	Quartz oscillator
ISL3 (bit 3)	Use prohibited		
	0: fBST/16 (fixed) 1: Not allowed		
ISL2 (bit 2) ISL1 (bit 1)	Noise filter time constant select		
	ISL2	ISL1	
	1	1	16Tcyc
	0	1	64Tcyc
	X	0	1Tcyc
ISL0 (bit 0)	T0 clock input pin select		
	0: P72/INT2/T0IN pin 1: P73/INT3/T0IN pin		

ISL5, ISL4 (bits 5, 4): base timer clock select

Select the base timer input clock.

ISL5	ISL4	Base timer input clock
0	0	Quartz oscillator

ISL3 (bit 3): prohibited

Use of this bit is prohibited.

ISL2, ISL1 (bits 2, 1): noise filter time constant select

Selects the time constant of the noise filter.

ISL2	ISL1	Time constant
0	0	1Tcyc

The table below shows values for the signal time constant and noise range figures.

Time constant	Noise *1	Noise/ signal *2	Noise *3
1Tcyc	< 1Tcyc	1Tcyc – 2Tcyc	2Tcyc <

Caution	• A signal not matching the time constant conditions is considered noise and is not input. • Sometimes even a signal matching the time constant conditions may be considered noise and not input. • A signal matching the time constant conditions is considered normal and is input.

ISL0 (bit 0): T0 clock input select

Sets port T0 external signal input to P73 (INT3/T0IN pin) or P72 (INT2/T0IN pin).

When reset to “0”, the signal at P72 (INT2/T0IN pin) is used as T0 clock.

When set to “1”, the signal at P73 (INT3/T0IN pin) is used as T0 clock.

Timer 0 prescaler register (T0PRR)

The timer 0 prescaler register (T0PRR) serves for setting the timer/counter 0 clock frequency. The 8-bit programmable counter allows 256 different settings.

The 8-bit prescaler uses the cycle clock directly as its clock. By setting the desired data in T0PRR (111H), the timer/counter 0 clock frequency TPR can be set.

8-bit prescaler: TPR = 1 × (256 - [T0PRR]) (decimal)

Tcyc: Cycle clock

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0PRR	111H	R/W	T0PRR7	T0PRR6	T0PRR5	TOPRR4	T0PRR3	T0PRR2	T0PRR1	T0PRR0
Reset			0	0	0	0	0	0	0	0

Timer 0 low register (T0L)

This is an 8-bit timer/counter.

It selects whether the output of the prescaler or the external signal from P72 (INT2/T0IN pin) or P73 (INT3/T0IN pin) is used as clock signal.

The clock is used for count-up. When overflow occurs, the overflow flag is set and an interrupt is generated.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0L	112H	R	T0L7	T0L6	T0L5	T0L4	T0L3	T0L2	T0L1	T0L0
Reset			0	0	0	0	0	0	0	0

Timer 0 low reload register (T0LR)

The data for reloading in the timer/counter 0 low (T0L) are set in this register. When using 8-bit mode, the contents of this register are reloaded into T0L.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0LR	113H	R/W	T0LR7	T0LR6	T0LR5	T0LR4	T0LR3	T0LR2	T0LR1	T0LR0
Reset			0	0	0	0	0	0	0	0

Timer 0 high register (T0H)

This is an 8-bit timer/counter.

Count-up is performed with the prescaler output or the T0L overflow (T0HOVF). When overflow occurs, the overflow flag is set and an interrupt is generated.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0H	114H	R	T0H7	T0H6	T0H5	T0H4	T0H3	T0H2	T0H1	T0H0
Reset			0	0	0	0	0	0	0	0

Timer 0 high reload register (T0HR)

The data for reloading in the timer/counter 0 high (T0H) are set in this register. When T0H overflow has occurred and when the count was stopped (T0HRUN = 0), the contents of this register are reloaded into T0H.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0HR	115H	R/W	T0HR7	T0HR6	T0HR5	T0HR4	T0HR3	T0HR2	T0HR1	T0HR0
Reset			0	0	0	0	0	0	0	0

External interrupt 2, 3 control register (I23CR)

Sets external signal detection and interrupt.

ISL0	I23CR7	I23CR6	I23CR3	I23CR2	External signal condition
1	0	1	–	–	P73/INT3/T0IN falling edge count
1	1	0	–	–	P73/INT3/T0IN rising edge count
1	1	1	–	–	P73/INT3/T0IN dual edge count
0	–	–	0	1	P72/INT2/T0IN falling edge count
0	–	–	1	0	P72/INT2/T0IN rising edge count
0	–	–	1	1	P72/INT2/T0IN dual edge count
–	0	0	0	0	No count

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I23CR	15EH	R/W	I23CR7	I23CR6	I23CR5	I23CR4	I23CR3	I23CR2	I23CR1	I23CR0
Reset			0	0	0	0	0	0	0	0

In combination with the input signal select register ISL0, this register specifies the count conditions. Possible combinations are shown in the following table.

Bit	Function
I23CR7 (bit 7)	INT3 rising edge detection control
	0: No detect 1: detect
I23CR6 (bit 6)	INT3 falling edge detection control
	0: No detect 1: detect
I23CR5 (bit 5)	INT3 interrupt source
	0: Interrupt source disabled 1: Interrupt source enabled
I23CR4 (bit 4)	INT3 interrupt request enabled
	0: Interrupt request disabled 1: Interrupt request enabled
I23CR3 (bit 3)	INT2 rising edge detection control
	0: No detect 1: detect
I23CR2 (bit 2)	INT3 falling edge detection control
	0: No detect 1: detect
I23CR1 (bit 1)	INT2 interrupt source
	0: Interrupt source disabled 1: Interrupt source enabled
I23CR0 (bit 0)	INT2 interrupt request enabled
	0: Interrupt request disabled 1: Interrupt request enabled

I23CR7 (bit 7): INT3 rising edge detection control

Specifies whether to detect the rising edge of the interrupt signal at P73 (INT3/T0IN pin).

When set to “1”, the rising edge of the INT3 signal at P73 is detected. When the INT3 interrupt is generated, I23CR5 is set to “1” and the interrupt processing routine specified by the interrupt vector is called, if interrupt request is enabled (I23CR4 = 1).

When reset to “0”, the rising edge of the interrupt signal is not detected.

I23CR6 (bit 6): INT3 falling edge detection control

Specifies whether to detect the falling edge of the interrupt signal at P73 (INT3/T0IN pin).

When set to “1”, the falling edge of the INT3 signal at P73 is detected. When the INT3 interrupt is generated, I23CR5 is set to “1” and the interrupt processing routine specified by the interrupt vector is called, if interrupt request is enabled (I23CR4 = 1).

When reset to “0”, the rising edge of the interrupt signal is not detected.

I23CR5 (bit 5): INT3 interrupt source

When interrupt edge detection at P73 (INT3/T0IN pin) has occurred, this flag is set.

The flag must be reset by the interrupt processing routine of the application.

I23CR4 (bit 4): INT3 interrupt request enable

Enables or disables the INT3 interrupt.

If set to “1”, the INT3 interrupt vector is called when the I23CR5 flag is set.

If reset to “0”, the interrupt processing routine is not called, also when an interrupt is generated.

I23CR3 (bit 3): INT2 rising edge detection control

Specifies whether to detect the rising edge of the interrupt signal at P72 (INT2/T0IN pin).

When set to “1”, the rising edge of the INT2 signal at P72 is detected. When the INT2 interrupt is generated, I23CR1 is set to “1” and the interrupt processing routine is called, if interrupt request is enabled (I23CR4 = 0).

I23CR2 (bit 2): INT2 falling edge detection control

Specifies whether to detect the falling edge of the interrupt signal at P72 (INT2/T0IN pin).

When set to “1”, the falling edge of the INT2 signal at P72 is detected. When the INT2 interrupt is generated, I23CR1 is set to “1” and the interrupt processing routine is called, if interrupt request is enabled (I23CR4 = 0).

I23CR1 (bit 1): INT2 interrupt source

When interrupt edge detection at P72 (INT2/T0IN pin) has occurred, this flag is set.

The flag must be reset by the interrupt processing routine of the application.

I23CR0 (bit 0): INT2 interrupt request enable

Enables or disables the INT3 interrupt.

If set to “1”, the INT3 interrupt vector is called when the I23CR1 flag is set.

If reset to “0”, the interrupt processing routine is not called, also when an interrupt is generated.

Caution	• When I23CR7 and I23CR6, or I23CR3 and I23CR2 are both “0”, edge detection is not performed. When both are “1”, both edges are detected. • Input from P73 (INT3/T0IN pin) is routed through a noise filter.

4.2.4 Circuit Configuration and Operation Principles

Timer 0 mode setting

Mode	T0LONG	T0LEXT
0	0	0
1	0	1
2	1	0
3	1	1

Mode 0: 8-bit reload timer x 2 channels

In mode 0, timer 0 functions as an 8-bit reload timer with two channels. The relationship between the timer value and the reload register (T0LR) setting value is as shown below.

Time until T0HOVF is set (1) (decimal) = (256 - T0HR setting value) × TPR
Time until T0LOVF is set (1) (decimal) = (256 - T0LR setting value) × TPR

TPR: Prescaler clock cycle

When the count control bit (T0HRUN, T0LRUN) is set, counting starts. When it is reset, counting stops, and the contents of the reload register (T0HR, T0LR) are sent to the counter (T0H, T0L).

When the timer/counter 0 (T0H, T0L) overflows, the overflow flag (T0HOVF, T0LOVF) is set, and the contents of the reload register (T0HR, T0LR) are sent to the counter (T0H, T0L).

When both the overflow flag (T0HOVF, T0LOVF) and interrupt request enable flag (T0HIE, T0LIE) are set, the interrupt request is signalled to the interrupt control circuit.

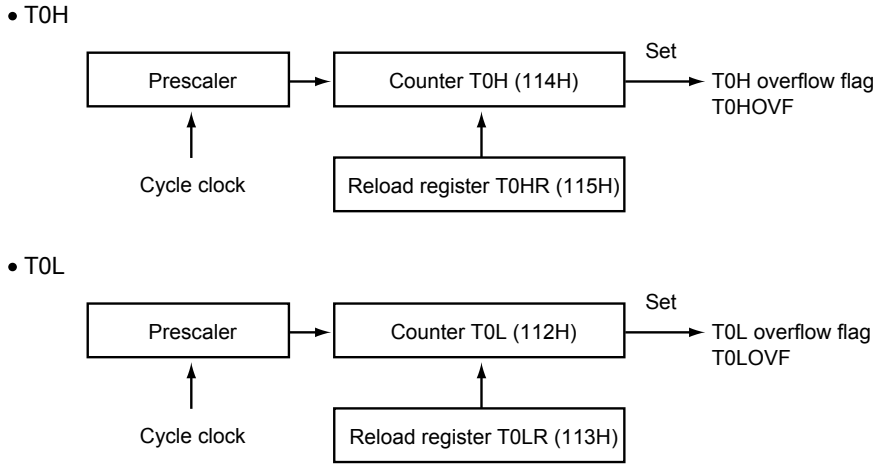


Fig. 4-6 Mode 0: 8-Bit Reload Timer × 2 Channel Circuit Configuration

Mode 0 sample program

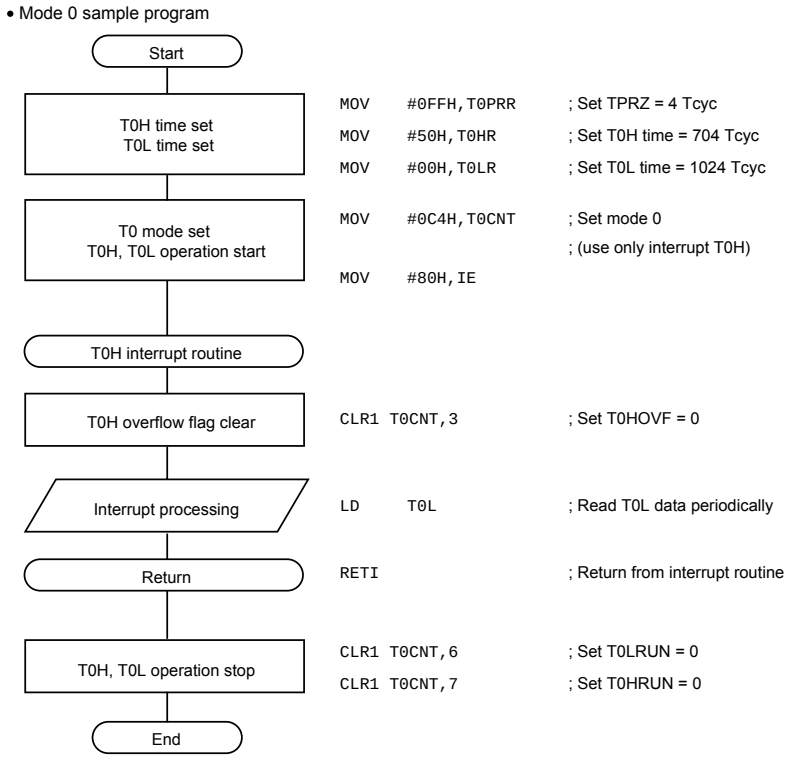


Fig. 4-7 Flow Chart and Program

Mode 1: 8-bit reload timer + 8-bit reload counter

8-bit reload timer

The upper 8 bits of timer 0 (T0H) operate as an 8-bit reload timer. The relationship between the timer value and the reload register (T0HR) setting value is as shown below.

Time until T0HOVF is set (1) (decimal) = (256 - T0HR setting value) × TPR

TPR: Prescaler clock cycle

Each time T0HOVF is set, the reload register value is sent to the counter T0H. Timer operation continues until the T0H count control bit (T0HRUN) is reset. Operation principles are the same as for mode 0.

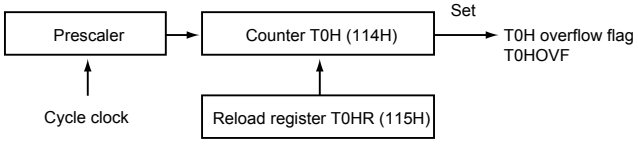


Fig. 4-8 Mode 1: 8-Bit Reload Timer (T0H) Block Diagram

8-bit reload counter

The lower 8 bits of timer 0 (T0L) are used to count up the external input signal. This signal is filtered by a noise filter circuit. For details, refer to the section "Input Signal Select Register (ISL)" in "4.2 Timer/Counter 0 (T0)".

The relationship between the count value and the reload register (T0LR) setting value is as shown below.

Time until T0LOVF is set (1) (decimal) = 256 - (T0LR setting value)

When the T0L overflow flag (T0LOVF) flag is set, the reload register (T0LR) value is sent to the counter (T0L). Timer operation continues until the T0H count control bit (T0LRUN) is reset.

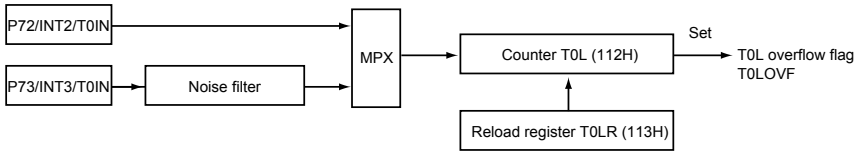


Fig. 4-9 Mode 1: 8-Bit Reload Timer (T0L) Block Diagram

Mode 1 sample program

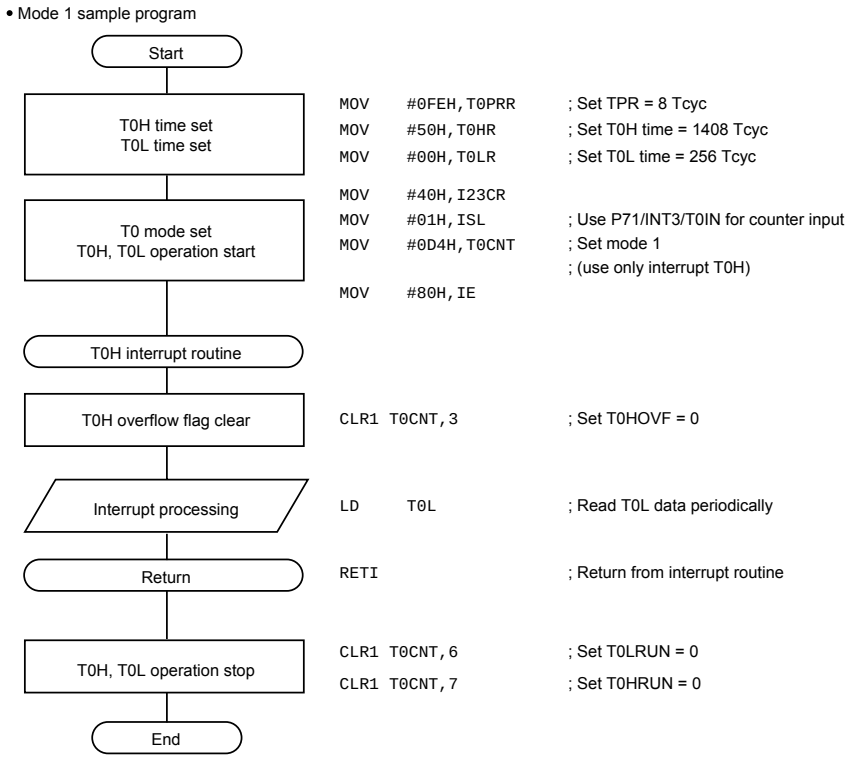


Fig. 4-10 Flow Chart and Program

Mode 2: 16-bit reload timer

In mode 2, T0H and T0L are connected in series and operate as a 16-bit timer.

To start the timer, the count control bits (T0HRUN, T0LRUN) of T0H and T0L must be set simultaneously.

The relationship between the timer value and the reload register (T0HR, T0LR) setting values is as shown below.

Time until T0HOVF is set (1) (decimal)
= (65536 - 256 × (T0HR setting value) - (T0LR setting value)) × TPR

TPR: Prescaler clock cycle

When T0LOVF and T0HOVF are both set, the reload register (T0HR, T0LR) values are sent simultaneously to T0L and T0H when T0HOVF occurs. Timer operation continues until the T0H count control bit is reset. Operation principles are the same as for mode 0.

For reading data from timer 0 (T0), use the following procedure.

```
T0L   LD    T0L          ; Read T0L data (1)
 ↓    ST    020H
T0H   LD    T0H          ; Read T0H data
 ↓    ST    021H
T0L   LD    T0L          ; Read T0L (2) data again
 ↓    BP    T0L,7,DES    ; When T0L (2) bit 7 is "0"
      BN    020H,7,DES   ; and T0L (1) bit 7 is "1"
      ST    020H
T0H   LD    T0H          ; Read T0H (2)
      ST    021H
      DES:  --  next program
```

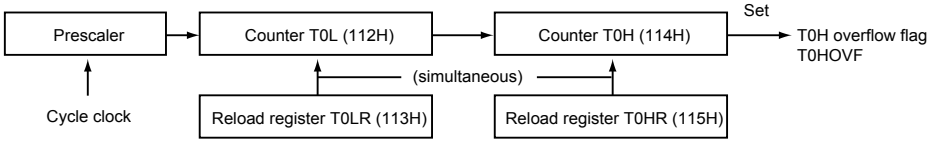


Fig. 4-11 Mode 2: 16-Bit Reload Timer Block Diagram

Mode 2 sample program

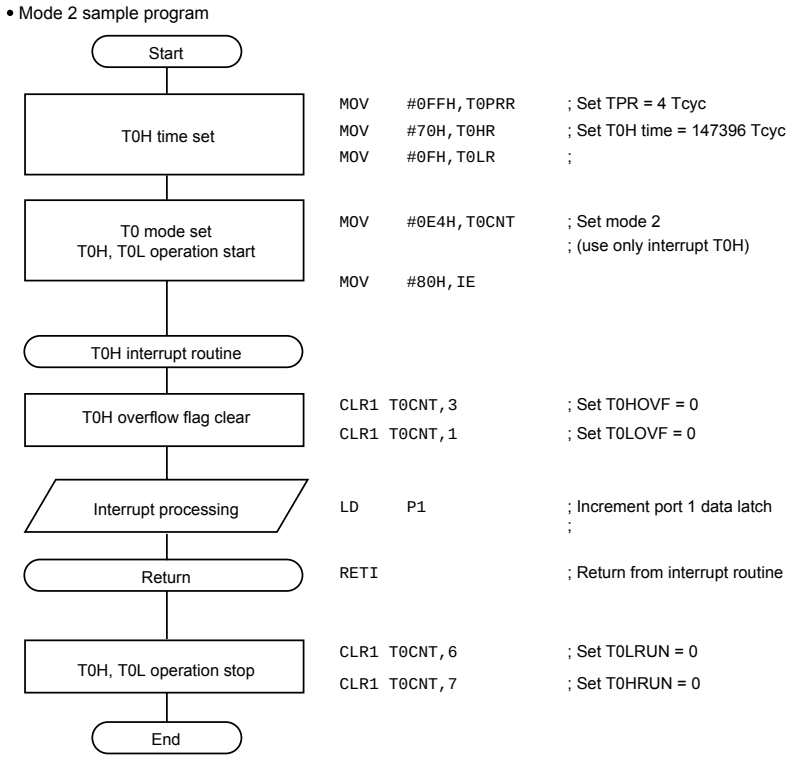


Fig. 4-12 Flow Chart and Program

Mode 3: 16-bit reload counter

In mode 3, T0H and T0L are connected in a cascaded configuration and operate as a 16-bit counter. The signal input to P72 (INT2/T0IN pin) or P73 (INT3/T0IN pin) is used as clock signal. Input signal selection is carried out by the ISL register of SFR. The input from P73 (INT3/T0IN pin) is routed through a noise filter.

To start the timer, the count control bits (T0HRUN, T0LRUN) of T0H and T0L must be set simultaneously.

The relationship between the count value and the reload register (T0HR, T0LR) setting values is as shown below.

Time until T0HOVF is set (1) (decimal) = 65536 - 256 × (T0HR setting value) - (T0LR setting value)

When T0LOVF and T0HOVF are both set, the reload data (T0HR, T0LR) are sent simultaneously to T0L and T0H when T0HOVF occurs. Timer operation continues until the count control bit is reset. Operation principles are the same as for mode 0.

For reading data from timer 0 (T0), use the following procedure.

```
T0L    LD     T0L          ; Read T0L data (1)
 ↓     ST     020H
T0H    LD     T0H          ; Read T0H data
 ↓     ST     021H
T0L    LD     T0L          ; Read T0L (2) data again
 ↓     BP     T0L,7,DES    ; When T0L (2) bit 7 is "0"
       BN     020H,7,DES   ; and T0L (1) bit 7 is "1"
       ST     020H
T0H    LD     T0H          ; Read T0H (2)
       ST     021H
       DES:   --  next program
```

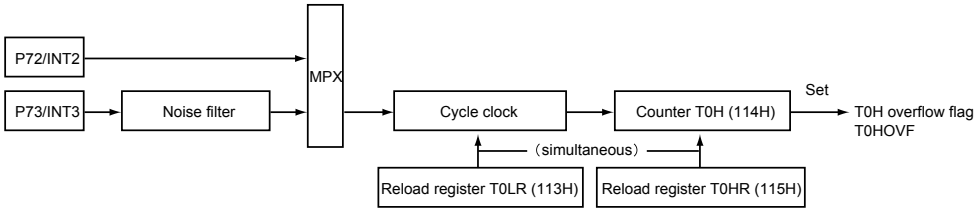


Fig. 4-13 Mode 3: 16-Bit Reload Timer Block Diagram

Mode 2 sample program

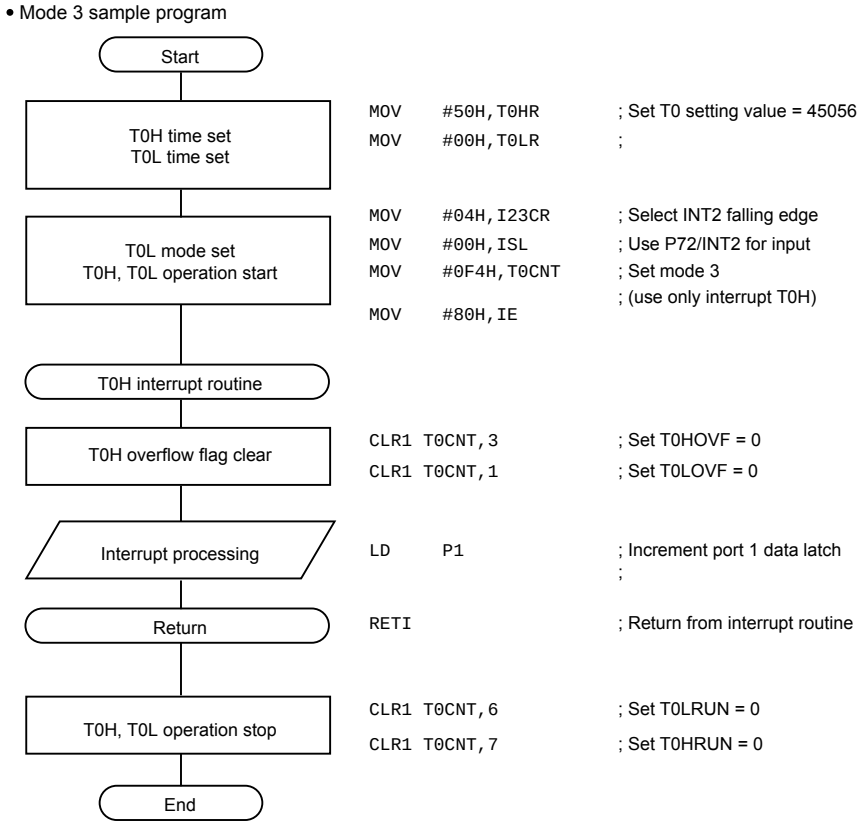


Fig. 4-14 Flow Chart and Program

4.3 Timer 1 (T1)

The timer/counter 1 (T1) in the VMS custom chip is a 16- bit timer with the following 4 functions.

- Mode 0: 8-bit reload timer × 2 channels
- Mode 1: 8-bit reload timer + 8-bit pulse generator
- Mode 2: 16-bit reload timer
- Mode 3: Variable bit length pulse generator (9 to 16 bits)

4.3.1 Functions

8-bit reload timer × 2 channels (mode 0)

The cycle clock is used to drive two separate 8-bit reload timers (T1H, T1L).

8-bit reload timer + 8-bit pulse generator (mode 1)

T1H operates as an 8-bit reload timer driven by the cycle clock. T1L operates as an 8-bit pulse generator whose output appears at the P17/pulse output pin.

16-bit reload timer (mode 2)

The overflow of T1L is used as clock for T1H, to drive the 16-bit reload timer. The input clock to T1L is the cycle clock. Each time a T1L overflow occurs, the T1LR and T1HR reload data are loaded into T1L and T1H.

The T1L clock can be the cycle clock or the cycle clock divided by 2.

Variable bit length pulse generator (9 to 16 bits) (mode 3)

T1L and T1H can be used to generate a 9 to 16 bit pulse signal. This signal is output via the P17/pulse output pin.

The T1L clock can be the cycle clock or the cycle clock divided by 2.

Interrupt generation

When the interrupt request enable bit is set, overflow of the register T1H or T1L generates a T1H or T1L interrupt request.

The following Special Function Registers must be operated to control the timer 1 (T1).

T1H, T1HR, T1HC, T1L, T1LR, T1LC, T1CNT, P1

4.3.2 Circuit Configuration

The timer/counter 1 (T1) configuration is shown in Fig. 4- 15.

Timer 1 low (T1L) ... ❷

This 8-bit reload timer uses the cycle clock or cycle clock divided by 2. When T1L overflow occurs, the T1LR value is reloaded. The T1LR value is reloaded when T1L overflow occurs. Resetting the T1LRUN (T1CNT bit 6) to '0' stops the timer and causes the T1LR data to be transferred to T1L.

Timer 1 low comparator (T1LC) ... ❵

This circuit consists of the 8-bit timer 1 low comparator data register (T1LC) and an 8-bit data comparator. It compares the data for T1L and T1LC.

Timer 1 high (T1H) ... ➥

This 8-bit reload timer uses the cycle clock or the T1L overflow as clock. At T1H overflow, the T1HR value is reloaded. Reload also occurs when T1HRUN (T1CNT bit 7) is reset to stop the timer.

Timer 1 high comparator (T1HC) ... ❸

This circuit consists of the 8-bit timer 1 high comparator data register (T1HC) and an 8-bit data comparator. It compares the data for T1H and T1HC.

Timer 1 control register (T1CNT) ... ❨

Serves for T1 mode setting and interrupt control.

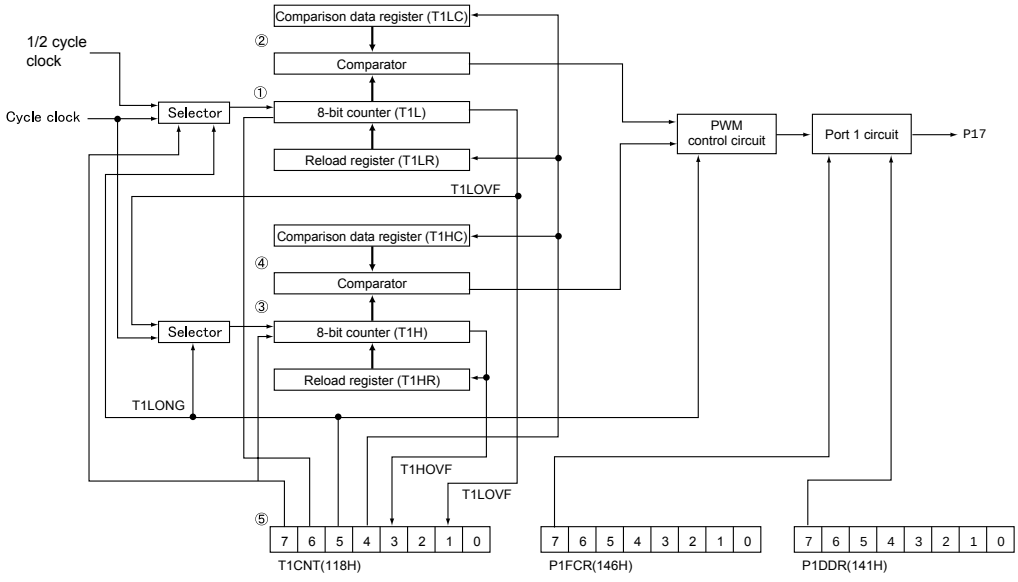


Fig. 4-15 Timer 1 Block Diagram

4.3.3 Related Registers

Timer 1 control register (T1CNT)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CNT	118H	R/W	T1HRUN	T1LRUN	T1LONG	ELDT1C	T1HOVF	T1HIE	T1LOVF	T1LIE
Reset			0	0	0	0	0	0	0	0

Bit name	Function
T1HRUN (bit 7)	T1H count control
	0: Count stop/data reload 1: Count start
T1LRUN (bit 6)	T1L count control
	0: Count stop/data reload 1: Count start
T1LONG (bit 5)	Timer 1 bit length selector
	0: 8 bit 1: 16 bit
ELDT1C (bit 4)	Pulse generator data update enabled
	0: Disabled 1: Enabled
T1HOVF (bit 3)	T1H overflow flag
	0: No overflow flag 1: Overflow flag
T1HIE (bit 2)	T1H interrupt request enabled
	0: Interrupt request disabled 1: Interrupt request enabled
T1LOVF (bit 1)	T1L overflow flag
	0: No overflow flag 1: Overflow flag
T1LIE (bit 0)	T1L interrupt request enabled
	0: Interrupt request disabled 1: Interrupt request enabled

T1HRUN (bit 7): T1H count control

Controls count-up start/stop of timer 1 high (T1H). When set to “1”, the T1H clock is supplied and counting starts. To stop counting, the bit must be reset to “0”. This stops the clock and sends the reload data (T1HR) to T1H.

T1LRUN (bit 6): T1L count control

Controls count-up start/stop of timer 1 low (T1L). When set to “1”, the T1L clock is supplied and counting starts. To stop counting, the bit must be reset to “0”. This stops the clock and sends the reload data (T1LR) to T1L.

T1LONG (bit 5): Timer 1 bit length select

Specifies the bit length of T1 as 16 or 8 bit.

Setting the bit to “1” selects a 16-bit timer. For using modes 0 and 1, specify “1”.

Resetting the bit to “0” selects an 8-bit timer. For using modes 2 and 3, specify “0”.

ELDT1C (bit 4): Pulse generator data update enable control

Controls whether the comparison data register (T1HC, T1LC) values for generating the pulse signal are sent to the comparator or not.

When set to “1”, the values are sent to the comparator and updated to new pulse generator data.

When reset to “0”, the data are not updated and the same pulse generator data are output.

To update both 8-bit counters at the same time, reset this flag, set the counter values, and then set the flag again. This will update both 8-bit counters at the same time.

T1HOVF (bit 3): T1H overflow flag

This flag is set when T1L overflow has occurred. If there is no overflow, the flag does not change.

This flag must be reset by the T1H interrupt processing routine or another routine of the application.

T1HIE (bit 2): T1H interrupt request enable control

Enables or disables interrupt request generation at T1H overflow.

When set to “1”, the interrupt generated by T1H overflow is accepted and the interrupt vector 002BH is called. When reset to “0”, the interrupt is not accepted and the interrupt processing routine is not called.

T1LOVF (bit 1): T1L overflow flag

This flag is set when T1L overflow has occurred. If there is no overflow, the flag does not change.

Regardless of the T1 bit length, the flag is set when overflow occurs at T1L.

This flag must be reset by the T1L interrupt processing routine or another routine of the application.

T1LIE (bit 0): T1L interrupt request enable control

Enables or disables interrupt request generation at T1L overflow.

When set to "1", the interrupt generated by T1L overflow is accepted and the interrupt vector 002BH is called.

Caution	• The overflow flags (T1HOVF, T1LOVF) must be reset to "0" by the application. • When using the 16-bit mode, the clock can be the cycle clock or the cycle clock divided by 2. Ttc = Tcyc: T1HRUN=1, T1LRUN=1, T1LONG=1 Ttc = 1/2Tcyc: T1HRUN=0, T1LRUN=1, T1LONG=1 Ttc is the clock cycle

Timer 1 low register (T1L)

The timer 1 low register is an 8-bit timer. It uses the cycle clock or the cycle clock divided by 2.

When T1L overflow occurs, the T1LR value is transferred and the T1L overflow flag is set.

In modes 1 and 3, this is used for pulse signal generation.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1L	11BH	R	T1L7	T1L6	T1L5	T1L4	T1L3	T1L2	T1L1	T1L0
Reset			0	0	0	0	0	0	0	0

Timer 1 low reload register (T1LR)

This is the reload register for timer 1 low (T1L).

Each time a T1L overflow occurs, and whenever T1LRUN=0 applies, the reload register value is loaded into T1L.

In modes 1 and 3, this is used for pulse signal generation.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LR	11BH	W	T1LR7	T1LR6	T1LR5	T1LR4	T1LR3	T1LR2	T1LR1	T1LR0
Reset			0	0	0	0	0	0	0	0

T1L and T1LR are at the same address. T1L is read-only, and T1LR is write-only.

Caution	When a bit operation instruction or the INC, DEC, or DBNZ instruction is used on a write-only register, a bit other than the specified bit will be set. For T1LR, use the following instructions: MOV, MOV @, ST, ST @, POP

Timer 1 low comparator data register (T1LC)

This is the comparator data register for timer 1 low (T1L).

When ELDT1C (bit 4 of T1CNT) is set and T1LONG=0 applies, the next T1L overflow will cause the value of this register to be sent to the pulse generator control circuit (comparator). When T1LONG=1 applies, the next T1H overflow will have the same effect.

When T1LRUN=0, the value of this register is always sent to the pulse generator control circuit.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LC	11AH	R/W	T1LC7	T1LC6	T1LC5	T1LC4	T1LC3	T1LC2	T1LC1	T1LC0
Reset			0	0	0	0	0	0	0	0

Timer 1 high register (T1H)

This is an 8-bit timer which uses the cycle clock or the T1L overflow (T1LOVF) as clock. At T1H overflow, the T1H overflow flag is set.

In mode 3, this is used for pulse signal generation.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1H	11DH	R	T1H7	T1H6	T1H5	T1H4	T1H3	T1H2	T1H1	T1H0
Reset			0	0	0	0	0	0	0	0

Timer 1 high reload register (T1HR)

This is the timer 1 high (T1H) reload register.

Each time a T1H overflow occurs, and whenever T1HRUN=0 applies, the reload register value is loaded into T1H.

In mode 3, this is used for pulse signal generation.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1HR	11DH	W	T1HR7	T1HR6	T1HR5	T1HR4	T1HR3	T1HR2	T1HR1	T1HR0
Reset			0	0	0	0	0	0	0	0

T1H and T1LH are at the same address. T1H is read-only, and T1LH is write-only.

Caution	When a bit operation instruction or the INC, DEC, or DBNZ instruction is used on a write-only register, a bit other than the specified bit will be set. For T1LR, use the following instructions: MOV, MOV @, ST, ST @, POP

Timer 1 high comparator data register (T1HC)

This is the comparator data register for timer 1 high (T1H).

When ELDT1C (bit 4 of T1CNT) is set and T1LONG=0 applies, the next T1L overflow will cause the value of this register to be sent to the pulse generator control circuit (comparator). When T1LONG=1 applies, the next T1H overflow will have the same effect.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1HC	11CH	R/W	T1HC7	T1HC6	T1HC5	T1HC4	T1HC3	T1HC2	T1HC1	T1HC0
Reset			0	0	0	0	0	0	0	0

4.3.4 Circuit Configuration and Operation Principles

Timer 1 mode setting

ModeClock frequencyT1LONGP17FCRP17DDRP17 0Tcyc00XX 1 Tcyc0110 2Tcyc _1/2Tcyc10XX 3Tcyc _1/2Tcyc11 10

Mode 0: 8-bit reload timer × 2 channels

In mode 0, timer 1 functions as an 8-bit reload timer with two channels. The relationship between the timer value and the reload register (T1LR) setting value is as shown below.

Time until T0HOVF is set (1) (decimal) = (256 - T1HR setting value) × Tcyc
Time until T0LOVF is set (1) (decimal) = (256 - T1LR setting value) × Tcyc

Tcyc: Cycle clock

When the count control bit (T1HRUN, T1LRUN) is set, counting starts. When it is reset, counting stops, and the contents of the reload register (T1HR, T1LR) are sent to the counter (T1H, T1L).

When the timer 1 (T1H, T1L) overflows, the overflow flag (T1HOVF, T1LOVF) is set, and the contents of the reload register (T1HR, T1LR) are sent to the counter (T1H, T1L).

When both the overflow flag (T1HOVF, T1LOVF) and interrupt request enable flag (T1HIE, T1LIE) are set, the interrupt request is signalled to the interrupt control circuit.

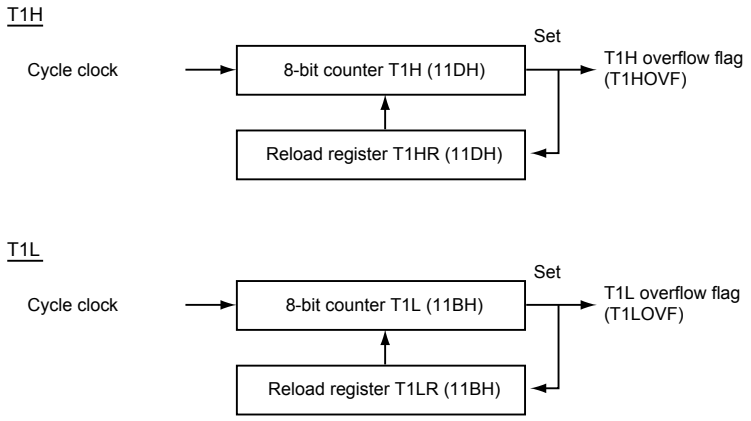


Fig. 4-16 8-Bit Reload Timer × 2 Channel Circuit Configuration

Mode 0 sample program

• Mode 0 sample program

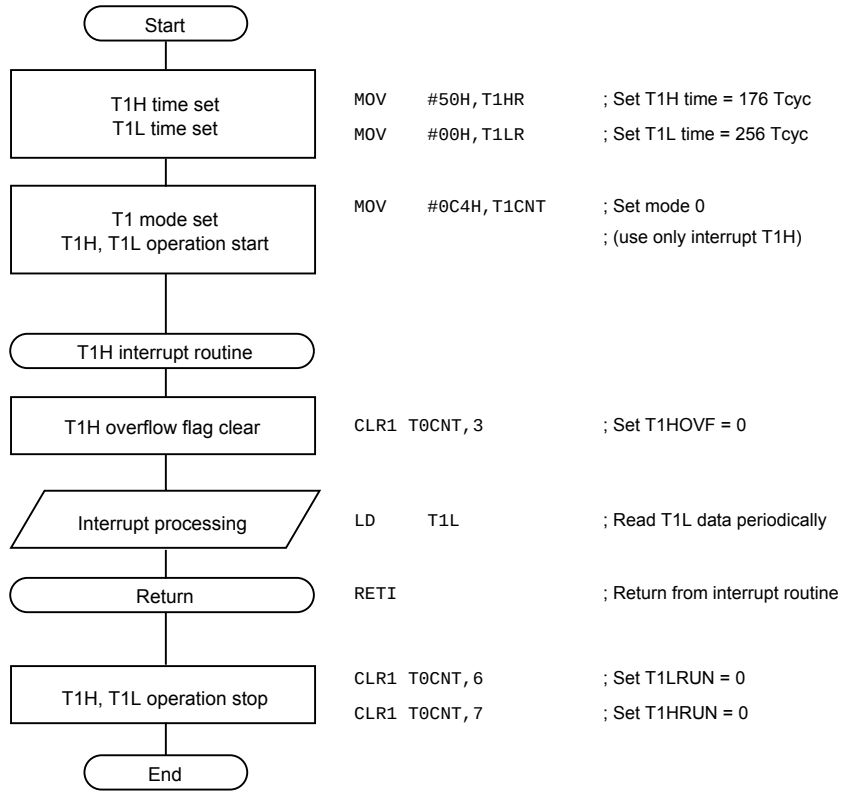


Fig. 4-17 Flow Chart and Program

Mode 1: 8-bit reload timer + 8-bit pulse generator

8-bit reload timer

The upper 8 bits of timer 0 (T1H) operate as an 8-bit reload timer. The relationship between the timer value and the reload register (T1HR) setting value is as shown below.

Time until T1HOVF is set (1) (decimal) = (256 - T1HR setting value) × Tcyc

Tcyc: Cycle clock

Each time T1HOVF is set, the reload register value is sent to the counter T1H. Timer operation continues until the T1H count control bit (T1HRUN) is reset. Operation principles are the same as for mode 0.

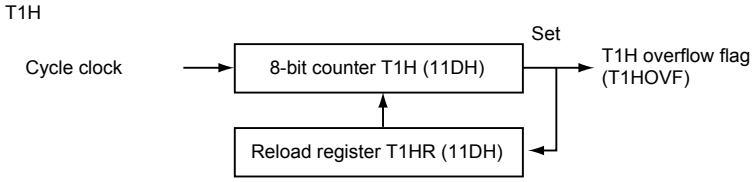


Fig. 4-18 Mode 1: 8-Bit Reload Timer (T1H) Block Diagram

8-bit pulse generator

The comparator compares the value of the T1L counted up from the reload value by the cycle clock to the value of the comparator data register T1LC. If there is no match (T1L ≠ T1LC), "0" is output. If there is a match (T1L = T1LC), "1" is output. This continues until T1L overflow is generated.

The pulse signal cycle is determined by the reload register T1LR. The relationship between the counter value and the pulse output waveform is shown in Fig. 4-19.

The pulse output waveform is determined by the comparison data register T1LC and the reload register T1LR. When the comparison data register T1LC is rewritten, there will be a pulse cycle delay until the pulse output waveform reflects the new data.

Whenever T1L overflows, the T1L overflow flag (T1LOVF) is set.

The pulse output signal related equations are shown below.

Pulse output signal "L" level pulse width (decimal) =
(T1LC setting value - T1LR setting value) × Tcyc
Pulse output signal cycle (decimal) = (256 - T1LR setting value) × Tcyc

Tcyc: Cycle clock

Caution	Make sure that T1LC ≥ T1LR applies

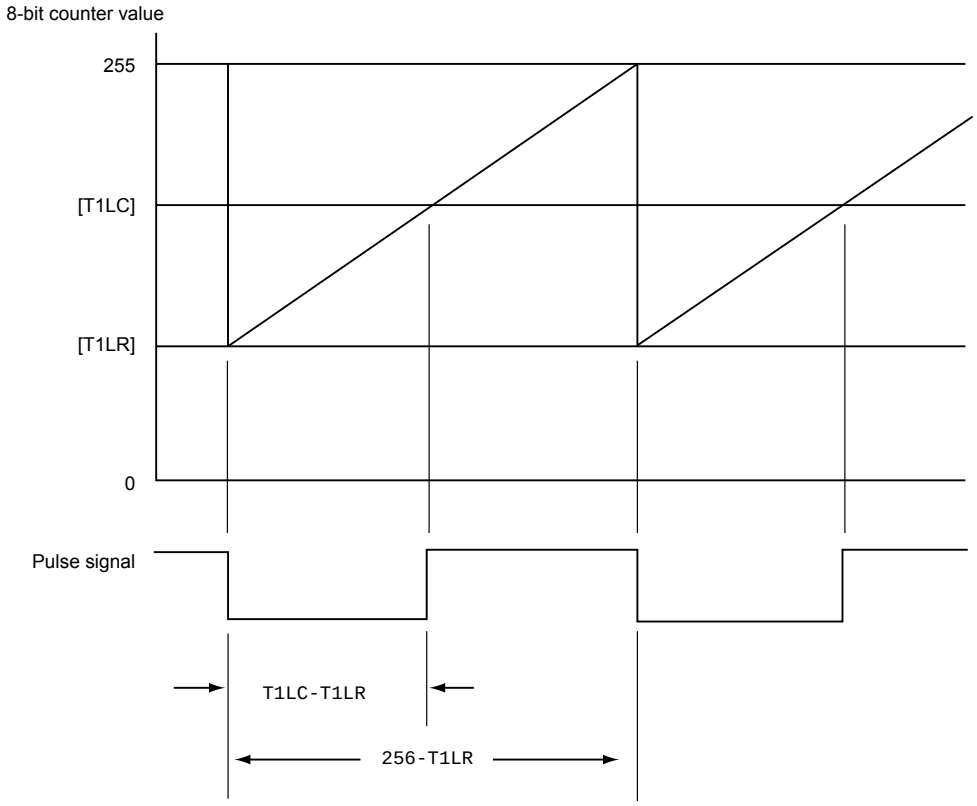


Fig. 4-19 Counter Value and Pulse Generator Output Waveform

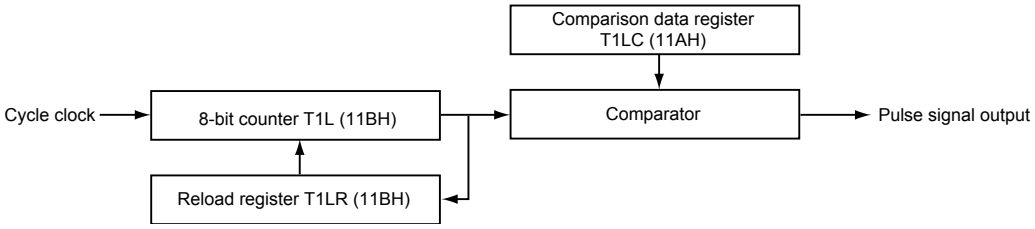


Fig. 4-20 Mode 1: 8-Bit Pulse Generator Block Diagram

Mode 1 sample program

• Mode 1 sample program

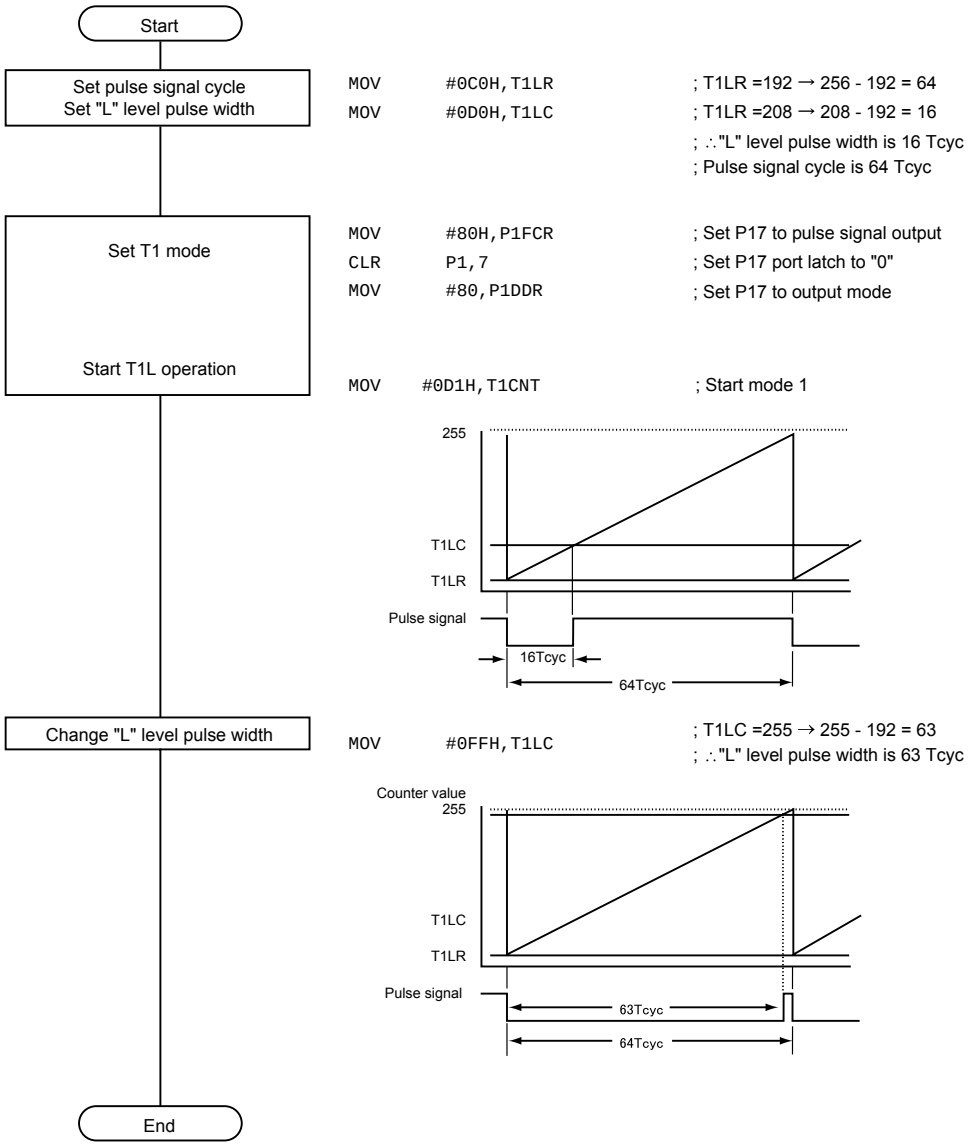


Fig. 4-21 Flow Chart and program

Mode 2: 16-bit reload timer

To operate the timer as 16-bit reload timer, T1LRUN and T1LONG must be set together. This should be done with the MOV instruction.

The T1L clock (Ttc) can be either the cycle clock (Tcyc) or half the cycle clock (1/2Tcyc). The setting is made as follows.

Ttc = Tcyc:	T1HRUN = 1, T1LRUN = 1, T1LONG = 1
Ttc = 1/2Tcyc:	T1HRUN = 0, T1LRUN = 1, T1LONG = 1

The relationship between the timer value and the reload register (T1HR, T1LR) is as shown below.

Caution	Note that this is different from the timer/counter 0 (T0).

Time until T1HOVF is set (1) (decimal) =
(256 - T1HR setting value) × (256 - T1LR setting value) × Ttc
Time until T1LOVF is set (1) (decimal) = (256 - T1LR setting value) × Ttc

Ttc: T1L clock (Tcyc or 1/2 Tcyc)

At each T1LOVF, the reload data (T1LR) are sent to T1L, and at each T1HOVF, the reload data (T1HR) are sent to T1H. Counting continues until the count control bit is reset. Operation principles are the same as for mode 0.

For reading data from timer 1 (T1), use the following procedure.

```
T1L   LD    T1L          ; Read T1L data (1)
 ↓    ST    020H
T1H   LD    T1H          ; Read T1H data
 ↓    ST    021H
T1L   LD    T1L          ; Read T1L (2) data again
 ↓    BP    T1L,7,DES    ; When T1L (2) bit 7 is "0"
      BN    020H,7,DES   ; and T1L (1) bit 7 is "1"
      ST    020H
T1H   LD    T1H          ; Read T1H (2)
      ST    021H
      DES:  - next program
```

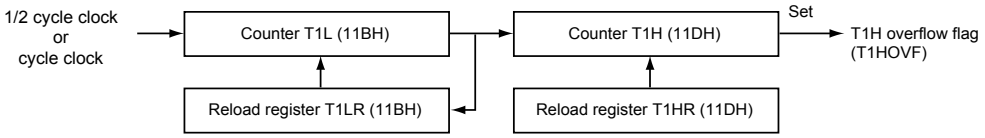


Fig. 4-22 Mode 2: 16-Bit Reload Timer Block Diagram

Mode 2 sample program

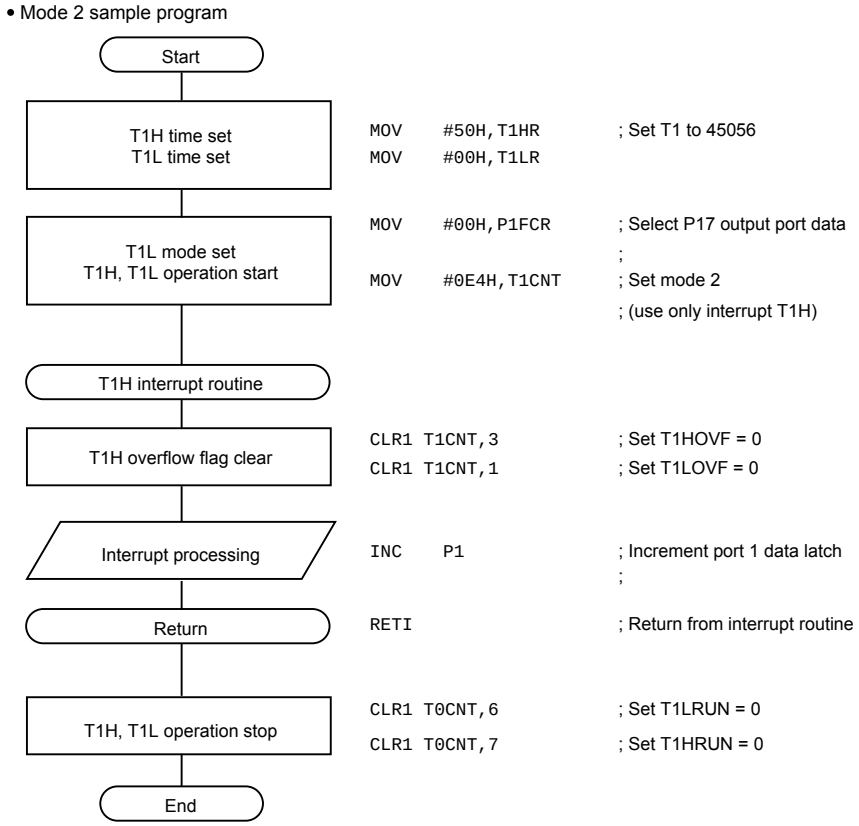


Fig. 4-23 Flow Chart and Program

Mode 3: Variable bit length pulse generator (9 to 16 bits)

In mode 3, timer 1 (T1L, T1H) operates as a variable bit length pulse generator. The range of 9 to 16 bits is determined by T1HR.

To activate the pulse generator, set the bit length of timer 1 to 16 (T1LONG=1) and set the T1L count control bit (T1LRUN). When the 16-bit length is selected, the control bit T1LRUN controls start/stop of all 16 bits. To set the timer 1 control register (T1CNT) bit at the same time, use the MOV instruction.

The pulse generator clock (Ttc) can be either the cycle clock (Tcyc) or half the cycle clock (1/2Tcyc). The setting is made as follows.

Ttc = Tcyc:	T1HRUN = 1, T1LRUN = 1, T1LONG = 1
Ttc = 1/2Tcyc:	T1HRUN = 0, T1LRUN = 1, T1LONG = 1

Whenever T1L overflows, the T1L overflow flag (T1LOVF) is set. Similarly, whenever T1H overflows, the T1H overflow flag (T1HOVF) is set. Counting continues until the count control bit is reset.

The relationship between the timer value and the reload register (T1Hr, T1LR) values is as shown below.

Time until T1HOVF is set (1) (decimal) =
(256 - T1HR setting value) × (256 - T1LR setting value) × Ttc
Time until T1LOVF is set (1) (decimal) = (256 - T1LR setting value) × Ttc

Ttc: T1L clock (Tcyc pr 1/2 Tcyc)

An example for a signal output from the pulse output pin P17 in mode 3 is shown in Fig. 4-24.

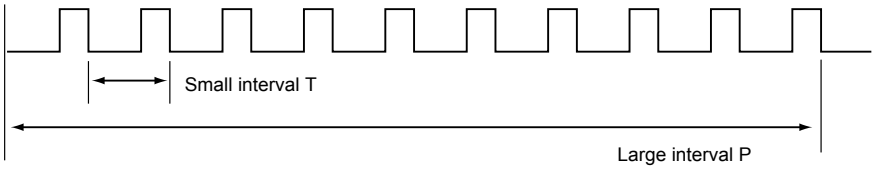


Fig. 4-24 Mode 3 Pulse Signal Output Waveform

The output signal consists of a repetition of the large interval P which is made up of upt o 256 repetitions of the small interval T. The number of repetitions for T can be set with T1HR. The "L" level width in the small interval T can be set with T1LC as in mode 1, and the smallest unit is Ttc. The total "L" level width <sigma>TL of the large interval P can be set with T1LC and T1HC. The T1HR value limits the data that can be obtained by T1HC.

For details on the relationship between the output waveform and T1HC and T1LC, refer to chapter 17 "Variable Bit Length Pulse Generator" in the appendix.

The relationship between the pulse generator bit length and the value of T1LR and T1HR, as well as the value of T1LC and T1HC is shown below. All T1LR bits are to be set to 00H.

Table 4-1 Bit Length and T1H/T1L Register

Pulse	Pulse bit length setting (binary)		"L" level pulse width setting (binary)	
Bit length	T1HR value	T1HL value	T1LC value (upper bit)	T1HC value (lower bit)
16	0000 0000	0000 0000	XXXX XXXX	XXXX XXXX
15	1000 0000	0000 0000	XXXX XXXX	XXXX XXX0
14	1100 0000	0000 0000	XXXX XXXX	XXXX XX00
13	1110 0000	0000 0000	XXXX XXXX	XXXX X000
12	1111 0000	0000 0000	XXXX XXXX	XXXX 0000
11	1111 1000	0000 0000	XXXX XXXX	XXX0 0000
10	1111 1100	0000 0000	XXXX XXXX	XX00 0000
9	1111 1110	0000 0000	XXXX XXXX	X000 0000

(X: 0 or 1) indicates effective bits

For example, if the bit length is 16 bits, the large interval P contains 256 small intervals T, and the following applies:

$TP = 256 \times T$

Because the small interval T is 256 × TTc (cycle clock or 1/2 cycle clock), the following applies:

TP = 256 × 256 × Ttc = 65536 × Ttc

The total “L” level additional pulse width ΣTL of the large interval P is set with T1HC.

ΣTL+ = [T1HC] × Ttc

Because the “L” level width of the small interval T can be set with T1LC, the total “L” level interval width ΣTL is calculated as follows.

ΣTL = (256 = mT1LC] + [T1HC]) × Ttc

When T1LC = 03H, T1HC = 0B4H

ΣTL = (256 x 03+180) × Tcyc = 948 × Ttc

The “L” level ratio RL is calculated as follows.

RL = ΣTL/TP = 948/65536 = approx. 1.447%

When T1LC = 0FFH, T1HC = 0FFH, the “L” level ratio RL is calculated as follows.

RL = ΣTL/TP = 65535/65536 = approx. 99.998%

The relationship between pulse bit length and settable pulse width is as follows.

Large interval P cycle TP

TP = 2[bit] × Ttc

Total "L" level pulse width ΣTL in large interval P

$$\Sigma TL = (2[bit] = mT1LC]/256+[T1HC]) \times Ttc$$

Note	T1HC and T1LC are decimal values. [T1HC] is the effective bit value.

Table 4-2 Bit Length and Pulse Width and Precision

Bit length	T1LC		T1HC		_TL		TP[Ttc]	Precision
	min.	max.	min.	max.	min.	max.		
16	0	255	0	255	0	65535	65535	1/65535
15	0	255	0	127	0	32767	32767	1/32767
14	0	255	0	63	0	16383	16383	1/16383
13	0	255	0	31	0	8191	8191	1/8191
12	0	255	0	15	0	4095	4095	1/4095
11	0	255	0	7	0	2047	2047	1/2047
10	0	255	0	3	0	1023	1023	1/1023
9	0	255	0	1	0	511	511	1/511

Note	T1HC represents the value for effective bits indicated in the table. For example, with a bit length of 11, bits 7 through 5 are effective. The maximum value therefore is 7.

Example: Setting values (binary) for use as 14-bit pulse generator

- T1HR value: 1100 0000B
- T1LR value: 0000 0000B
- Pulse generator 14-bit setting value

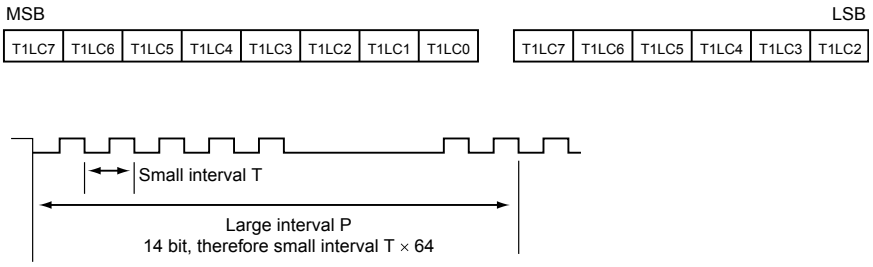


Fig. 4-25 14-Bit Pulse Generator

In the small interval T, two types of pulses are output. In the large interval P, pulse (1) is (64-T1HC) times, and pulse (2) is output T1HC times.

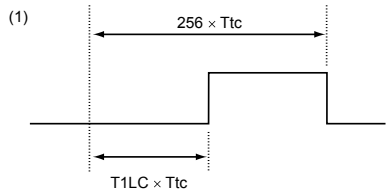


Fig. 4-26 T1LC x Ttc pulse

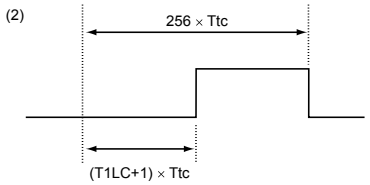


Fig. 4-27 (T1LC +1) x Ttc pulse

For details on the relationship between the output waveform and T1HC and T1LC, refer to chapter 17 "Variable Bit Length Pulse Generator" in the appendix.

Caution	• To set the "L" level pulse width, use the following procedure. (1) Reset data update enable flag ELDT1C to "0". (2) Rewrite T1LC and T1HC values. (3) Set data update enable flag ELDT1C to "1". • The delay between rewriting T1LC and T1HC and the waveform output based on the new data is equal to the interval between setting ELDT1C to "1" and the maximum pulse cycle. • For using 16-bit mode, the clock can be either the cycle clock or the cycle clock divided by 2. Ttc = Tcyc: T1HRUN=1, T1LRUN=1, T1LONG=1 Ttc = 1/2Tcyc: T1HRUN=0, T1LRUN=1, T1LONG=1

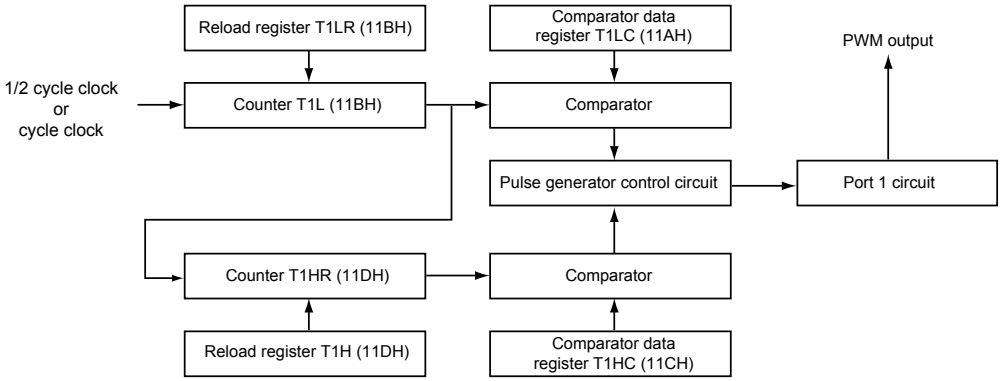


Fig. 4-28 Mode 3: Variable Bit Length Pulse Generator Block Diagram

Mode 3 sample program

• Mode 3 sample program

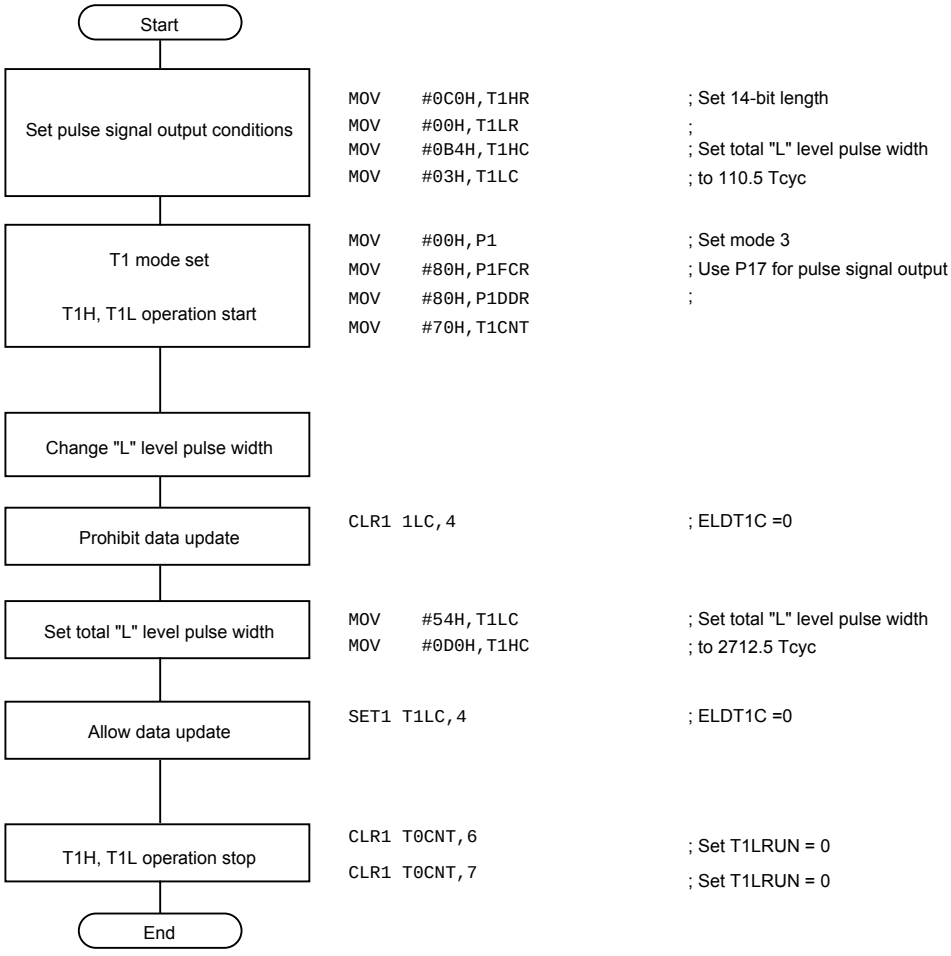


Fig. 4-29 Flow Chart and Program

4.4 Base Timer

The base timer in the VMS custom chip is a 14-bit binary up counter with the following 4 functions.

Caution	The clock function of the VMS is implemented by counting the interrupts generated in 0.5 second intervals by the base timer. The port 3 interrupt is a level interrupt which is maintained for as long as the user presses a button. If another timer is used to frequently generate interrupts or to accept the port 3 level interrupt, the internal clock may run slow. When using the base timer interrupt, call the user-side handler immediately after the label timer_ex_exit in GHEAD.ASM. The user-side handler must be designed to keep processing time at a minimum, so that the interrupt can be properly processed every 0.5 seconds. Care must be taken to prevent clock slow-down already when designing an application.

- Clock timer
- 14-bit binary up counter
- Fast-forward mode (using 6-bit base timer)

4.4.1 Functions

Clock timer

When the 32.768 kHz quartz oscillator is used as count clock for the base timer, a clock with 0.5 second steps can be implemented. The base timer count clock is a quartz oscillator.

14-bit binary up counter

By using the 8-bit binary up counter and 6-bit binary up counter in conjunction, a 14-bit binary up counter can be configured. The counters can be cleared by the application.

Fast-forward mode (using 6-bit base timer)

When the 6-bit timer is used as base timer, and the 32.768 kHz quartz oscillator is used as count clock, a clock with 2 millisecond steps can be implemented. Bit length switching is performed by the base timer control register (BTCR).

Interrupt generation

When the interrupt request enable bit is set, an interrupt request generated by the base timer will call the register vector 001BH. There are two types of interrupt requests

that can be generated by the base timer, referred to as base timer interrupt 0 and base timer interrupt 1.

The following Special Function Registers must be operated to control the base timer.

BTCR, P1, timer 0 functions, interrupt functions

4.4.2 Circuit Configuration

The base counter configuration is shown in Fig. 4-30.

8-bit binary up counter ... ❷

This is an up counter whose input is selected by the input signal select register (ISL). It generates a 4 kHz/2 kHz buzzer output signal. When the counter overflows, it generates a base timer interrupt 1 source. The overflow becomes the clock for the 6-bit binary counter.

6-bit binary up counter ... ❶

This is a 6-bit up counter which uses either the signal selected by the input signal select register (ISL) or 8- bit counter overflow as input. When the counter overflows, it generates a base timer interrupt 0 and 1 source. Input clock switching is performed by the base timer control register BCTR.

Base timer input clock source ... ➥

The quartz oscillator should be selected by the input signal select register (ISL) as base timer clock. Do not use other oscillators.

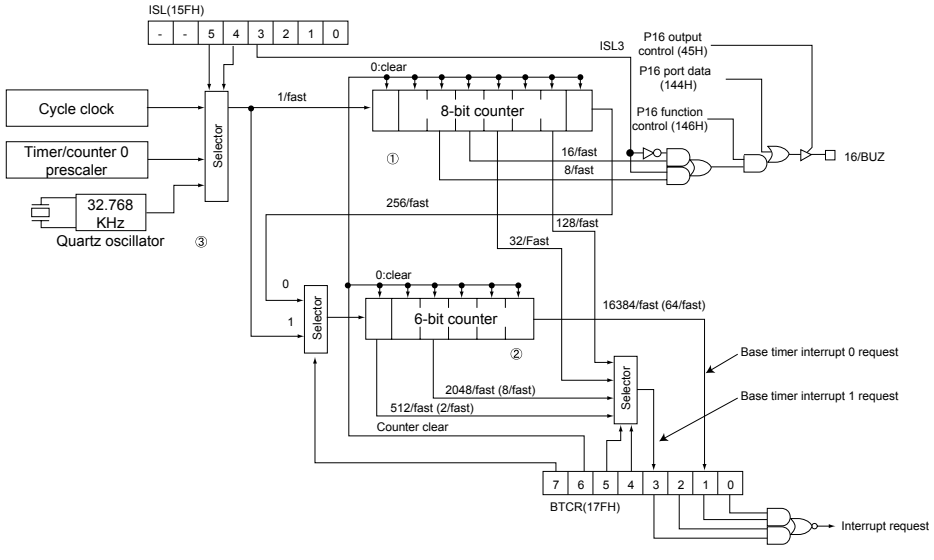


Fig. 4-30 Base timer block diagram

4.4.3 Related Registers

Base timer control register (BTCR)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BTCR	17FH	R/W	BTCR7	BTCR6	BTCR5	BTCR4	BTCR3	BTCR2	BTCR1	BTCR0
Reset			0	0	0	0	0	0	0	0

Caution	BTCR7, BTCR6, BTCR0 may not be manipulated by an application. To operate other bits, be sure to use bit level instructions.

Bit name	Function			
BTCR7 (bit 7)	Base timer interrupt 0 cycle control			
	0: 16384/fBST			
BTCR6 (bit 6)	Base timer operation control			
	1: base timer start			
BTCR5 (bit 5) BTCR4 (bit 4)	Base timer interrupt 1 cycle control			
	BTCR7	BTCR5	BTCR4	
	X	0	0	32/fBST
	X	0	1	128/fBST
	0	1	0	512/fBST
	0	1	1	2048/fBST
BTCR3 (bit 3)	Base timer interrupt 1 source			
	0: Interrupt source disabled 1: Interrupt source enabled			
BTCR2 (bit 2)	Base timer interrupt 1 request enabled			
	0: Interrupt request disabled 1: Interrupt request enabled			
BTCR1 (bit 1)	Base timer interrupt 0 source			
	0: Interrupt source disabled 1: Interrupt source enabled			
BTCR0 (bit 0)	Base timer interrupt 0 request enabled			
	1: Interrupt request enabled			

BTCR7 (bit 7): base timer interrupt 0 cycle control 0: fixed

Specifies the cycle for base timer interrupt 0 source generation.

When set to “0”, the cycle is 16384/fBST. In this case, the interval at which the interrupt 0 source is generated for 14-bit counter overflow is 16384/fBST.

When reset to “1”, the cycle is 64/fBST. To use the fast- forward mode, set this flag.

Caution	Because the base timer is used for the clock of the VMS, these registers may never be manipulated by an application.

BTCR6 (bit 6): base timer operation control 1: fixed

Starts or stops the base timer count operation.

When set to “1”, the count operation starts.

When reset to “0”, the count operation stops, and the 14- bit counter is cleared.

Caution	Because the base timer is used for the clock of the VMS, these registers may never be manipulated by an application.

BTCR5 - BTCR4 (bits 5 - 4): base timer interrupt 1 cycle control

Select the cycle for base timer interrupt 1 source generation.

BTCR7	BTCR5	BTCR4	Base timer interrupt 1 cycle
x	0	0	32/fBST
x	0	1	128/fBST
0	1	0	512/fBST
0	1	1	2048/fBST

fBST: Input clock frequency

BTCR3 (bit 3): base timer interrupt 1 source flag

This flag is set whenever the base timer interrupt 1 source is generated at the cycle set with BTCR7, BTCR5, and BTCR4. When no interrupt is generated, the flag does not change.

Caution	This flag must be reset by a suitable interrupt processing routine.

BTCR2 (bit 2): base timer interrupt 1 request enable control

Enables or disables the base timer interrupt 1 request.

When set to “1”, the base timer interrupt 1 source will generate an interrupt request to interrupt vector 001BH.

When reset to “0”, no interrupt request is generated.

BTCR1 (bit 1): base timer interrupt 0 source flag

This flag is set whenever the base timer interrupt 0 source is generated at the cycle set with BTCR7. When no interrupt is generated, the flag does not change.

Caution	This flag must be reset by a suitable interrupt processing routine.

BTCR0 (bit 0): base timer interrupt 0 request enable control 0: fixed

Enables or disables the base timer interrupt 0 request.

When set to “1”, the base timer interrupt 0 source will call the interrupt vector 001BH.

When reset to “0”, no interrupt request is generated.

Caution	Because the base timer is used for the clock of the VMS, these registers may never be manipulated by an application.

Caution	• In fast-forward mode (BTCR7, BTCR5 = 1), do not set both the system clock and the base timer to the quartz oscillator. • BTCR may occasionally become “1” when BTCR5 and BTCR4 are changed. This is a rare occurrence, but to guard against it, you should save the value of BTCR3 before changing BTCR5 and BTCR4 and then set the value again in BTCR3 after the change.

Input Signal Select Register (ISL)

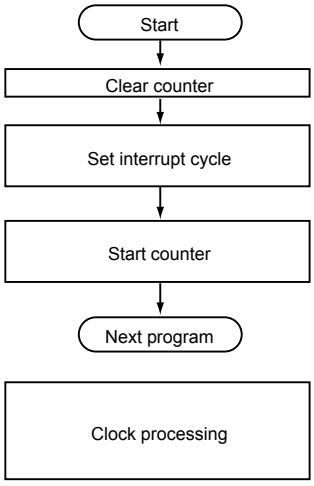
For details, refer to the section “Input Signal Select Register (ISL)” in “4.2 Timer/Counter 0 (T0)”.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ISL	15FH	R/W	–	–	ISL5	ISL4	ISL3	ISL2	ISL1	ISL0
Reset			0	0	0	0	0	0	0	0

Bit name	Function		
ISL5 (bit 5) ISL4 (bit 4)	Base timer clock select		
	ISL5	ISL4	
	X	0	Fixed to quartz oscillator
ISL3 (bit 3)	Use prohibited		
	0: fBST/16 (fixed)		
ISL2 (bit 2) ISL1 (bit 1)	Noise filter time constant select		
	ISL2	ISL1	Time constant
	1	1	16Tcyc
	0	1	64 Tcyc
	X	0	1 Tcyc
ISL0 (bit 0)	T0 clock input pin select		
	0: P72/INT2/T0IN pin 1: P73/INT3/T0IN pin		

4.4.4 Using the Base Timer

• Clock timer



	Related register	Sample program		
Clear counter	BTCR	CLR1	BTCR,6	; Stop base timer count
Set interrupt cycle	BTCR	CLR1	BTCR,7	; Select 14-bit base timer mode
	ISL	CLR1	ISL,4	; Set base timer clock to sub clock
				;
Start counter	BTCR	CLR1	BTCR,0	; Enable base timer interrupt 0
		SET	BTCR,6	; Start base timer count
Next program	Interrupt is generated here			
Clock processing	Interrupt processing routine	CLR1	BTCR,1	; Base timer interrupt
				; Set flag to 0
		RETI		; Return to main routine

Fig. 4-31 Clock timer

4.5 Serial Interface

The custom chip of the VMS incorporates a 2-channel synchronous serial interface with a data word length of 8 bits. It uses port 1 and allows two VMS units to communicate directly. When connected to the Dreamcast, the interface is automatically switched to the dedicated Dreamcast interface.

Reference	For information on how to ensure problem-free serial communication, refer to chapter 21 "Serial Communication Precautions" in the appendix.

Caution	Because the dedicated Dreamcast interface (Maple bus mode) also uses port 1, it cannot be used at the same time as the synchronous serial interface. When using the synchronous serial interface, an application must prohibit activation of the dedicated Dreamcast interface.

The serial interface has the following main functions and features.

- 2-channel synchronous serial interface
- Selectable transfer clock
- Serial interface SIO0 transfer clock with switchable polarity
- LSB/MSB switchable start sequence
- Switchable operation modes
- Overrun detection
- Transfer bit length control

4.5.1 Functions and Features

2-channel synchronous serial interface

Two serial interface channels are provided: SIO0 using P10 to P12 and SIO1 using P13 to P15.

Normally, the VMS uses SIO0 as master and SIO1 as slave.

Selectable transfer clock

The following three clock types can be selected. For SIO0, the transfer clock polarity can also be selected.

- Internal clock
- External clock
- Software clock

Serial interface SIO0 transfer clock with switchable polarity

The polarity of the transfer clock for the serial interface SIO0 can be selected as follows.

1. Operation stop, SCK0 = “1” ; data output hold
2. Operation stop, SCK0 = “0” ; data output bit 0 of SBUF0

LSB/MSB switchable start sequence

Data transfer via the serial interface can start either with the LSB or the MSB. This setting can be made individually for each channel.

Overrun detection

An error is generated when a clock exceeding 8 bits is received.

Transfer bit length control

A setting is available to control whether operation stops or continues after 8 bits have been transferred.

Interrupt

The following Special Function Registers must be operated to control the serial interface.

When the interrupt request enable bit is set, overflow of the octal counter generates an SIO0 or SIO1 interrupt request.

SCON0, SCON1, SBR, SBUF0, SBUF1, P1, P1DDR, P1FCR

Caution

When data transfer via the serial interface has been carried out, observe the following points.

(1) Do not make any settings for serial transfer while no other VMS unit is connected. When transfer is completed, be sure to make the settings listed in (3).
To make serial transfer settings, check the status of port 7 to verify whether another VMS unit is connected.

(2) When connection of another VMS unit has been verified, make serial transfer settings.
Use port 7 to verify whether another VMS unit is connected.

	P70	P72	P73
Connected to VMS	L	L	L
Not connected to VMS	L	L	L

(3) At the end of serial transfer, and when no other VMS unit is connected, establish the following settings.
SCON0 = 00H
SCON1 = 00H
P1FCR = 0BFH
P1DDR = 0A4H
If serial transfer settings are made while no other VMS unit is connected, normal operation is not assured.

4.5.2 Circuit Configuration

The serial interface configuration is shown in Fig. 4-32 and Fig. 4-33.

Shift register ... ❷

This component consists of two 8-bit shift registers (SBUF0, SBUF1) to operate the specified clock.

Octal counter ... ◗

Counts the shift clock and detects transfer end.

Baud rate generator ... ➥

Consists of an 8-bit register (SBR) for data settings and an 8-bit reload counter. When "internal clock" is selected as transfer clock, the clock data created here are transferred. The baud rate generator is used both by SIO0 and SIO1.

Reference	For more information on the internal clock, refer to section 4.5.6 "Serial Transfer Clock".

Polarity switcher

Controls the transfer clock polarity before and after serial transfer.

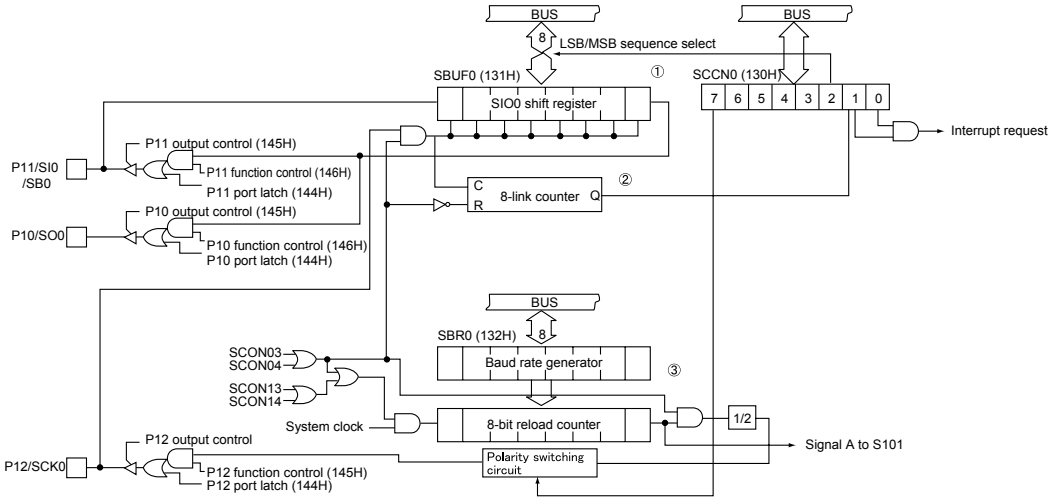


Fig. 4-32 Serial interface (SIO0) Block Diagram

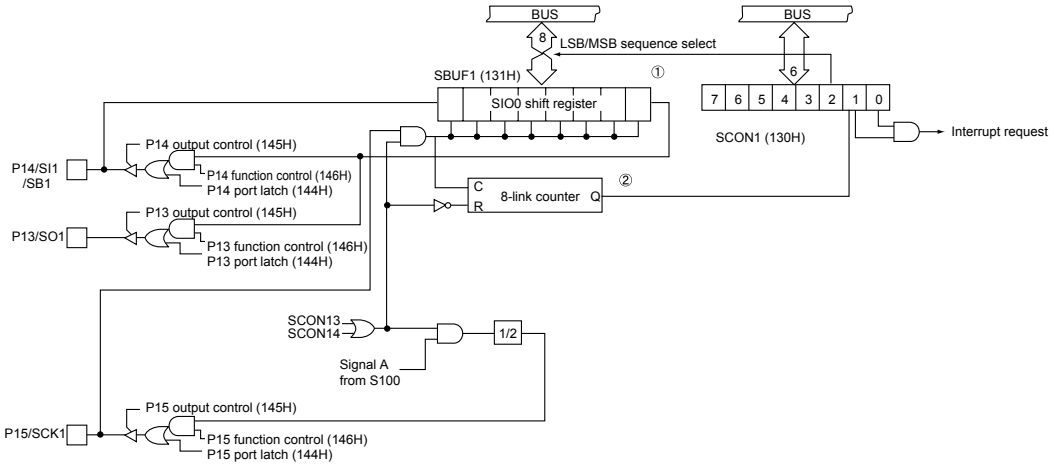


Fig. 4-33 Serial interface (SIO1) Block Diagram

4.5.3 Related Registers

SIO0 control register (SCON0)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCON0	130H	R/W	SCON07	SCON06	–	SCON04	SCON03	SCON02	SCON01	SCON00
Reset			0	0	H	0	0	0	0	0

Bit name	Function
SCON07 (bit 7)	Polarity control
	0: at operation stop, SCK0 = 1, maintain data output 1: at operation stop, SCK0 = 0, output data is bit 0 of SBUF0
SCON06 (bit 6)	Overrun flag
	0: No overrun 1: Overrun
SCON04 (bit 4)	Transfer bit length control
	0: 8-bit transfer 1: Continuous transfer
SCON03 (bit 3)	Transfer control
	0: Stop transfer 1: Start transfer
SCON02 (bit 2)	LSB/MSB sequence select
	0: LSB first 1: MSB first
SCON01 (bit 1)	Serial transfer end flag
	0: Transfer in progress 1: Transfer end
SCON00 (bit 0)	Interrupt request enabled
	0: Interrupt request disabled 1: Interrupt request enabled

SCON07 (bit 7): SCK0 polarity control

Controls the polarity of transfer clock SCK0 used by SIO0.

When set to “1”, SCK0 is “0” when SIO0 operation stops, and bit 0 of SBUF0 is output.

When reset to “0”, SCK0 is “1” when SIO0 operation stops, and the last transferred data is held at the output.

SCON06 (bit 6): overrun flag

Detects serial transfer errors in SIO0.

When an 8-bit data transfer is completed (SCON01 has become “1”) and the transfer clock is received (falling edge was detected), the flag is set.

During continuous transfer, the overflow flag is set every 8 bits.

Caution	This flag is not reset automatically. It must be reset by the application.

SCON04 (bit 4): transfer bit length control

Switches the SIO0 transfer data bit length to 8 bit continuous (1) or 8 bit (0).

When set to “1”, data of 2 byte or more can be sent continuously in 8-bit units.

When reset to “0”, only 8 bits of data (1 byte) can be sent.

This flag is not reset after transfer. It must be reset by the application.

SCON03 (bit 3): SIO0 operation control

Starts or stops SIO0 transfer.

When set to “1”, 8-bit serial transfer at SIO0 starts. When 8 bits have been transferred, the flag is reset.

When reset to “0”, serial transfer at SIO0 stops.

SCON02 (bit 2): LSB/MSB start select

Selects whether data are transferred starting with the MSB or LSB.

When set to “1”, the transfer starts with the MSB.

When reset to “0”, the transfer starts with the LSB.

Caution	This flag applies both to sending and receiving data. The setting must match at both ends.

SCON01 (bit 1): SIO0 transfer end flag

Detects the end of serial transfer.

The flag is set when a serial transfer of 8 bits is completed.

When this flag is set, and a falling edge of the transfer clock is detected, the overrun flag is set.

Caution	This flag is not reset automatically. It must be reset by the application.

SCON00 (bit 0): SIO0 interrupt request enable control

Enables or disables interrupt request generation at SIO0 transfer end.

When set to “1”, the interrupt vector 0033H is called when SIO0 transfer ends.

When reset to “0”, no interrupt request is generated.

Caution	The transfer end flag becomes “1” at the end of 8 bits (1 byte) transfer, regardless of the transfer bit length control setting. The overrun flag is only set when overrun was detected. It does not generate an interrupt.

SIO1 control register (SCON1)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCON1	134H	R/W	–	SCON16	–	SCON14	SCON13	SCON12	SCON11	SCON10
Reset			H	0	H	0	0	0	0	0

Bit name	Function
SCON16 (bit 6)	Overrun flag
	0: No overrun 1: Overrun
SCON14 (bit 4)	Transfer bit length control
	0: 8-bit transfer 1: Continuous transfer
SCON13 (bit 3)	Transfer control
	0: Stop transfer 1: Start transfer
SCON12 (bit 2)	LSB/MSB sequence select
	0: LSB first 1: MSB first
SCON11 (bit 1)	Serial transfer end flag
	0: Transfer in progress 1: Transfer end
SCON10 (bit 0)	Interrupt request enabled
	0: Interrupt request disabled 1: Interrupt request enabled

SCON16 (bit 6): overrun flag

Detects serial transfer errors in SIO1.

When an 8-bit data transfer is completed (SCON11 became “1”) and the transfer clock is received (falling edge was detected), the flag is set.

During continuous transfer, the overflow flag is set every 8 bits.

Caution	This flag is not reset automatically. It must be reset by the application.

SCON14 (bit 4): transfer bit length control

Switches the SIO1 transfer data bit length to 8 bit continuous (1) or 8 bit (0).

When set to “1”, data of 2 byte or more can be sent continuously in 8-bit units.

When reset to “0”, only 8 bits of data (1 byte) can be sent. When the 8-bit transfer is completed, the transfer end flag (SCON11) is set.

Caution	This flag is not reset after transfer. It must be reset by the application.

SCON13 (bit 3): SIO1 operation control

Starts or stops SIO1 transfer.

When set to "1", 8-bit serial transfer at SIO1 starts. When 8 bits have been transferred, the flag is reset.

When reset to "0", serial transfer at SIO1 stops.

SCON12 (bit 2): LSB/MSB start select

Selects whether data are transferred starting with the MSB or LSB.

When set to "1", the transfer starts with the MSB.

When reset to "0", the transfer starts with the LSB.

Caution	This flag applies both to sending and receiving data. The setting must match at both ends.

SCON11 (bit 1): SIO1 transfer end flag

Detects the end of serial transfer.

The flag is set when a serial transfer of 8 bits is completed.

When this flag is set, and a falling edge of the transfer clock is detected, the overrun flag is set.

Caution	This flag is not reset automatically. It must be reset by the application.

SCON10 (bit 0): SIO1 interrupt request enable control

Enables or disables interrupt request generation at SIO1 transfer end.

When set to "1", the interrupt vector 003BH is called when SIO1 transfer ends.

When reset to "0", no interrupt request is generated.

Caution	The transfer end flag becomes "1" after 8 bits (1 byte) have been transferred, regardless of the transfer bit length control setting. The overrun flag is only set when overrun was detected. It does not generate an interrupt.

Baud rate generator register (SBR)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SBR	132H	R/W	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
Reset			0	0	0	0	0	0	0	0

When the internal clock is used as transfer clock, this register sets the transfer rate. The value is common to both SIO0 and SIO1. The transfer rate TSBR can be obtained by the following equation.

T_{SBR} = (256 - [SBR setting value]) × 2 × Tcyc (Tcyc = cycle clock)

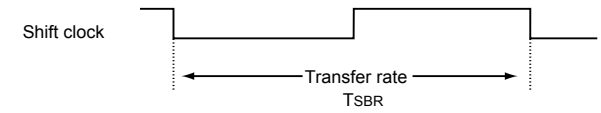


Fig. 4-34 SIO0, SIO1 transfer rate

Serial buffer 0 (SBUF0)

Stores transfer data (8 bits) from SIO0.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SBUF0	131H	R/W	SBUF07	SBUF06	SBUF05	SBUF04	SBUF03	SBUF02	SBUF01	SBUF00
Reset			0	0	0	0	0	0	0	0

Serial buffer 1 (SBUF1)

Stores transfer data (8 bits) from SIO1.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SBUF1	135H	R/W	SBUF17	SBUF16	SBUF15	SBUF14	SBUF13	SBUF12	SBUF11	SBUF10
Reset			0	0	0	0	0	0	0	0

Dedicated Dreamcast interface circuit

In addition to the serial interface described above, port 1 also operates as an input/output port for the dedicated Dreamcast interface. It is not possible to use the dedicated Dreamcast interface and the serial interface simultaneously.

4.5.4 Serial Interface Operation

Transfer via the serial interface starts when the transfer control bit (SCON03, SCON13) or transfer bit length select bit (SCON04, SCON14) is set. The transfer mode can be the Maple bus mode or normal mode, but applications can only use the normal mode.

Normal mode

Data transfer uses two data lines and one clock line. The data lines are S1 (dedicated input) and S0 (dedicated output). This mode follows conventional transfer principles and is suited for communication with a specific device.

Caution	For communication between two VMS units, use normal mode.

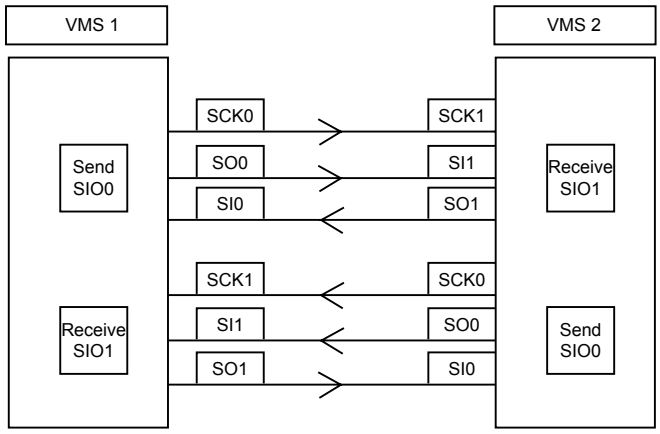


Fig. 4-35 Connection of two VMS units

The transfer mode can be specified by operating the Special Function Register assigned to port 1 (refer to Table 4-3). The mode can be specified separately for SIO0 and SIO1.

Serial transfer timing

The shift registers are synchronized to the falling edge of the serial clock SCK0 and SCK1, and data from the shift registers are output at the SO0 and SO1 pins. At the rising edge of the serial clock, the data input from pins SI0 and SI1 are read into the shift registers.

4.5.5 Operation Mode Settings

Normal mode

The output pins or port latch data for the transfer clock used as internal clock must be reset. The following pins are used in normal mode.

Table 4-3 Pins Used in Normal Mode

Mode	SIO0	SIO1
Input pin	P11/SI0/SB0	P14/SI1/SB1
Output pin	P10/SO0_P11/SI0/SB0_	P13/SO1_P14/SI1/SB1_
Transfer clock	P12/SCK0	P15/SCK1

Caution	1 Tcyc before starting transfer, SCKn is set to "1". At less than 1 Tcyc, correct data will not be output.

Table 4-4 Port 1 Settings for SIO0 (Special Function Registers)

Pin	Function	Special function register value
P11/SI0/SB0 P10/SO0	Receive Send	P11DDR = 0 P10 = 0 P10DDR = 1 P10FCR = 1
P11/SI0/SB0 P10/SO0	Receive General input/output	P11DDR = 0 P10FCR = 0
P12/SCK0	Internal clock	P12 = 0 P12DDR = 1 P12FCR = 1

Note	The software clock is programmed to alternately write "0" and "1" to the port (P12), and the output is used as transfer clock.

Table 4-5 Port 1 Settings for SIO1 (Special Function Registers)

Pin	Function	Special function register value
P14/SI11/SB1 P13/SO1	Receive General input/output	P14DDR = 0 P13FCR = 0

Note	The software clock is programmed to alternately write "0" and "1" to the port (P15), and the output is used as transfer clock.

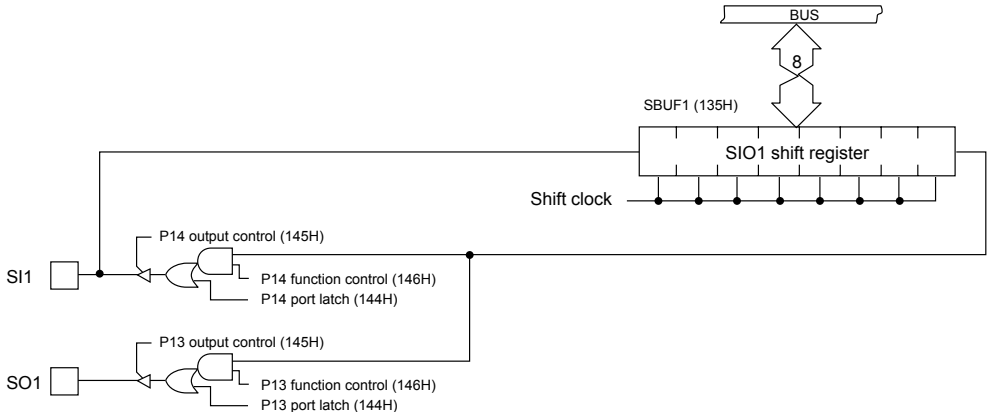


Fig. 4-36 Normal mode signal path (SIO1 example)

Caution	To set Pn to "output", PnFCR must be set to "1" before PnDDR. If PnDDR is set first, "0" may be output from Pn when PnDDR is set. This applies both to SIO0 and SIO1.

4.5.6 Serial transfer clock

The serial transfer clock (shift clock) uses the P12/SCK0 pin for SIO0 and the P15/SCK1 pin for SIO1. According to the applied circuit specifications, one of the following three clock types can be selected for SIO0 and SIO1 separately. For SIO0, the transfer clock polarity can also be selected.

- Internal clock
- External clock
- Software clock

Internal clock

Normally, the internal clock is used for serial transfer. The dedicated serial baud rate generator (SBR) integrated in the VMS custom chip generates the transfer clock which is supplied to the SIO0 and SIO1 circuitry.

When the serial interface is driven with the internal clock, the baud rate generator must be activated. When this is done, the serial transfer clock will be output from the serial interface clock pin (P12/SCK0, P15/SCK1).

The relationship between the transfer rate and the baud rate generator setting is as shown below. The setting values are decimal.

TSBR = (256 - [SBR setting value]) × 2 × Tcyc (Tcyc is the cycle clock)

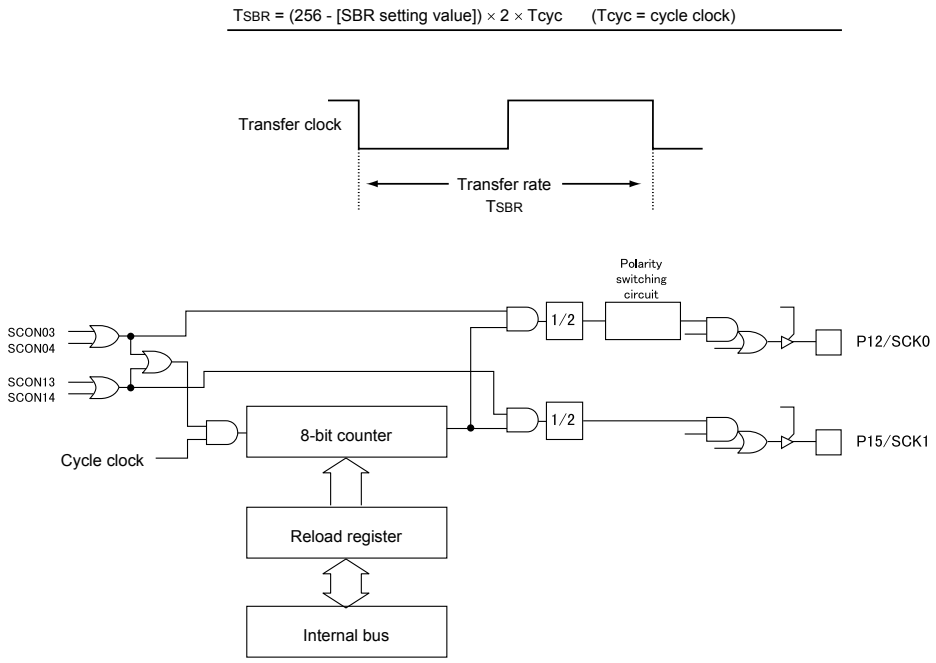


Fig. 4-37 Baud Rate Generator Configuration Diagram

Caution	To set Pn to “output”, PnFCR must be set to “1” before PnDDR. If PnDDR is set first, “0” may be output from Pn when PnDDR is set. This applies both to SIO0 and SIO1.

External clock

The VMS custom chip can perform serial transfer using an externally supplied clock.

Software clock

Using a program to alternately output “0” and “1” from ports P12/SCK0 and P15/SCK1, the output can be used as serial transfer clock.

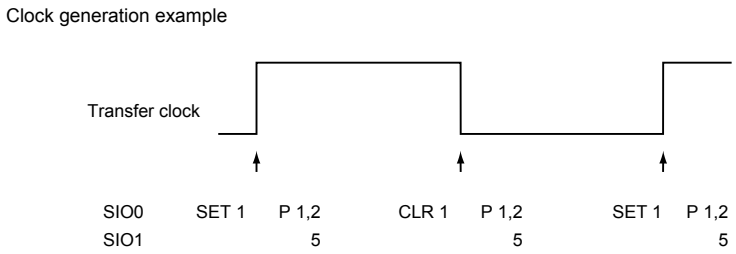


Fig. 4-38 Clock Generation Example

To use this type of transfer clock, appropriate settings must be made first for ports P12/SCK0 and P15/SCK1.

Table 4-6 Transfer Clock Settings

Pin	Function	Special function register value
P12/SCK0	Internal clock	P12 = 0 P12DDR = 1 P12FCR = 1
	External clock	P12DDR = 0
	Software clock	P12 = 0/1 P12DDR = 1 P12FCR = 0
P15/SCK1	Internal clock	P15 = 0 P15DDR = 1 P15FCR = 1
	External clock	P15DDR = 0
	Software clock	P15 = 0/1 P15DDR = 1 P15FCR = 0

Caution

- Serial data and serial clock pulse width must be at least 1/2 the cycle time.
  This is especially important when using the quartz oscillator or the external clock. For example, when using the 32.768 kHz quartz oscillator, the cycle clock will be 366 μs, requiring a pulse width of at least 183 μs.
- When outputting the serial clock from port 1, the port 1 registers must be set in the order shown below, otherwise correct operation is not assured.
  (1) P1FCR setting
  (2) P1DDR setting
  (3) SCONn setting (transfer control bit setting)

4.5.7 Serial Transfer Timing

During serial transfer, the transfer clock SCK0 of SIO0 (when SCON07 = 0) and SIO1 outputs a "High" level (SCK0 = 1) before and after operation. The last transferred data is held at the output.

When SCON07 = 1, the transfer clock SCK0 of SIO0 outputs a "Low" level (SCK0 = 0) before and after operation. Bit 0 (SBUF00) of the serial buffer 0 (SBUF0) is held at the output (refer to Table 4-40). In SIO1, polarity switching is not possible.

SIO0

SCON07 = 0	At operation stop, SCK0 = 1, maintain data output
SCON07 = 1	At operation stop, SCK0 = 0, output data is bit 0 of SBUF0

SIO1

At operation stop, SCK0 = 1, maintain data output

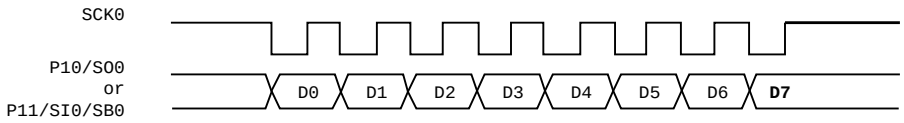


Fig. 4-39 Transfer Clock and Output Data (1)

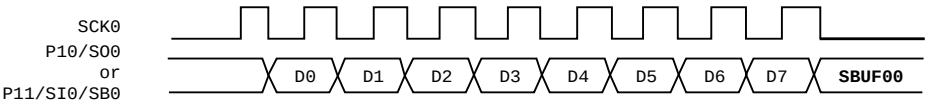


Fig. 4-40 Transfer Clock and Output Data (2)

4.5.8 LSB/MSB Switchable Start Sequence

The serial transfer buffer read/write order for data can be set to either LSB → MSB or MSB → LSB.

Note	For VMS, either of these is acceptable. The method used by computers complying to the RS-232C standard is LSB → MSB.

This function allows selection of "LSB-first" or "MSB- first" order for serial data transfer. The setting is made with the serial transfer control register (SCON0, SCON1).

Caution	The LSB/MSB order selection must be made before the start of transfer. If the setting is changed after transfer has started, the transfer continues with the original setting.

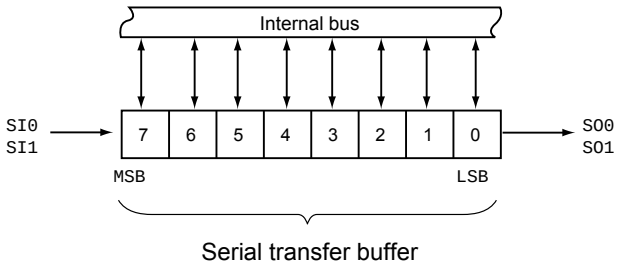


Fig. 4-41 Serial Transfer Buffer and Internal Bus When LSB First Is Selected

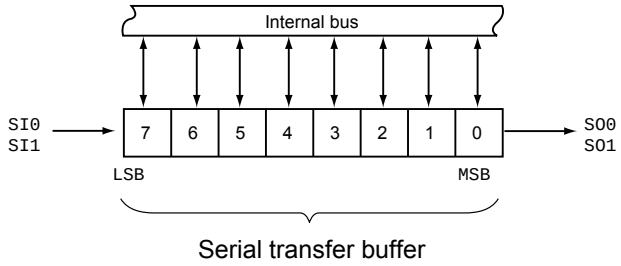


Fig. 4-42 Serial Transfer Buffer and Internal Bus When MSB First Is Selected

Figures 4-43 and 4-44 show the serial transfer send/receive timing in SIO0 when using LSB first and MSB first.

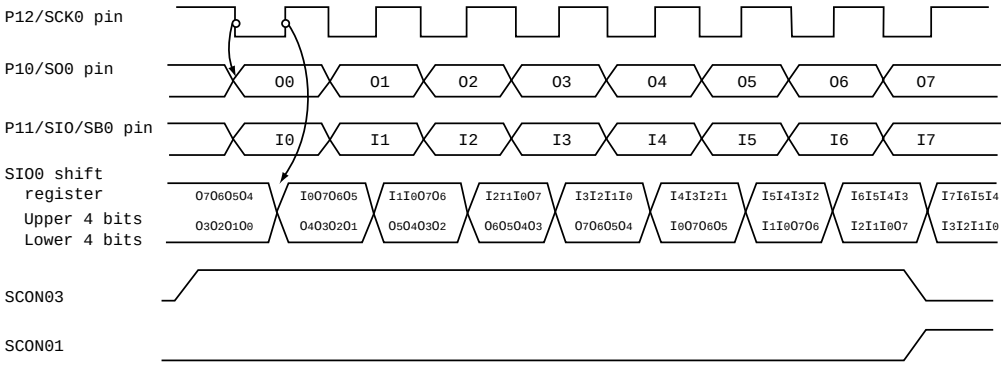


Fig. 4-43 Serial Transfer Buffer and Internal Bus When LSB First Is Selected

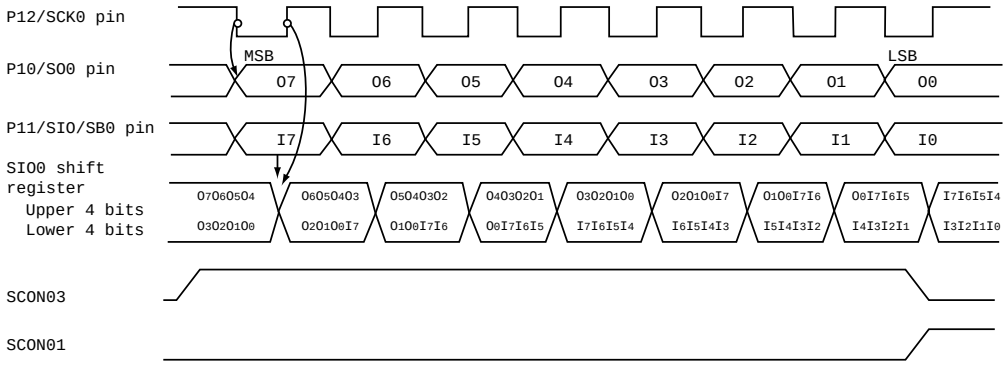


Fig. 4-44 Serial Transfer Buffer and Internal Bus When MSB First Is Selected

4.5.9 Overrun Detection

The overrun detection function serves to catch serial transfer errors.

If the interrupt source flag (SCON01, SCON11) is set, the overrun flag (SCON06, SCON16) will be set at the falling edge of the transfer clock.

The normal send timing and overrun timing are shown in Fig. 4-45. At the rising edge of the transfer clock for the 8th data bit, the interrupt source flag (SCON01, SCON11) is set. When the falling edge of the transfer clock is detected in this condition, the overrun detection flag is set.

The overrun flag is only set when overrun is detected. It does not generate an interrupt or have other results.

Caution	• Before checking the overrun flag, wait at least 1/2 transfer clock cycles after the interrupt source flag was set to “1”. • Also when a transfer mode exceeding 8 bits (continuous transfer mode) was set, the overrun detection function operates in the same way as for 8-bit transfer.

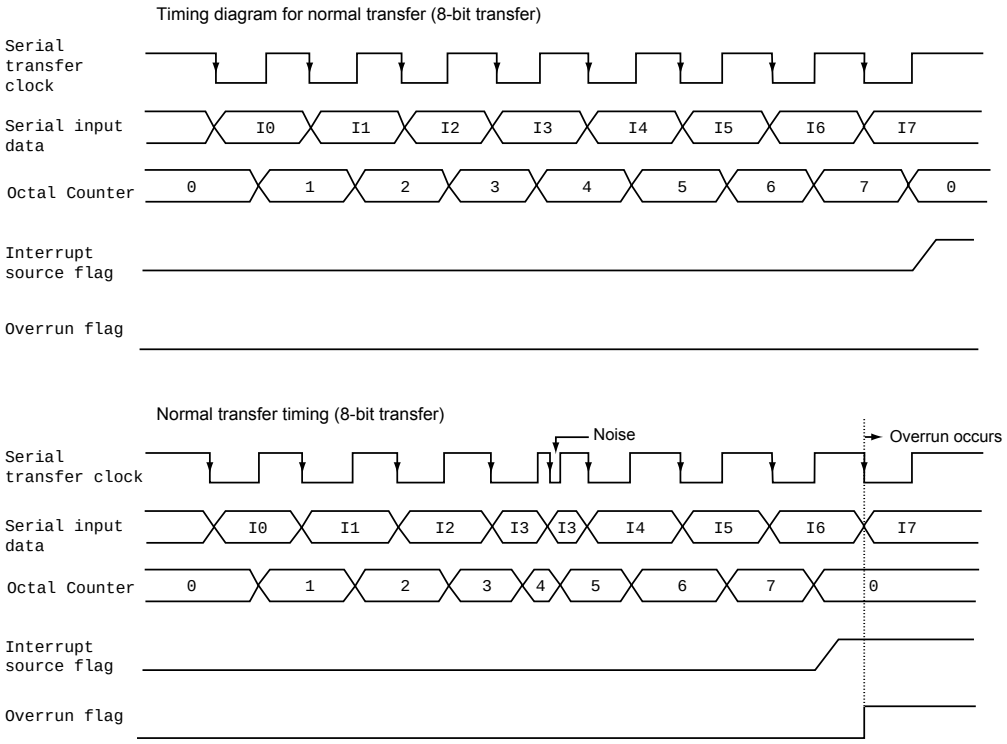


Fig. 4-45 Serial Transfer Buffer and Internal Bus When LSB First Is Selected

4.5.10 Transfer Bit Length Control

When sending more than 8 bits of serial data, continuous transfer can be selected with the transfer bit length control bit SCON04 or SCON14.

- When SCON04 or SCON14 is set, serial transfer starts. This bit is not reset also after 8 bits of data have been transferred.
- The interrupt source flag is set with the same timing as for 8-bit transfer (at 8-bit transfer end).
- The overrun detection bit SCON06 or SCON16 is set at the serial clock falling edge when 8 bits have been exceeded. For information on the timing, see section 4.5.9 “Overrun Detection”.
- When the transfer bit length has been set to 8 bits, transfer starts when the transfer control bit SCON03 or SCON13 is set. When 8 bits of data have been transferred, the transfer control bit is reset which in turn causes the interrupt source flag (SCON01, SCON11) to be set. Serial transfer stops automatically.
- When the transfer bit length has been set to continuous, transfer starts when the transfer bit length control bit SCON04 or SCON14 is set. Transfer continues until the bit is reset. The interrupt source flag is set after 8 bits of data have been transferred.

4.5.11 Sample Program

SIO0 serial transfer (1) (send) sample program

Transfer parameters

- 8-bit transfer
- Transfer data: 038H (8 bits)
- MSB-first
- Falling edge output
- Normal mode
- Internal clock
- Baud rate: 25.6 ms
- System clock: 32.768 kHz quartz oscillator

The baud rate equation yields the following

$TSBR = (256 - [SBR]) \times 2 \times Tcyc$

$\therefore [SBR] = 256 - TSBR / (2 \times Tcyc)$

Here, the result is TSBR = 25.6 ms, Tcyc = 366 us. The baud rate generator register (SBR) setting value therefore is as follows.

$[SBR] = 256\text{-}25600 / (2 \times 366)$

$= \text{approx. } 221 \text{ (decimal)}$

$\rightarrow 0DDH \text{ (hex)}$

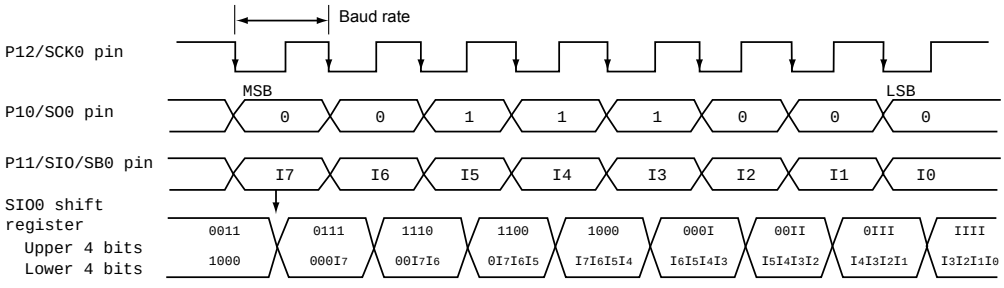


Fig. 4-46 Serial Transfer (1) Timing

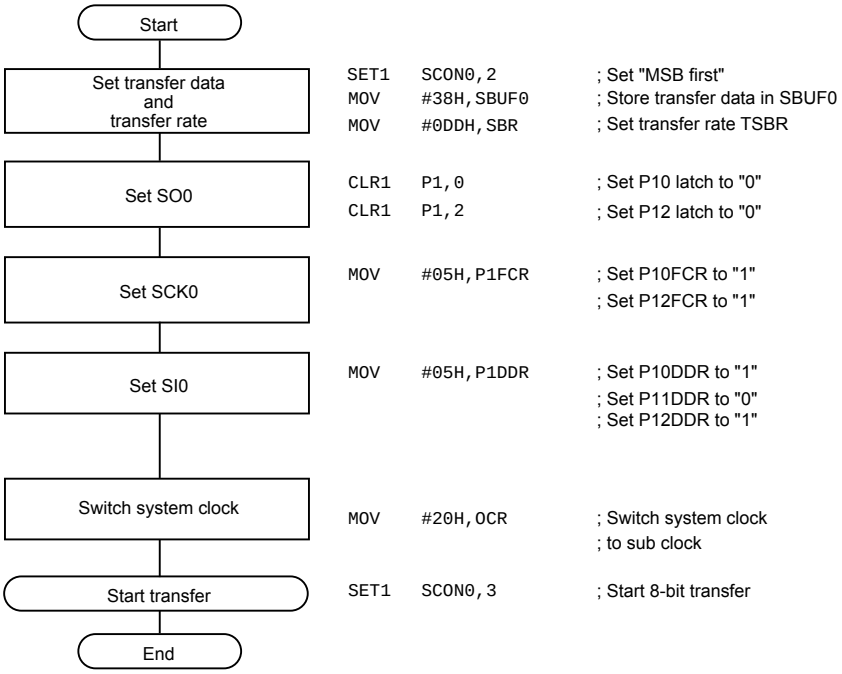


Fig. 4-47 Serial Transfer (Send) Sample Program

SIO1 serial transfer (2) (receive) sample program

Transfer parameters

- 16-bit transfer
- LSB-first
- External clock
- Same output data from SO1 as SB1
- Store upper 8 bit of read data at RAM address 031H, and lower 8 bit at RAM address 030H

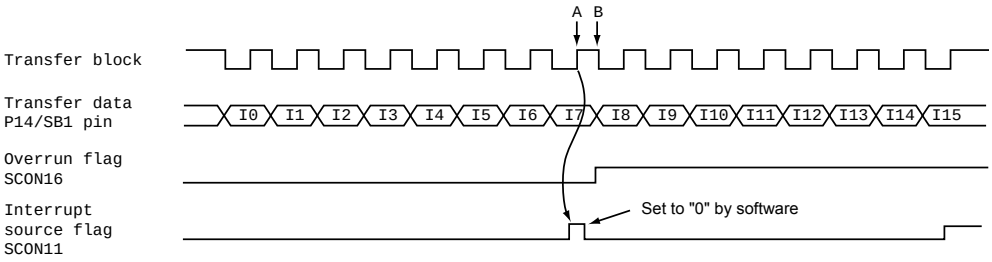
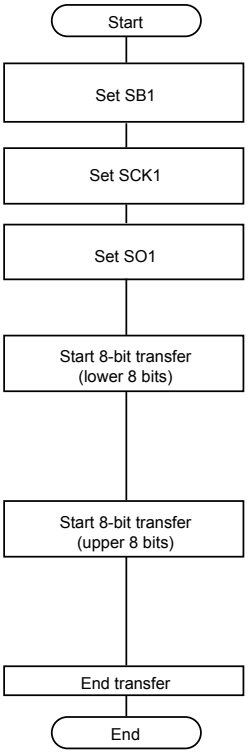


Fig. 4-48 Serial Transfer (2) Timing




```
        CLR1   SCON1,2          ; Set "LSB first"
        CLR1   P1,4             ; Set P14 latch to "0"
        CLR1   P1,3             ; Set P13 latch to "0"

        MOV    #08H,P1DDR       ; Set P13FCR to "1"

        MOV    #08H,P1DDR       ; Set P13DDR to "1"
                                ; Set P14DDR to "0"
                                ; Set P15DDR to "0"

        SET1   SCON1,4          ; Start 16-bit transfer
SELF0:  BN     SCON1,1,SELF0    ; Wait until end of 8-bit transfer
        LD     SBUF1            ; Store data in accumulator
        ST     #30H             ; Store lower 8 bits in RAM30H

        CLR1   SCON1,1          ; Set interrupt source flag to "0"

SELF1:  BN     SCON1,1,SELF1    ; Wait until end of 8-bit transfer
        LD     SBUF1            ; Store data in accumulator
        ST     #31H             ; Store upper 8 bits in RAM31H

        MOV    #0H,SCON1        ; End transfer
```

Fig. 4-49 Serial Transfer (Receive) Sample Program

Caution	• In this example, if there is a rising edge (B) of the transfer clock between the instruction following SELF0 and the SELF1 instruction, an error will occur. The transfer clock should be set with a sufficiently long cycle in relation to the cycle clock. • Set SCKn to "1" 1 Tcyc before the start of transfer. If less than 1 Tcyc, correct data will not be obtained. • To set Pn to "output", PnFCR must be set to "1" before PnDDR. If PnDDR is set first, "0" may be output from Pn when PnDDR is set. This applies both to SIO0 and SIO1.

4.6 Dot Matrix LCD Controller

The LCD controller/driver automatically reads data stored in display RAM and generates the signals to drive the dot matrix LCD. The display mode is a graphics mode in which one bit of data in display RAM corresponds to on/off of one dot on the LCD.

The dot matrix LCD controller/driver consists of the following circuit blocks.

- Display RAM (XRAM)
- Display control register
- LCD power supply

4.6.1 Functions

- Display duty cycle: 1/33
- Display bias: 1/5
- Graphics display
- LCD instruction
  display on/off
- Graphics display capability
  48 (horizontal) × 32 (vertical) matrix + 4 mode icons

The following Special Function Registers must be operated to control the display.

- MCR: display on/off control
- STAD: display start address control
- CNR: horizontal byte number control
- TDR: display duty cycle control
- VCCR: display contrast control
- XBNK: display RAM bank address control

4.6.2 Display RAM (XRAM)

The display RAM consists of two banks of 96 x 8 bits for dot matrix control and three 6-bit banks for icon control.

The LCD controller/driver reads the data stored in XRAM and generates the signal to drive the LCD.

Caution	Before reading from or writing to XRAM, set the system clock to the RC oscillator.

Symbol	Address	R/W	Designation	Initial value	Bank
XRAM	180H - 1FBH	R/W	Display RAM	Undetermined	Bank 0
	180H - 1FBH				Bank 1
	180H - 185H		For icons		Bank 2

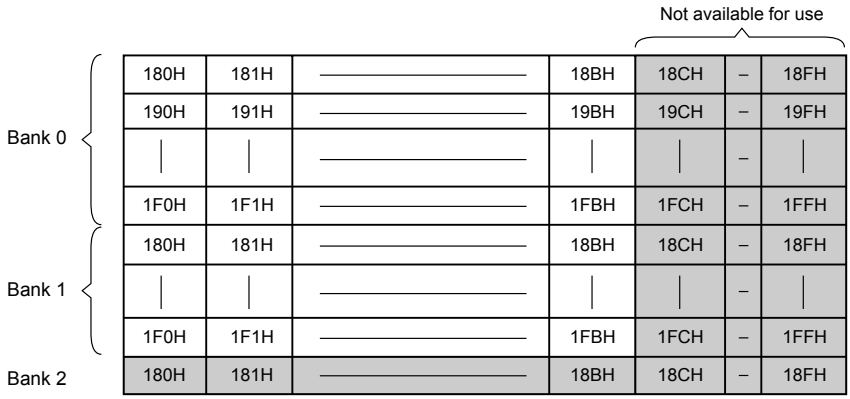


Fig. 4-50 Display XRAM Configuration

4.6.3 Display Control Registers

Mode control register (MCR)

Controls display controller operation start/stop, cursor display, and LCD clock division ratio.

Caution	The mode control register is write-only. When a bit operation instruction or the INC, DEC, or DBNZ instruction is used on a write-only register, a bit other than the specified bit will be set. Use the following instructions for manipulating this register. MOV, MOV@, ST, ST@, POP When accessing the register, bits 7 - 5 and bit 0 must be set to their fixed values.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	120H	W	MCR7	MCR6	MCR5	MCR4	MCR3	–	–	MCR0
Reset			0	0	0	0	0	0	0	0

<table>
<tr><th>Bit name</th><th colspan="4">Function</th></tr>
<tr><td rowspan="3">MCR7 (bit 7)
MCR6 (bit 6)
MCR5 (bit 5)</td><td colspan="4">LCD clock division ratio selection</td></tr>
<tr><td>MCR7</td><td>MCR6</td><td>MCR5</td><td>Division ratio</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1/1 * Always set MCR7 - MCR5 to 0</td></tr>
<tr><td rowspan="2">MCR4 (bit 4)</td><td colspan="4">LCD clock 1/2 division ratio select circuit</td></tr>
<tr><td colspan="4">0: Signal determined by MCR7 - MCR5 is divided by 2 and selected as LCD clock
1: Signal determined by MCR7 - MCR5 is selected as LCD clock (direct mode)</td></tr>
<tr><td rowspan="2">MCR3 (bit 3)</td><td colspan="4">LCD controller control</td></tr>
<tr><td colspan="4">0: LCD controller stop
1: LCD controller start/continue</td></tr>
<tr><td rowspan="2">MCR0 (bit 0)</td><td colspan="4">Display mode select</td></tr>
<tr><td colspan="4">1: Graphic mode * Always set MCR4 to 1</td></tr>
</table>

MCR7, MCR5 (bits 7, 5): LCD clock division ratio

Be sure to reset MCR7, MCR5 to “0”.

MCR4 (bit 4): LCD clock 1/2 division select

This bit controls whether to divide the LCD clock selected with MCR7 - MCR5 by 2.

When reset to “0”, the LCD clock is divided by 2.

When set to “1”, the LCD clock is not divided.

The frame frequency is as follows.

1/2 division (MCR4 = 0): 82.7 Hz

1/1 division (MCR4 = 1): 165.5 Hz

MCR3 (bit 3): LCD controller control

This bit controls display controller operation start/stop.

When set to “1”, the LCD controller operation starts.

When reset to “0”, the LCD controller operation stops. This means that the display will not be updated also when the XRAM contents change. The actual LCD is not switched on and off.

MCR0 (bit 0): display mode select

The display mode should always be set to graphics mode.

Graphics display MCR0 = 1

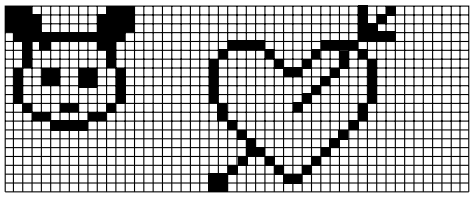


Fig. 4-51 Dot Matrix Display

Display start address control register (STAD)

Controls the display start address.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STAD	122H	R/W	STAD7	STAD6	STAD5	STAD4	STAD3	STAD2	STAD1	STAD0
Reset			0	0	0	0	0	0	0	0

Bit name	Function								
STAD7 (bit 7)	Display RAM start address setting								
|	STAD7	STAD6	STAD5	STAD4	STAD3	STAD2	STAD1	STAD0	Start address
STAD0 (bit 0)	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	↓	↓							↓
	1	1	1	1	1	1	1	1	255

Caution

Changing the STAD value affects not only XRAM bank 0 but also banks 1 and 2, which can cause icons to flash or all icons to be shown simultaneously. In such a case, change the data in bytes 1 to 6 of the display start address specified by STAD as follows.

Start address specified by STAD	-6	-5	-4	-3	-2	-1
Write data	00H	00H	00H	10H	00H	00H

This will cause the game icon only to be shown. Access to bank 2 of XRAM is normally prohibited, but if STAD is set to a value other than 00H, it is allowed.

When flash memory access is carried out while STAD is set to a value other than 00H, a part of the screen is rewritten. This is because the BIOS causes the flash memory access icon to be shown. Therefore, STAD should be reset to 00H before calling the BIOS to access the flash memory.

STAD7, STAD0 (bits 7, 0): XRAM display start address setting

These bits set the starting address of the display data for the LCD (XRAM 180H is assumed as STAD = 00H).

The data changes in 2-byte units.

Start address	XRAM address	STAD7	STAD6	STAD5	STAD4	STAD3	STAD2	STAD1	STAD0
0H	180H (bank 0)	0	0	0	0	0	0	0	0
1H	182H (bank 0)	0	0	0	0	0	0	0	1
2H	184H (bank 0)	0	0	0	0	0	0	1	0
3H	186H (bank 0)	0	0	0	0	0	0	1	1
4H	188H (bank 0)	0	0	0	0	0	1	0	0
5H	18AH (bank 0)	0	0	0	0	0	1	0	1
6H	Not available	0	0	0	0	0	1	1	0
7H	Not available	0	0	0	0	0	1	1	1
8H	190H (bank 0)	0	0	0	0	1	0	0	0
9H	192H (bank 0)	0	0	0	0	1	0	0	1
0AH	194H (bank 0)	0	0	0	0	1	0	1	0
0BH	196H (bank 0)	0	0	0	0	1	0	1	1
0CH	198H (bank 0)	0	0	0	0	1	1	0	0
0DH	19AH (bank 0)	0	0	0	0	1	1	0	1
0EH	Not available	0	0	0	0	1	1	1	0
0FH	Not available	0	0	0	0	1	1	1	1
10H	1A0H (bank 0)	0	0	0	1	0	0	0	0
11H	1A2H (bank 0)	0	0	0	1	0	0	0	1
:	:	:	:	:	:	:	:	:	:
3DH	1FAH (bank 0)	0	0	1	1	1	1	0	1
3EH	Not available	0	0	1	1	1	1	1	0
3FH	Not available	0	0	1	1	1	1	1	1
40H	180H (bank 1)	0	1	0	0	0	0	0	0
41H	182H (bank 1)	0	1	0	0	0	0	0	1
:	:	:	:	:	:	:	:	:	:
7DH	1FAH (bank 1)	0	1	1	1	1	1	0	1
7EH	Not available	0	1	1	1	1	1	1	0
7FH	Not available	0	1	1	1	1	1	1	1
80H	180H (bank 2)	1	0	0	0	0	0	0	0
81H	182H (bank 2)	1	0	0	0	0	0	0	1
82H	184H (bank 2)	1	0	0	0	0	0	1	0
83H - FFH	Not available								

Caution	As indicated in the table, some start addresses can lead to operation errors. Do not use xx6H, xx7H, xxEH, xxFH as start addresses.

Character number register (CNR) 123H

This register may not be accessed by applications.

Time division register (TDR) 124H

This register may not be accessed by applications.

Bank address register (XBNK)

Switches the XRAM bank.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XBNK	125H	R/W	–	–	–	–	–	–	XBNK1	XBNK0
Reset			0	0	0	0	0	0	0	0

Bit name	Function		
XRBK1 (bit 1) | XRBK0 (bit 0)	Set display RAM start address		
	XRBK1	XRBK0	Bank address
	0	0	0
	0	1	1
	1	0	2
	1	1	Not available

XBNK1, XBNK0 (bits 1, 0): display RAM bank address control

Switches the XRAM bank.

The dot matrix display RAM banks 0 and 1 have a capacity of 96 bytes each. Applications can access only banks 0 and 1.

XRAM bank 2 contains 6 bytes and serves for the icons that indicate the VMS operation mode.

Caution	Applications may not manipulate XRAM bank 2.

LCD contrast control register (VCCR)

This register controls the LCD on/off state.

Caution	• The unit does not incorporate a contrast control. • The LCD contrast control register is write-only. When a bit operation instruction or the INC, DEC, or DBNZ instruction is used on a write-only register, a bit other than the specified bit will be set. Use the following instructions for manipulating this register. MOV, MOV@, ST, ST@, POP When accessing the register, bits 5 to 0 must be set to their fixed values.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VCCR	127H	W	VCCR7	VCCR6	VCCR5	VCCR4	VCCR3	VCCR2	VCCR1	VCCR0
Reset			0	0	0	0	0	0	0	0

Bit name	Function
VCCR7 (bit 7)	Liquid crystal display control
	0: Liquid crystal display OFF 1: Liquid crystal display ON
VCCR6 (bit 6)	LCD RAM access control
	0: CPU RAM access enabled 1: CPU RAM access disabled
VCCR5 (bit 5) | VCCR0 (bit 0)	* Always set VCCR5 - VCCR0 to 0

VCCR7 (bit 7): LCD display control

This bit specifies whether display is carried out or not.

When reset to “0”, power to the LCD is shut off, so that the display is deactivated.

(1) Liquid crystal display ON (VCCR7 = 1)

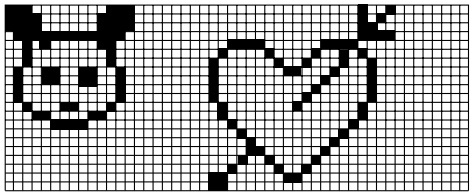


Fig. 4-52 LCD ON State

When set to “1”, power to the LCD is supplied, so that the display is activated.

(2) Liquid crystal display OFF (VCCR7 = 0)

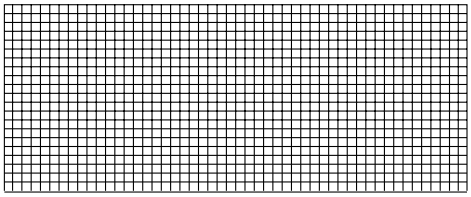


Fig. 4-53 LCD OFF State

Caution	Always start the display controller (MCR3 = 1) before activating the display (VCCR7 = 1). To deactivate the display, first set VCCR7 to “0” and then set MCR3 to “0”.

VCCR6 (bit 6): LCD display RAM access control

When the quartz oscillator is used as system clock and LCD display is on, be sure to disable access from the CPU to the XRAM (VCCR6 = 1) after changing the system clock.

When reading from or writing to XRAM, or when the RC oscillator is used for the system clock and the display is used, enable access from the CPU to the XRAM (VCCR6 = 0).

The procedure for changing the system clock while the display is used is as follows.

- RC oscillator → quartz oscillator

  VCRR6 = 1

  OCR5 = 1, OCR4 = 0

- Quartz oscillator → RC oscillator

  VCRR6 = 0

  OCR5 = 0, OCR4 = 0 (RC oscillator)

VCCR5 - VCCR0 (bits 5 - 0)

Always set these bits to “0”.

Caution	When using the LCD, set the VCCR last.

4.7 External Interrupt Function

The VMS custom chip has a function that detects external input signals on P70/INT0, P71/INT1, P72/INT2/T0IN, and P73/INT3/T0IN and generates interrupt requests to four vector addresses.

The signal types to be detected can be selected by the application. P70 is used for detecting when the VMS is connected to the controller. P71 is used for low-voltage detection.

Detection pins and interrupt vectors

Pin	Vector address	Pin	Vector address
P70/INT0	003H	P72/INT2/T0IN Pin	013H
P71/INT1	00BH	P73/INT3/T0IN Pin	01BH

Signals that can be detected

The priority ranking of the INT0 and INT1 pin interrupts can be set to either “high” or “low” by the master interrupt enable control register (IE). When set to “high”, interrupt processing is carried out regardless of the master interrupt enable setting. The priority ranking of interrupts other than INT0 and INT1 can be set to either

"high" or "low" by the interrupt priority control register (IP). A noise filter with switchable time constant is connected to the P73/INT3/TOIN pin.

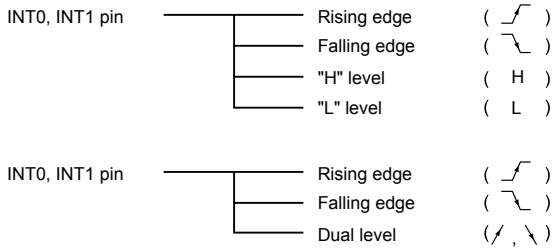


Fig. 4-54 Interrupt Detection Signals

Detection of other VMS unit

When another VMS unit is connected, the values at pins P70 through P73 are as follows.

	P70	P72	P73
Connected to VMS	L	L	H
Not connected to VMS	L	L	L

To use the external interrupt function, the following Special Function Registers must be operated.

I01CR, I23CR, ISL, IE

4.7.1 Circuit Configuration

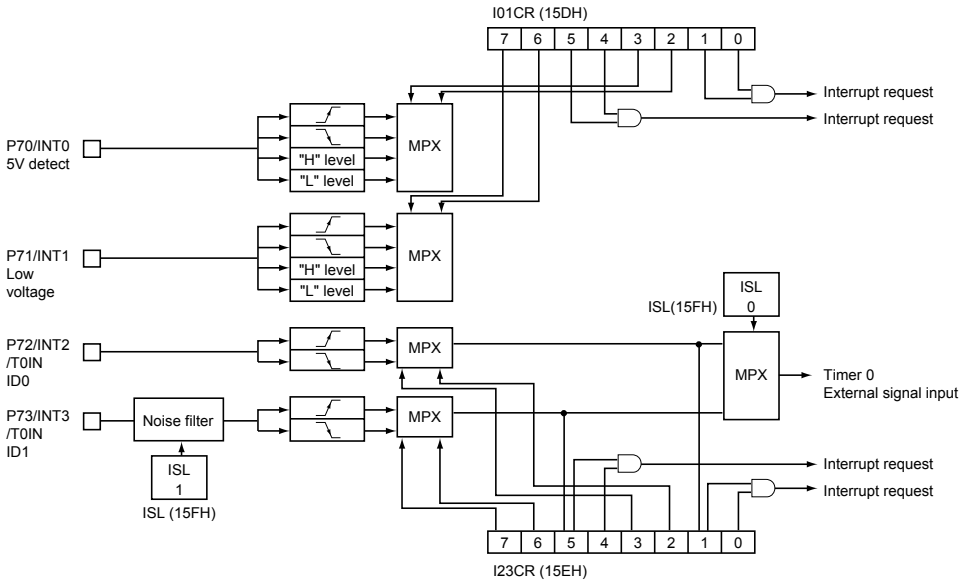


Fig. 4-55 External Interrupt Circuit Block Diagram

4.7.2 Related Registers

External interrupt 0, 1 control register (I01CR)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I01CR	15DH	R/W	I01CR7	I01CR6	I01CR5	I01CR4	I01CR3	I01CR2	I01CR1	I01CR0
Reset			0	0	0	0	0	0	0	0

Bit name	Function		
IO1CR7 (bit 7) IO1CR6 (bit 6)	INT1 detection level/edge select		
	I01CR7	I01CR6	INT1 interrupt condition
	0	0	Detect falling edge
	0	1	Detect “L” level
	1	0	Detect falling edge
	1	1	Detect “H” level
IO1CR5 (bit 5)	INT1 interrupt source		
	0: Interrupt source disabled 1: Interrupt source enabled		
IO1CR4 (bit 4)	INT1 interrupt control		
	0: Interrupt disabled 1: Interrupt enabled		
IO1CR3 (bit 3) IO1CR2 (bit 2)	INT0 detection level/edge select		
	I01CR3	I01CR2	INT0 interrupt condition
	0	0	Detect falling edge
	0	1	Detect “L” level
	1	0	Detect falling edge
	1	1	Detect “H” level
IO1CR (bit 0)	INT0 interrupt source		
	0: Interrupt source disabled 1: Interrupt source enabled		
IO1CR1 (bit 1)	INT0 interrupt control		
	0: Interrupt disabled 1: Interrupt enabled		

I01CR7, I01CR6 (bits 7, 6): INT1 detection level/edge select

Selects the INT1 interrupt condition for signals input on the P71/INT1 pin.

I01CR7	I01CR6	INT1 interrupt condition
0	0	Detect falling edge
0	1	Detect “L” level
1	0	Detect rising edge
1	1	Detect “H” level

Note	When level detection is used, an interrupt is generated continuously while the signal is at “High” or “Low” level.

I01CR5 (bit 5): INT1 interrupt source

This bit is set if the condition specified by bits I01CR7 and I01CR6 is met. If INT1 interrupt is enabled (I01CR4 = 1), the interrupt vector 00BH is called and interrupt processing begins.

Caution	This flag is not reset automatically. It must be reset by the application.

I01CR4 (bit 4): INT1 interrupt enable control

This bit enables or disables the external INT1 interrupt.

When set to “1”, INT1 interrupt processing is carried out if I01CR5 is set.

When reset to “0”, interrupt processing is not carried out.

I01CR3, I01CR2 (bits 3, 2): INT0 detection level/edge select

Selects the INT0 interrupt condition for signals input on the P70/INT0 pin.

I01CR3	I01CR2	INT0 interrupt condition
0	0	Detect falling edge
0	1	Detect “L” level
1	0	Detect rising edge
1	1	Detect “H” level

Note	When level detection is used, an interrupt is generated continuously while the signal is at “High” or “Low” level.

I01CR1 (bit 1): INT0 interrupt source

This bit is set if the condition specified by bits I01CR3 and I01CR2 is met. If INT0 interrupt is enabled (I01CR0 = 1), the interrupt vector 0003H is called and interrupt processing begins.

Caution	This flag is not reset automatically. It must be reset by the application.

I01CR0 (bit 0): INT0 interrupt enable control

This bit enables or disables the external INT0 interrupt.

When set to “1”, INT0 interrupt processing is carried out if I01CR1 is set.

When reset to “0”, interrupt processing is not carried out.

External interrupt 2, 3 control register (I23CR)

For details, refer to 4.2 “Timer/Counter 0 (T0)”, section “External Interrupt 2, 3 Control Register (I23CR)”.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I23CR	15EH	R/W	I23CR7	I23CR6	I23CR5	I23CR4	I23CR3	I23CR2	I23CR1	I23CR0
Reset			0	0	0	0	0	0	0	0

Bit name	Function
I23CR7 (bit 7)	INT3 rising edge detect control
	0: Detect disabled 1: Detect enabled
I23CR6 (bit 6)	INT3 falling edge detect control
	0: Detect disabled 1: Detect enabled
I23CR5 (bit 5)	INT3 interrupt source
	0: Interrupt source disabled 1: Interrupt source enabled
I23CR4 (bit 4)	INT3 interrupt control
	0: Interrupt disabled 1: Interrupt enabled
I23CR3 (bit 3)	INT2 rising edge detect control
	0: Detect disabled 1: Detect enabled
I23CR2 (bit 2)	INT2 falling edge detect control
	0: Detect disabled 1: Detect enabled
I23CR1 (bit 1)	INT2 interrupt source
	0: Interrupt source disabled 1: Interrupt source enabled
I23CR0 (bit 0)	INT2 interrupt control
	0: Interrupt disabled 1: Interrupt enabled

Input Signal Select Register (ISL)

For details, refer to 4.2 “Timer/Counter 0 (T0)”, section “External Signal Select Register (ISL)”.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ISL	15FH	R/W	–	–	ISL5	ISL4	ISL3	ISL2	ISL1	ISL0
Reset			H	H	0	0	0	0	0	0

Bit name	Function		
ISL5 (bit 5) ISL4 (bit 4)	Base timer clock select		
	ISL5	ISL4	
	1	1	Timer/counter T0 prescaler
	0	1	Cycle clock
	X	0	Quartz oscillator
ISL3 (bit 3)	Use prohibited		
	0: fBST/16 (fixed) 1: Not allowed		
ISL2 (bit 2) ISL1 (bit 1)	Noise filter time constant select		
	ISL2	ISL1	Time constant
	1	1	16Tcyc
	0	1	64Tcyc
	X	0	1Tcyc
ISL0 (bit 0)	T0 clock input pin select		
	0: P72/INT2/T0IN pin 1: P73/INT3/T0IN pin		

Master interrupt enable register (IE)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IE	108H	R/W	IE7	–	–	–	–	–	IE1	IE0
Reset			0	H	H	H	H	H	0	0

Bit name	Function			
IE7 (bit 7)	Master interrupt control (high level, low level)			
	0: All interrupt requests disabled 1: All interrupt requests enabled			
IE1 (bit 1) IE0 (bit 0)	INT0, INT1 interrupt priority control			
	IE1	IE0	INT1 priority level	INT0 priority level
	0	0	Highest	Highest
	1	0	Low	Highest
	X	1	Low	Low

IE7 (bit 7): master interrupt enable control

Enables or disables acceptance of all interrupts, regardless of priority level.

When set to “1”, all interrupt requests are enabled.

When reset to “0”, interrupts of “high” and “low” priority are disabled.

IE1 - IE0 (bits 1 - 0): INT0, INT1 interrupt priority control

Controls the priority level of external interrupts.

IE1	IE0	INT1 priority level	INT0 priority level
0	0	Highest	Highest
1	0	Low	Highest
X	1	Low	Low

Caution	• INT0 and INT1 can be set to “low” priority but not to “high” priority with IE7. • It is not possible to set the external interrupt INT1 only to “high” priority.

4.8 Port Interrupt Functions

In addition to its digital I/O functions, port 3 can be used to generate an interrupt in response to an external input signal, or to cancel the sleep (HALT) condition.

A port interrupt can be implemented through port 3.

4.8.1 Function

In addition to its digital I/O functions, port 3 generates an interrupt when it detects a “Low” level signal.

To use the port interrupt function, the following Special Function Registers must be operated.

P3, P3DDR, P3INT, IE

4.8.2 Circuit Configuration

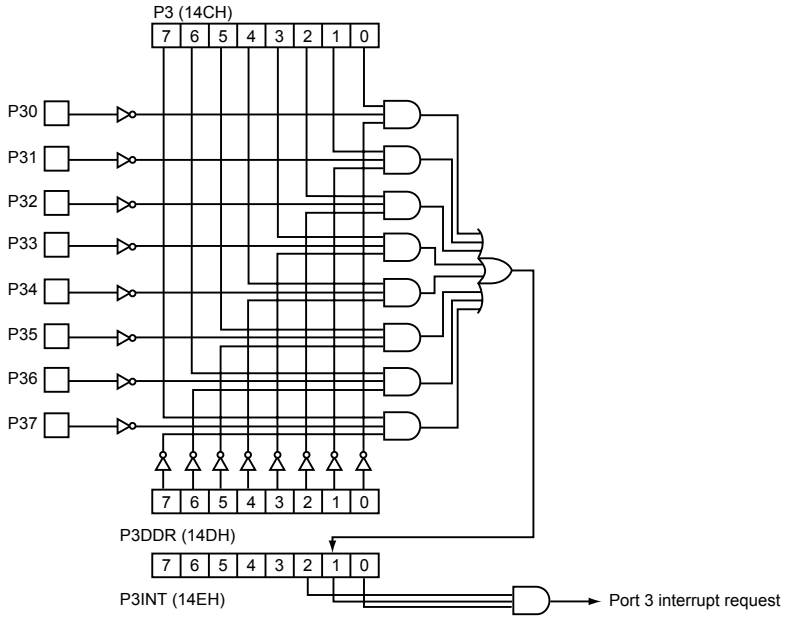


Fig. 4-56 Port 3 Interrupt Circuit Block Diagram

4.8.3 Related Registers

Port 3 interrupt control register (P3INT)

For details, refer to the section “Port 3 Interrupt Control Register (P3INT)” in 4.1.2 “Port 3”.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3INT	14EH	R/W	–	–	–	–	–	P32INT	P31INT	P30INT
Reset			H	H	H	H	H	0	0	0

Bit name	Function
P32INT2 (bit 2)	Port 3 interrupt control flag
	0: Port 3 interrupt generation disabled 1: Port 3 interrupt generation enabled
P32INT1 (bit 1)	Port 3 interrupt source flag
	0: Interrupt source disabled 1: Interrupt source enabled
P32INT0 (bit 0)	Port 3 interrupt request control
	0: Interrupt request disabled 1: Interrupt request enabled

Master interrupt enable control register (IE)

For details, refer to the section “Master interrupt Control Register (IE)” in 4.7 “External Interrupt Functions”.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IE	108H	R/W	IE7	–	–	–	–	–	IE1	IE0
Reset			0	H	H	H	H	H	0	0

Bit name	Function			
IE7 (bit 7)	Master interrupt control (high level, low level)			
	0: All interrupt requests disabled 1: All interrupt requests enabled			
IE1 (bit 1) IE0 (bit 0)	INT0, INT1 interrupt priority control			
	IE1	IE0	INT1 priority level	INT0 priority level
	0	0	Highest	Highest
	1	0	Low	Highest
	X	1	Low	Low

4.8.4 Operation Description

Port 3 interrupt

1. Set bit 2 of the port 3 control register (P3INT) to “1”. This selects port 3 interrupt.
2. Among the pins of port 3 (P37 to P30), select the Special Function Register on which “Low” level detection should occur. The following conditions must be met to accept a port 3 interrupt.
   - The corresponding bit in the port 3 control register (P3DDR) must be set to input mode.

     P3mDDR = 0 (m = 0 to 7)

   - The corresponding bit in the port 3 register (P3) must be set.

     P3n = 1 (m = 0 to 7)

3. When a “Low” level is detected, the interrupt source is set to “1”. If the interrupt request enable flag has been set, an interrupt request is generated, and if the master interrupt enable flag has been set, the interrupt vector 004BH is called.
4. If the conditions listed in 2. are met while in HALT mode, the HALT mode is terminated and the interrupt vector 004BH is called.

4.8.5 State Transition

The flowchart below shows activation and cancellation of HALT mode.

HALT mode state transition

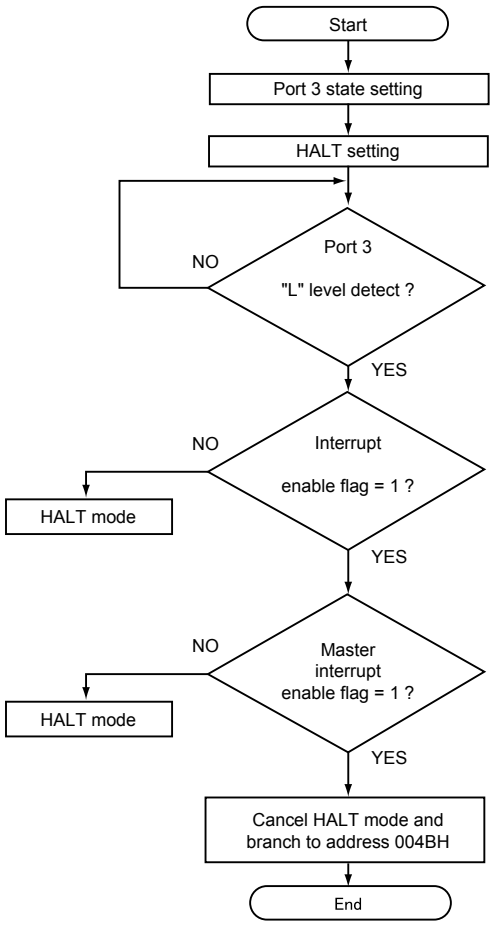


Fig. 4-57 Flow Chart

4.9 VMS Work RAM

The VMS contains 256 bytes × 2 banks of RAM to be used as communications buffer when connected to the Dreamcast. When not connected to the Dreamcast, this RAM is available for applications.

To determine whether data transfer with the Dreamcast is being carried out, check the ASEL flag in the VSEL register. When the flag is "1", data transfer is in progress.

Note that data integrity will not be assured if an application writes to this RAM while data transfer is in progress.

4.9.1 Work RAM Control Registers

VMS control register (VSEL)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VSEL	163H	R/W	–	–	–	INCE	–	–	SIOSEL	ASEL
Reset			H	H	H	0	H	H	0	0

The application can alter only bit 4. Be sure to use a bit level instruction.

INCE (bit 4): VTRBF address counter automatic increment

This bit controls the automatic incrementing of the address counter when reading/writing VTRBF.

When set to "1", the address counter is automatically incremented by 1 after VTRBF has been accessed.

When reset to "0", the address counter maintains its setting.

SIOSEL (bit 1): P1 port use select control

Specifies whether the P1 port (P10 to P15) is to be used as a normal I/O port for synchronous serial communication or as dedicated Dreamcast interface.

ASEL (bit 0): VTRBF address input select control

Controls access to the VTRBF used as buffer for the VMS and dedicated Dreamcast interface.

Work RAM access address (VRMAD1, VRMAD2)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VRMAD1	164H	R/W	VRMAD7	VRMAD6	VRMAD5	VRMAD4	VRMAD3	VRMAD2	VRMAD1	VRMAD0
Reset			0	0	0	0	0	0	0	0

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VRMAD2	165H	R/W	–	–	–	–	–	–	–	VRMAD8
Reset			H	H	H	H	H	H	H	0

Set the address for reading/writing the work RAM (VTRBF). VRMAD1 specifies the lower 8 bits of the address and VRMAD2 the bank. When bit 4 of VSEL is set to "1", VRMAD is incremented each time the VTRBF is accessed.

Work RAM (VTRBF)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VTRBF	166H	R/W	VTRBF7	VTRBF6	VTRBF5	VTRBF4	VTRBF3	VTRBF2	VTRBF1	VTRBF0
Reset			0	0	0	0	0	0	0	0

This register serves for reading and writing data in the address specified by VRMAD.

When this register is written to, the data are written to the RAM address specified by VRMAD.

When this register is read from, the data are read from the RAM address specified by VRMAD.

When bit 4 of VSEL is set to "1", VRMAD is incremented each time the register is accessed.

4.9.2 Accessing Work RAM

To access work RAM, store the desired RAM address in the VRMAD1 and VRMAD2 registers. Then read or write to VTRBF to access data in the work RAM.

Caution	The VRMAD1 and VRMAD2 registers are provided with an auto-increment function. To enable this function, set the INCE bit of VSEL to "1". To disable it, set the bit to "0".

4.9.3 Precautions for Using Work RAM Address Register

The work RAM access procedure is shown in Fig. 4-58.

To access work RAM from an application, specify the work RAM address in the VRMAD1 and VRMAD2 registers.

If the INCE flag of the VSEL register is set to “1”, VRMAD is incremented automatically after each time VTRBF is accessed. The program therefore must be written so as to take the status of the INCE flag of the VSEL register into account.

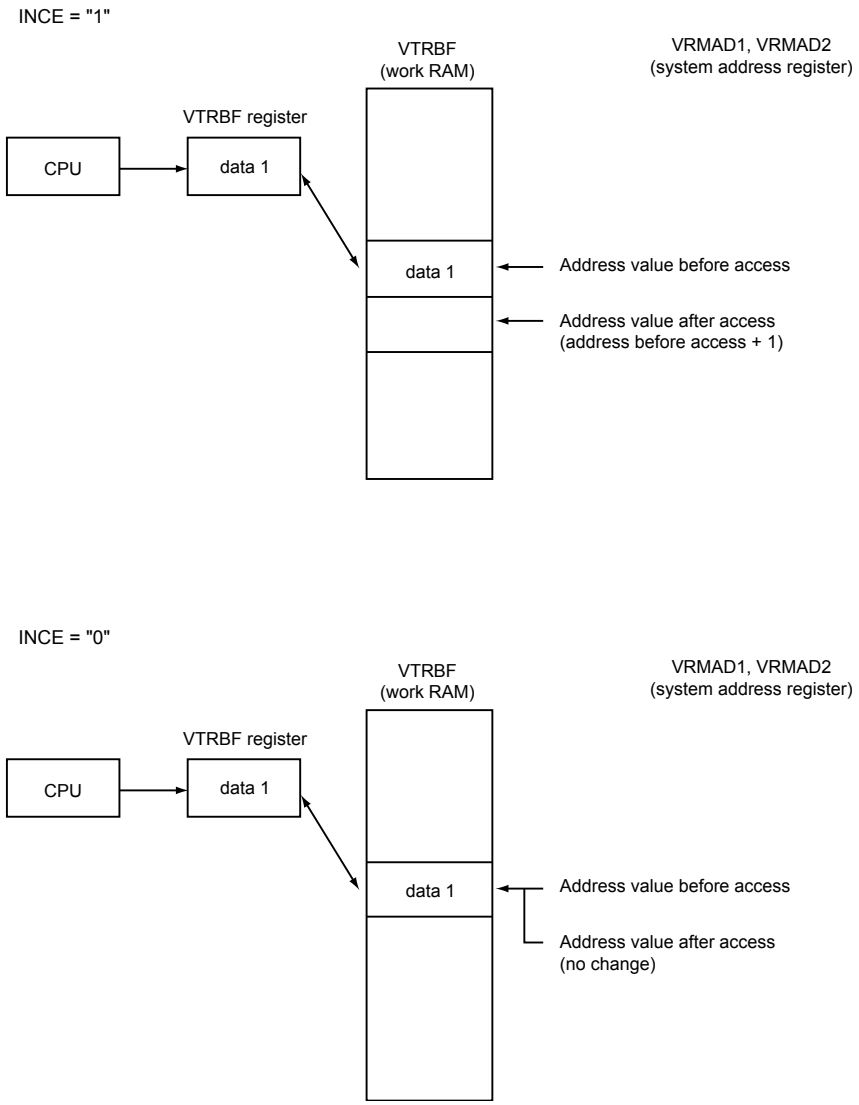


Fig. 4-58 Work RAM Access

4.10 Flash Memory

The VMS custom chip incorporates 128 KB of flash memory (EEPROM = Electrically Erasable Programmable ROM) which can be used to store application program code or data.

4.10.1 Features and Functions

- Capacity: 131072 × 8 bits (program/data area)
- Programmable/erasable in block (page) units
  1 block = 128 bytes (1 page)
- Number of write/erase cycles:
  50,000 times/page (managed by program) (at 25°C ambient temperature)
- Integrated step-up circuit for writing
- Write end detection possible (by calling OS program)
  Toggle bit principle
  Data polling principle
- Software batch erase possible

4.10.2 Accessing Program/Data Area of Flash Memory

The program/data area of flash memory is accessed by calling an OS program. For details, refer to the "BIOS" section.

When connected to the Dreamcast, an application can be transferred to the VMS from the Dreamcast. For details on the transfer procedure, refer to the section on the buSaveExecFile() function in the SEGA Library Manual Vol. 2.

By connecting the development computer to the Dreamcast with a special cable and using the dedicated Memory Card Utility, an application can be transferred to the VMS. For details, refer to the VMS Tutorial.

Chapter 5 Control Functions

This section contains information about the interrupt controller and the system clock.

5.1 Interrupt Functions

Interrupts are used to temporarily interrupt a running program in order to execute other program with higher priority. The VMS incorporates circuits for generating 13 types of interrupts. These are shown in the table below.

Caution	Some interrupt processing functions cannot be set freely by applications.

5.1.1 Interrupt Types

Table 5-1 Interrupt Table

Priority sequence	Interrupt type	Internal /External	Vector address	Interrupt request	Source flag	Enable flag	Register address	Priority setting
1	External interrupt INT0	External	0003H	P70/INT0 event detection	I01CR1	I01CR0	15DH	Highest /low
2	External interrupt INT1	External	000BH	P71/INT1 event detection	I01CR5	I01CR4	15DH	
3	External interrupt INT2	External	0013H	P72/INT2 event detection	I23CR1	I23CR0	15EH	High/low
	Timer/counter T0L (lower 8 bits)	Internal		Timer/counter T0L lower 8 bits overflow	T0CNT1	T0CNT0	110H	
4	External interrupt INT3	External	001BH	P73/INT3 event detection	I23CR5	I23CR4	15EH	High/low
	Base timer	Internal		Base timer overflow	BTCR1	BTCR0	17FH	
					BTCR3	BTCR2		
5	Timer/counter T0H (lower 8 bits)	Internal	0023H	Timer/counter T0L lower 8 bits overflow	T0CNT1	T0CNT0	110H	High/low
6	Timer T1	Internal	002BH	Timer T1L overflow	T1CNT1	T1CNT0	118H	High/low
				Timer T1H overflow	T1CNT3	T1CNT2		
7	SIO0	Internal	0033H	SIO0 end detect	SCON01	SCON00	130H	High/low
8	SIO1	Internal	003BH	SIO1 end detect	SCON11	SCON10	134H	High/low
9	VMS interrupt	Internal	0043H	VMS transfer receive end detect	RFB	RFBENA	160H/161H	High/low
10	Port 3 interrupt (P32INT = 1)	External	004BH	Port 3 "L" level detect	P31INT	P30INT	14EH	High/low

Caution	• The priority ranking indicates which interrupt is handled first if several interrupts are generated simultaneously. The priority ranking changes if specified in the interrupt priority control register (IP).

5.1.2 Interrupt Function Operation

When an interrupt as listed in **Table 5-1** is generated, the corresponding interrupt request flag is set. This indicates to the interrupt control circuit that an interrupt request has occurred.

The interrupt control circuit accepts interrupts in the order of their priority. There are three priority levels: “highest”, “high”, and “low”. To enable interrupts with “high” and “low” priority, the master interrupt flag (IE7) must also be set in addition to the individual interrupt enable flags. IE7 controls interrupts with “high” and “low” priority. If INT0 or INT1 are set to “highest” priority by the interrupt priority control flag (IE1, IE0), interrupt processing occurs regardless of the master interrupt enable flag.

Interrupt sources with a priority ranking from 3 to 9 can be specified as having either “high” or “low” priority by the interrupt priority control register (IP).

When an interrupt is generated, the interrupt control circuit waits until the currently executing instruction is completed. Then it stores the program counter (PC) contents in the stack (in RAM) and executes the interrupt processing routine. This operation uses 2 bytes of stack (RAM) and increments the stack pointer (SP) by +2. After returning from the interrupt processing routine, the stack pointer is decremented by -2.

By executing a RETI instruction at the end of the interrupt processing routine, execution returns to the original program.

Interrupt nesting is possible and can be up to 3 levels deep.

During execution of the RETI instruction or an instruction (MOV, ST, etc.) that writes to one of the special function registers listed below, or while writing to flash memory, interrupt request flag acceptance processing is not performed.

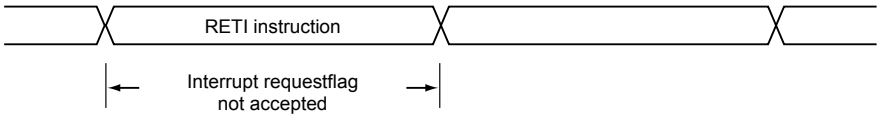


Fig. 5-1 Cycle Without Interrupt Processing

To use the interrupt function, the following Special Function Registers must be operated.

IE, IP, SP [< Caution >], special function registers in the function block that accepts the interrupt

Caution	System program settings are made during a hardware reset. It is not possible to directly manipulate the SP from an application.

5.1.3 Circuit Configuration

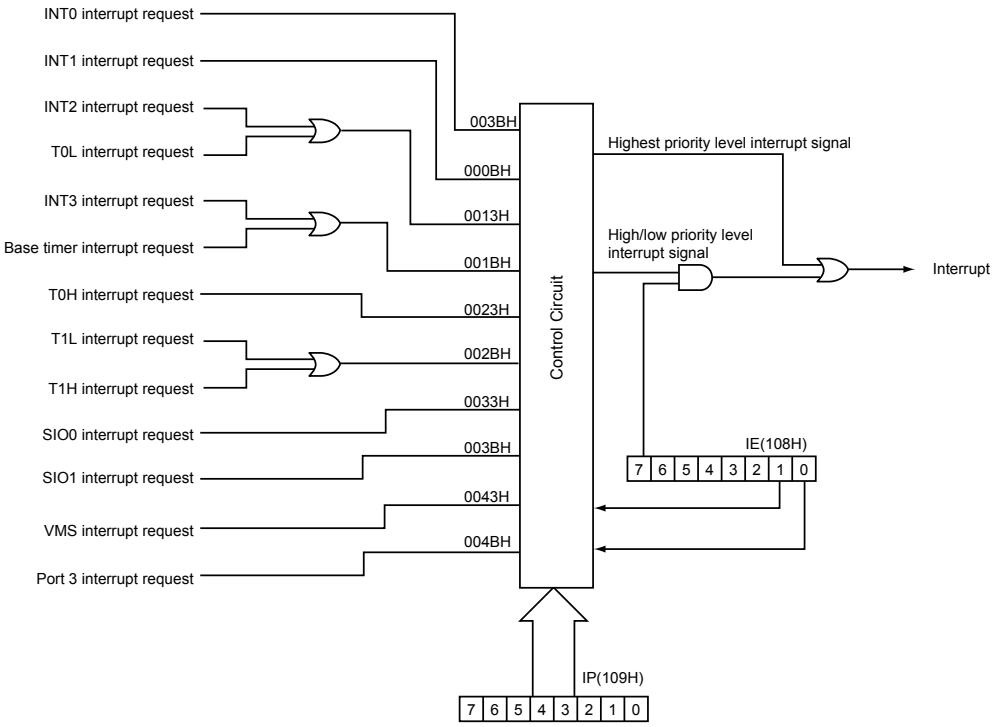


Fig. 5-2 Interrupt Function 1 Block Diagram

5.1.4 Related Registers

Master interrupt enable control register (IE)

For details, refer to the section “Master Interrupt Control Register (IE)” in 4.7 “External Interrupt Functions”.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IE	108H	R/W	IE7	–	–	–	–	–	IE1	IE0
Reset			0	H	H	H	H	H	0	0

Bit name	Function			
IE7 (bit 7)	Master interrupt control (high level, low level)			
	0: All interrupt requests disabled 1: All interrupt requests enabled			
IE1 (bit 1) IE0 (bit 0)	INT0, INT1 interrupt priority control			
	IE1	IE0	INT1 priority level	INT0 priority level
	0	0	Highest	Highest
	1	0	Low	Highest
	X	1	Low	Low

Interrupt Priority Control Register (IP)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP	109H	R/W	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0
Reset			0	–	0	0	0	0	0	0

Bit name	Function
IP7 (bit 7)	Port 3 interrupt priority level setting
	0: Low 1: High
IP5 (bit 5)	SIO1 interrupt priority level setting
	0: Low 1: High
IP4 (bit 4)	SIO0 interrupt priority level setting
	0: Low 1: High
IP3 (bit 3)	T1 interrupt priority level setting
	0: Low 1: High
IP2 (bit 2)	T0H interrupt priority level setting
	17 0: Low 1: High
IP1 (bit 1)	INT3 and base timer interrupt priority level setting
	0: Low 1: High
IP0 (bit 0)	INT2 and T0L interrupt priority level setting
	0: Low 1: High

IP7 (bit 7): port 3 interrupt priority level setting

This bit selects either “high” (1) or “low” (0) for the port 3 interrupt priority level.

When set to “1”, the priority setting for this interrupt is “high”, giving the interrupt higher priority than the INT0 and INT0 interrupts (IE0 = 1).

When reset to “0”, the priority setting for this interrupt is “low”.

IP5 (bit 5): SIO1 interrupt priority level setting

This bit selects either “high” (1) or “low” (0) for the SIO1 interrupt priority level.

When set to “1”, the priority setting for this interrupt is “high”, giving the interrupt higher priority than the INT0 and INT0 interrupts (IE0 = 1).

When reset to “0”, the priority setting for this interrupt is “low”.

IP4 (bit 4): SIO0 interrupt priority level setting

This bit selects either “high” (1) or “low” (0) for the SIO0 interrupt priority level.

When set to “1”, the priority setting for this interrupt is “high”, giving the interrupt higher priority than the INT0 and INT0 interrupts (IE0 = 1).

When reset to “0”, the priority setting for this interrupt is “low”.

IP3 (bit 3): T1 interrupt priority level setting

This bit selects either “high” (1) or “low” (0) for the T1 interrupt priority level.

When set to “1”, the priority setting for this interrupt is “high”, giving the interrupt higher priority than the INT0 and INT0 interrupts (IE0 = 1).

When reset to “0”, the priority setting for this interrupt is “low”.

IP2 (bit 2): T0H interrupt priority level setting

This bit selects either “high” (1) or “low” (0) for the TOH interrupt priority level.

When set to “1”, the priority setting for this interrupt is “high”, giving the interrupt higher priority than the INT0 and INT0 interrupts (IE0 = 1).

When reset to “0”, the priority setting for this interrupt is “low”.

IP1 (bit 1): INT3/base timer interrupt priority level setting

This bit selects either “high” (1) or “low” (0) for the INT3/base timer interrupt priority level.

When set to “1”, the priority setting for this interrupt is “high”, giving the interrupt higher priority than the INT0 and INT0 interrupts (IE0 = 1).

When reset to “0”, the priority setting for this interrupt is “low”.

IP0 (bit 0): INT2/T0L interrupt priority level setting

This bit selects either “high” (1) or “low” (0) for the INT2/TOL interrupt priority level.

When set to “1”, the priority setting for this interrupt is “high”, giving the interrupt higher priority than the INT0 and INT0 interrupts (IE0 = 1).

When reset to “0”, the priority setting for this interrupt is “low”.

5.1.5 Interrupt Priority Ranking

The priority ranking of interrupts is as follows.

Highest level > high level > low level

If multiple interrupts of the same priority level are generated simultaneously, the processing order will be as shown in Table 5-1. The multiple interrupt control circuit controls overlapping interrupts, allowing nesting of “low” level → “high” level → “highest” level interrupts.

Highest level

The external interrupts INT0 and INT1 can be set to the “highest” priority level. Interrupts with this priority level are not controlled by the masking enable flag (IE7).

High level

Interrupt sources other than the external interrupts INT0 and INT1 that correspond to the bits that are set in the interrupt priority control register (IP). Interrupts with this priority level are controlled by the masking enable flag (IE7).

Low level

Interrupt sources INT0 or INT1 for which "low" level is set in IE0 or IE1 and that correspond to the bits that are reset in the interrupt priority control register (IP). Interrupts with this priority level are controlled by the masking enable flag (IE7).

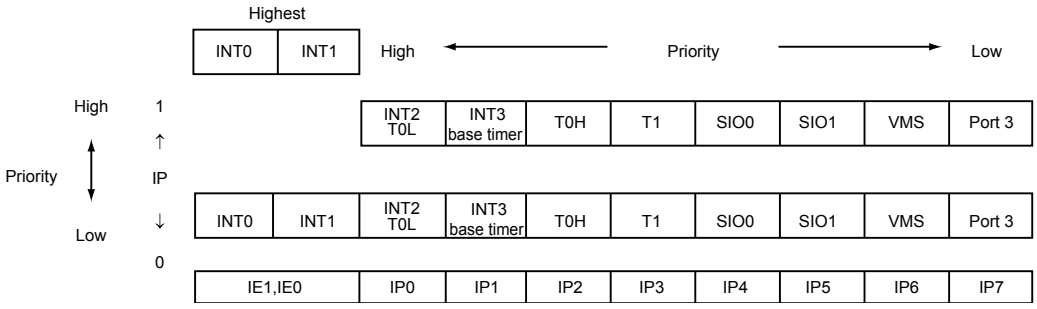


Fig. 5-3 Interrupt Priority Sequence

For example, to give the SIO1 end interrupt higher priority than the INT0 interrupt, set IE0 and IP5 to "1" (IE0 = 1, IP = 00100000B).

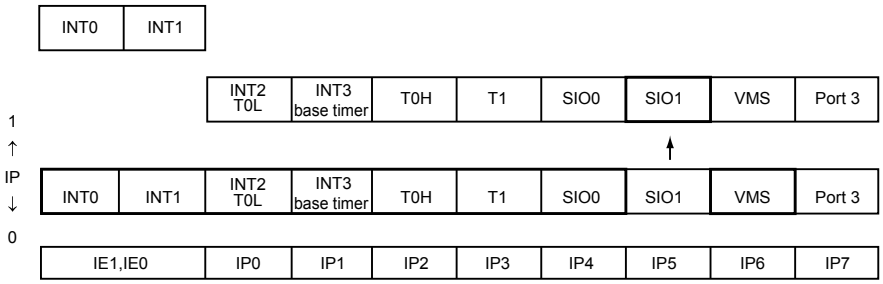


Fig. 5-4 SIO1 Interrupt Priority Sequence

To give the SIO1 end interrupt priority between INT2 and INT0, set IE0, IP5, and IP0 to "1" (IE0 = 1, IP = 00100001B).

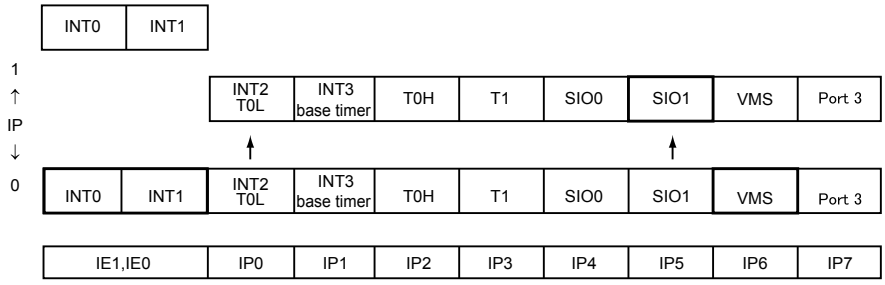


Fig. 5-5 T0L → SIO1 → INT0 Priority Change

Multiple interrupt handling

If a “low” priority interrupt is generated while a “high” priority interrupt routine is being executed, the “low” priority interrupt is accepted after the “high” priority interrupt routine is completed and one instruction was executed.

If an interrupt routine is being executed and an interrupt of the same priority level is generated, the second interrupt request is not accepted.

5.2 System Clock Generation

The VMS incorporates two oscillator circuits: a RC oscillator and a quartz oscillator. Either of these can be selected to supply the system clock. This selection is performed through software.

The oscillation frequencies and cycle clock data for these circuits are shown below.

Oscillator	Frequency	Cycle clock	Purpose	Characteristics
RC oscillator	879.236KHz	6.284 µs	Flash memory access	Uses this clock for flash memory For write operations, set the division ratio to 1/6.
Quartz oscillator	32.768KHz	183.105 µs	Standalone operation clock	Processing speed is reduced for preserving battery power

Note	Because the RC oscillator consumes more battery power, you should normally use the quartz oscillator for the system clock. Due to individual tolerances in CR circuits, correct audio frequency output will not be obtained with PWM. For PWM, use the quartz oscillator. The tolerance range of the RC oscillator is 600 to 1200 kHz. Applications using the oscillator should be designed for a reference frequency of 879.236 kHz.

5.2.1 Features and Functions

- Generation of system clock to be used as reference for instructions
- System clock can be selected through software, using either RC oscillator or quartz oscillator.
- Generation of base timer clock
- RC oscillator can be stopped through software.
- Two system clocks are generated: system clock 1 (S1) for circuit block that operates also in HALT mode, and system clock 2 (S2) for circuit block that stops to operate in HALT mode
  To use the system clock, the following Special Function Registers must be operated.

  OCR, PCON

5.2.2 Circuit Configuration

The system clock generator configuration is shown in **Fig. 5-6.**

Quartz oscillator ... ❷

This quartz oscillator has an oscillation frequency of 32.768 kHz.

RC oscillator ... ❶

This circuit uses capacitors (C) and resistors (R) to generate a frequency of 879.236 kHz. The tolerance range is 600 - 1200 kHz.

System clock selector ... ➥

Bits 4 and 5 of the oscillation control register (OCR) are used to select either the quartz oscillator or the RC oscillator.

System clock generation circuit ... ❸

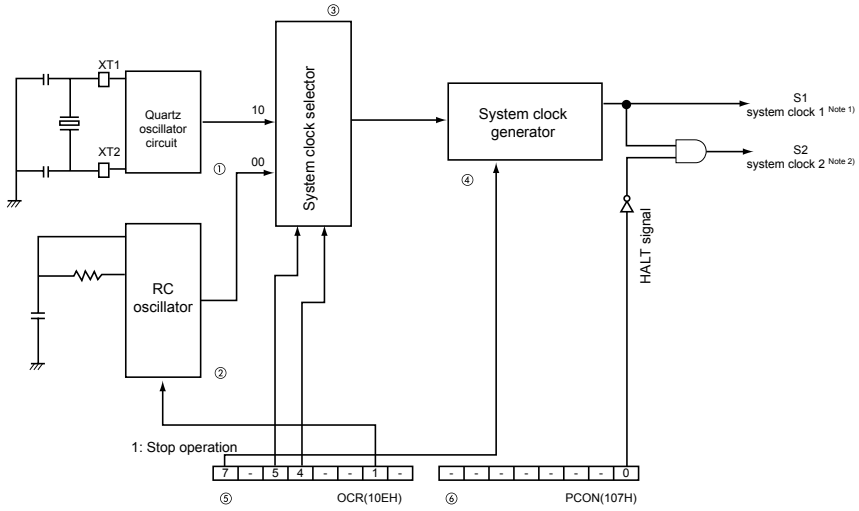
This circuit generates system clock 1 and system clock 2 from the source selected by the system clock selector. System clock 1 (S1) operates while instructions are executed and during HALT mode. System clock 2 (S2) operates while instructions are executed.

Oscillation control register (OCR) ... ❨

Serves for RC oscillator start/stop, system clock source selection, and cycle clock control.

Power control register (PCON) ... ❩

Sets the standby state (HALT mode).



Note 1): System clock 1 (S1) is used for circuits that operate during instruction execution and in HALT mode.
Note 2): System clock 2 (S2) is used for circuits that operate during instruction execution but not in HALT mode.

Fig. 5-6 System Clock Generator Block Diagram

Block status during reset and HALT

Table 5-2 Operation Status at Standby

Block	Condition	
	Reset	HALT
RC oscillator	Operates	Same as when activated
Quartz oscillator	Stops	Same as when activated
System clock generator	Operates	Operates

Caution	Immediately after a hardware reset, the RC oscillator is automatically selected for the system clock. The system BIOS then switches to the quartz oscillator.

5.2.3 Related Registers

Oscillation control register (OCR)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OCR	10EH	R/W	OCR7	–	OCR5	OCR4	–	–	OCR1	–
Reset			0	H	0	0	H	H	0	0

Bit name	Function		
OCR7 (bit 7)	System clock generator control		
	0: Cycle time source is oscillator frequency x 1/2 1: Cycle time source is oscillator frequency x 1/6		
OCR6 (bit 6) OCR4 (bit 4)	System clock select		
	OCR5	OCR4	System clock
	0	0	RC oscillator
	0	1	Prohibited
	1	0	Quartz oscillator
	1	1	Prohibited
			Reset/HALT cancel: RC oscillator
OCR1 (bit 1)	RC oscillator select		
	0: RC oscillator operation start/continue 1: RC oscillator operation stop		

OCR7 (bit 7): system clock generation circuit control

This bit controls whether the clock source for the cycle clock is divided by 12 or by 6.

When set to "1", the cycle clock is 1/6 of the clock source.

When set to "0", the cycle clock is 1/12 of the clock source.

For the VMS, the setting should be as follows.

System clock	OCR7
RC oscillator	OCR7=0/1
Quartz oscillator	OCR7=1

Caution	To use the quartz oscillator, be sure to set this bit to "1". When using the RC oscillator, select the 1/12 division ratio (OCR7 = "0") except when writing to flash memory.

OCR5, OCR4 (bit 5, 4): system clock select

This bit selects the system clock. During a hardware reset, the system clock is automatically set to the RC oscillator.

OCR5	OCR4	System clock
0	0	RC oscillator
0	1	Prohibited
1	0	Quartz oscillator
1	1	Prohibited

OCR1 (bit 1): RC oscillator control

This bit stops/starts the RC oscillator.

When set to "1", the RC oscillator is stopped.

When reset to "0", the RC oscillator operates.

Caution	Note that negative logic is employed here. "0" means start and "1" means stop.

Power control register (PCON)

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	107H	R/W	–	–	–	–	–	–	–	PCON0
Reset			H	H	H	H	H	H	0	0

Bit name	Function
PCON0 (bit 0)	HALT mode control
	0: 1: Set to HALT mode

PCON0 (bit 0): HALT mode control

This bit sets the VMS to the sleep state.

When set to "1", the VMS custom chip goes into HALT mode, causing the VMS to enter the sleep state. Program execution stops at the address where HALT was execute, and the oscillator maintains its current state. The system clock 2 (S2) stops.

The HALT mode is canceled by an interrupt. When the HALT mode is canceled, this bit will be automatically reset.

Directly resetting the bit to "0" does not cause a state change.

Note	In HALT mode, the LCD driver, LCD, timer 0, and timer 1 continue to operate. For details, refer to section 5.3 "Sleep Function".

5.2.4 System Clock Operation Mode

There are three types of system clock.

RC oscillator

This clock is selected in the cases listed below. The oscillation frequency of the RC oscillator is 879.236 kHz.

Caution	Due to the characteristics of RC oscillators, there is a wide variation in frequency. The tolerance range for the VMS is 600 to 1200 kHz.

- Hardware reset
- Power-on (battery replacement)

Quartz oscillator

This oscillator allows implementing a slow processing mode with reduced current consumption, for long-term backup. The oscillation frequency is 32.678 kHz.

When the quartz oscillator is used for the system clock, the RC oscillator can be stopped, using the oscillation control register (OCR). This allows a further reduction in current consumption.

The VMS custom chip enters the HALT mode as shown in the state transition diagram of Fig. 5-7.

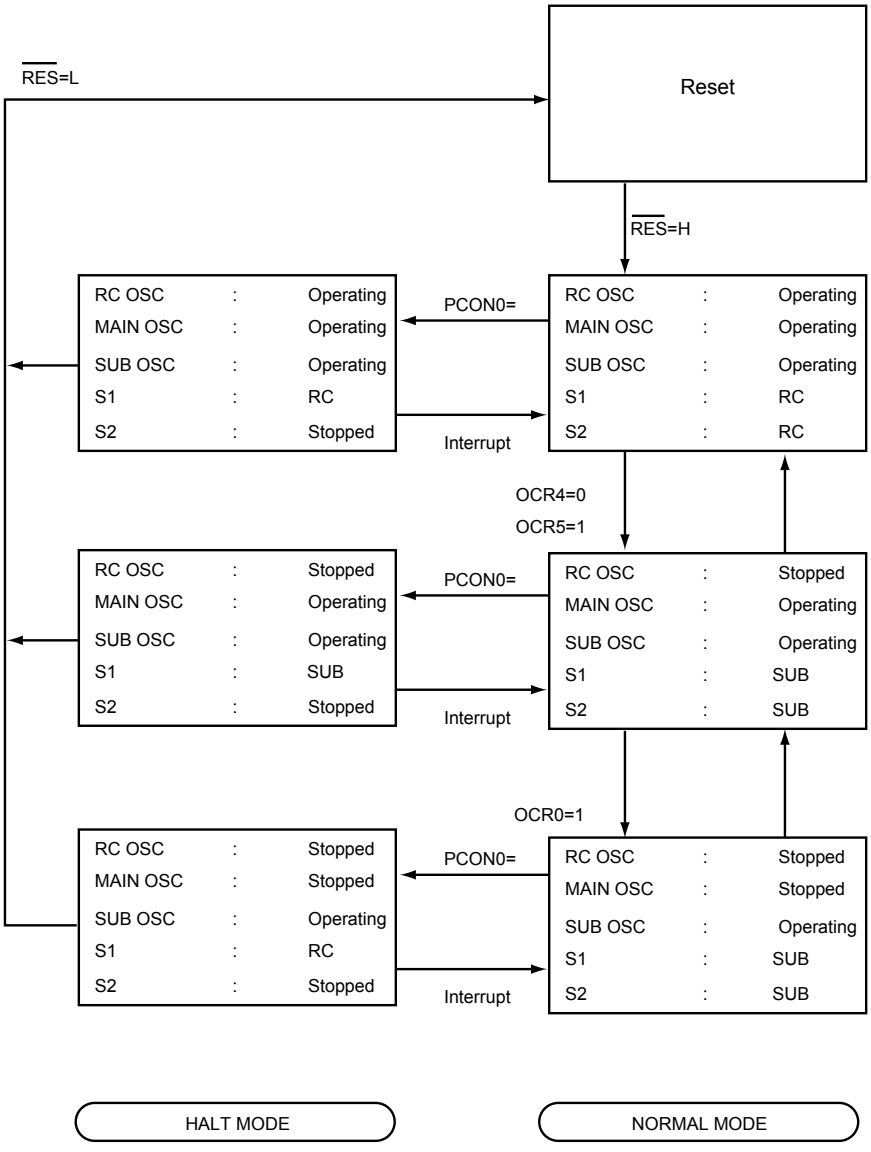


Fig. 5-7 Clock Operation Mode Transition Diagram

RC OSC:	RC oscillator	SUB:	Quartz oscillator frequency
SUB OSC:	Quartz oscillator circuit	PCON0:	power control register bit 0 (HALT control)
S1:	System clock 1	OCR1:	Oscillation control register bit 1
S2:	System clock 2	OCR4:	Oscillation control register bit 4
RC:	RC oscillator frequency	OCR5:	Oscillation control register bit 5

Caution	When switching the system clock to the stopped quartz oscillator, a wait period is required to allow the oscillator to stabilize. For the quartz oscillator in the VMS (32.678 kHz), this wait period is approx. 200 µs.

5.3 Sleep Function

The VMS custom chip provides a HALT mode designed to reduce power consumption during program standby and to delay battery exhaustion. In this mode, the CPU does not execute instructions. The sleep mode of the VMS makes use of the HALT mode.

5.3.1 Related Registers

Power control register (PCON)

For details, refer to the section “Power control register (PCON)” in 5.2 “System Clock Generation”.

Symbol	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	107H	R/W	–	–	–	–	–	–	–	PCON0
Reset			H	H	H	H	H	H	0	0

Bit name	Function
PCON0 (bit 0)	HALT mode control
	0: 1: Set to HALT mode

5.3.2 Standby Operation Status

Table 5-3 Block Operation Status in Standby Mode

Item		HALT mode
Setting method		PCON0=1
Oscillation circuits	CF oscillator	Operation continues
	RC oscillator	
	Quartz oscillator	Operation continues
Internal clock	S1	Operation continues
	S2	Operation stops
CPU		Operation stops
I/O ports		Hold data from immediately before HALT mode
RAM		Hold data from immediately before HALT mode
Base timer		Operation continues
Timer 0		Operation continues
Timer 1		Operation continues
Serial transfer		Operation continues
Interrupt circuits		Operation continues
LCD controller		Operation continues
Remote control circuit		Operation continues
Watchdog timer		Operation continues or stops
Released by		Reset Accepted interrupt request

Note	If the quartz oscillator is selected for the system clock, stop the RC oscillator through software (OCR1 = 1).

5.3.3 HALT Mode

The HALT mode allows stopping program execution while keeping the quartz oscillator and RC oscillator circuits running.

Power consumption can be reduced through intermittent operation of the system by recurringly setting HALT mode and having it released in response to an interrupt.

Setting HALT mode

HALT mode is set by setting bit 0 of the power control register (PCON0).

Releasing HALT mode

HALT mode can be released in one of two ways: through a hardware reset or through receiving an interrupt request.

Releasing HALT mode through hardware reset

When a “Low” level signal is input to the pin, HALT mode is released and the CPU enters the reset state. Returning the pin to “High” level triggers a normal cold start procedure, with the system program executing the VMS initialization routine.

Releasing HALT mode through interrupt request

When the master interrupt enable flag (IE7) and interrupt request enable flag are both set and an interrupt request is generated, the HALT mode is canceled. Subsequently, the processing routine corresponding to the interrupt is called.

If HALT mode was activated in interrupt processing routine A, and the interrupt requested generated while in HALT mode has the same or a lower priority level than interrupt A, the interrupt is not accepted and HALT mode will not be canceled.

Caution	• If the external interrupts INT0 and INT1 are set to the “highest” priority level, the master interrupt enable flag has no effect. • Set the interrupt used to cancel HALT mode to a higher priority level than the interrupt in effect when the system entered the HALT mode.

Table 5-4 HALT Mode Cancel Interrupt Priority Levels

Interrupt level in HALT mode	Interrupt level for HALT mode cancel
Normal level	Low, High, or Highest
Low	High or Highest
High	Highest
Highest	(Cancel by interrupt not possible)

Normal level: No interrupt is present.

5.4 Hardware Reset Function

The hardware reset function serves for initializing the VMS for example when the batteries are replaced or while the unit is running.

5.4.1 External Reset Pin Function

Applying an “L” level signal to the pin for 200 us or more reliably triggers a reset.

Caution	A very narrow “Low” level pulse can also cause a reset.

The configuration of the reset circuit is shown in **Fig. 5-8**.

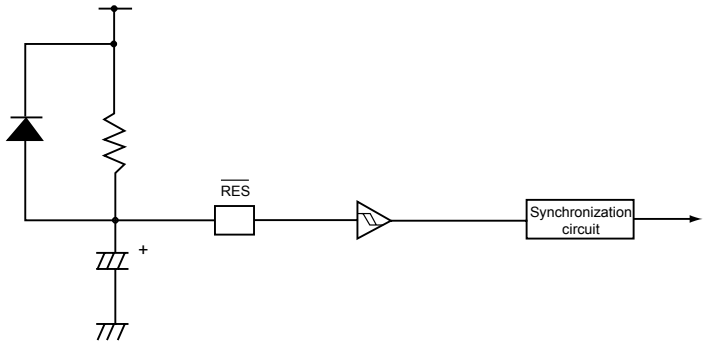


Fig. 5-8 Reset Circuit Block Diagram

5.4.2 Hardware Status During a Reset

When a reset is generated through the pin, the entire hardware is initialized according to the reset signal, which is synchronized with the system clock.

When a reset occurs, the system clock is switched to the RC oscillator. Therefore the hardware is initialized immediately after power-on.

During reset, the program counter (PC) is set to 0000H. The special function registers (SFR) are set to the initial values listed in **Table 5-5**.

The contents of RAM, work RAM, stack pointer, and XRAM are maintained.

Caution	The initial values listed below are the values established by the BIOS after a reset.

Table 5-5 Special Function Register Initial Values

Symbol	Address	R/W	Designation	Initial value	See page
RAM (bank 0)	000H-0FFH	R/W	Data memory	XXXXXXXX (retained after a reset)	43
RAM (bank 1)	000H-0FFH	R/W	Data memory	XXXXXXXX (retained after a reset)	43
ACC	100H	R/W	Accumulator	00000000	50
PSW	101H	R/W	Program status word	00H00000	52
B	102H	R/W	B register	00000000	51
C	103H	R/W	C register	00000000	51
TRL	104H	R/W	Table reference register lower byte	00000000	54
TRH	105H	R/W	Table reference register upper byte	00000000	54
SP	106H	R/W	Stack pointer	XXXXXXXX	53
PCON	107H	R/W	Power control register	HHHHHH00	158
IE	108H	R/W	Master interrupt enable control register	0HHHHH00	138
IP	109H	R/W	Interrupt priority control register	00000000	151
EXT	10DH	R/W	External memory control register	HHHH0000	–
OCR	10EH	R/W	Oscillation control register	0H00HH00	156
T0CNT	110H	R/W	Timer 0 control register	00000000	67
T0PRR	111H	R/W	Timer 0 prescaler data	00000000	71
T0L	112H	R	Timer 0 low	00000000	71
T0LR	113H	R/W	Timer 0 low reload register	00000000	71
T0H	114H	R	Timer 0 high	00000000	72
T0HR	115H	R/W	Timer 0 high reload register	00000000	72
T1CNT	118H	R/W	Timer 1 control register	00000000	83
T1LC	11AH	R/W	Timer 1 low compare data	00000000	86
T1L	11BH	R	Timer 1 low	00000000	85
T1LR		W	Timer 1 low reload data	00000000	85
T1HC	11CH	R/W	Timer 1 high compare data	00000000	87
T1H	11DH	R	Timer 1 high	00000000	86
T1HR		W	Timer 1 high reload data	00000000	86
MCR	120H	W	Mode control register	00000000	127
STAD	122H	R/W	Start address register	00000000	129
CNR	123H	W	Character count register	H0000000	130
TDR	124H	W	Time division register	HH000000	130
XBNK	125H	R/W	Bank address register	HHHHHH00	130
VCCR	127H	W	LCD contrast control register	00000000	131
SCON0	130H	R/W	SIO0 control register	00H00000	108
SBUF0	131H	R/W	SIO0 buffer	00000000	113
SBR	132H	R/W	SIO baud rate generator	00000000	113
SCON1	134H	R/W	SIO1 control register	00000000	111
SBUF1	135H	R/W	SIO1 buffer	00000000	113
P1	144H	R/W	Port 1 latch	00000000	58
P1DDR	145H	W	Port 1 data direction register	00000000	58
P1FCR	146H	W	Port 1 function control register	10111111	59
P3DDR	14DH	W	Port 3 data direction register	00000000	62
P3INT	14EH	R/W	Port 3 interrupt function control register	11111101	62

Symbol	Address	R/W	Designation	Initial value	See page
P7	15CH	R	Port 7 latch	HHHHXXXX	64
I01CR	15DH	R/W	External interrupt 0, 1 control	00000000	135
I23CR	15EH	R/W	External interrupt 2, 3 control	00000000	137
ISL	15FH	R/W	Input signal select	11000000	138
VSEL	163H	R/W	Control register	11111100	143
VRMAD1	164H	R/W	System address register 1	00000000	144
VRMAD2	165H	R/W	System address register 2	HHHHHHH0	144
VTRBF	166H	R/W	Send/receive buffer	XXXXXXXX	144
BTCR	17FH	R/W	Base timer control	01000001	101
RAM (XRAM) (Bank 0)	180H-1FBH	R/W	LCD display memory	XXXXXXXX (retained after a reset)	126
RAM (XRAM) (Bank 1)	180H-1FBH	R/W			
RAM (XRAM) (Bank 2)	180H-185H	R/W			

PART 3

System

BIOS

This section deals with system programs and OS programs contained in the ROM of the VMS, and also with headers used to call OS programs.

Applications must be designed to call OS programs for performing tasks such as reading from and writing to flash memory, reading clock data, and controlling the low- voltage check function.

Chapter 6 Programs in ROM

The ROM of the VMS contains the following programs. Together, these are called the system BIOS.

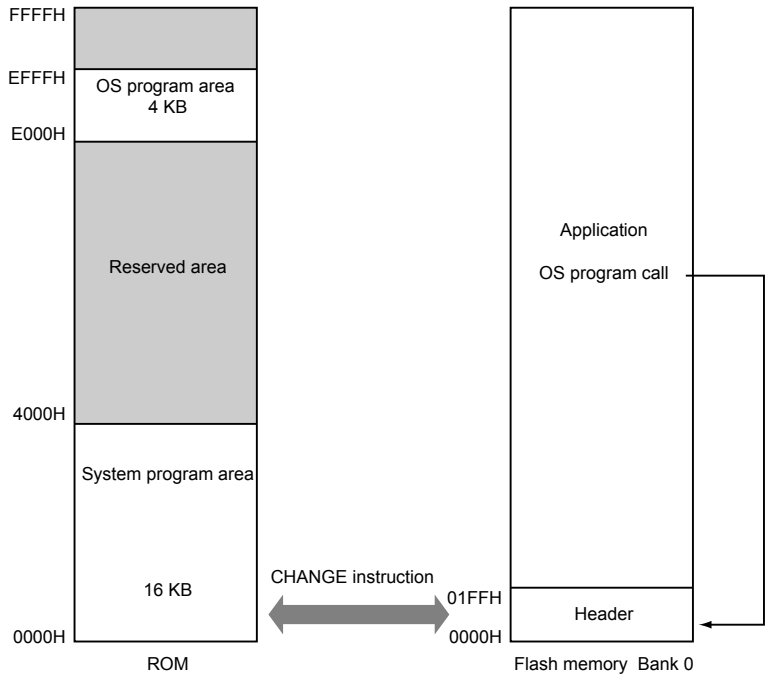


Fig. 6-1 ROM Memory Map

6.1 System Programs

Programs designed for performing VMS file management, clock display, and control functions when connected to the Dreamcast are called system programs.

These programs are permanently placed in ROM. In addition, the ROM also contains the VMS initialization routine that is executed when a hardware reset is performed.

6.2 OS Programs

OS programs perform basic VMS functions such as reading from and writing to flash memory, getting clock data, checking for the low-voltage condition, etc.

These programs correspond to the BIOS in a conventional computer. The programs can be called by an application. Because software interrupts cannot be used, headers defined in the upper region of the flash memory are used to call the programs.

6.3 Headers

The assembler file GHEAD.ASM supplied with the VMS SDK contains the headers.

By using “include” when compiling an application, the headers are placed in the 0000H - 01FFH range of the flash memory. The size can be changed by changing GHEAD.ASM.

The header area defines the interface for switching between a game application and the system application, the interface for calling an OS program from an application, as well as application-specific interrupt vectors.

Caution	It is not possible to obtain the system BIOS version or VMS hardware revision from an application.

Chapter 7 Memory Space

The system BIOS uses the following memory areas.

RAM

The system BIOS uses the RAM bank 0 for processing. The range from 080H to 0FFH of RAM bank 0 is used as stack area.

RAM bank 0 can generally not be accessed by an application, except for reading the internal clock and the low-voltage auto detect flag.

The stack area (080H to 0FFH of RAM bank 0) can also be accessed by an application, but care must be taken not to corrupt the stack.

The 256 bytes of RAM bank 1 can be used by an application.

Special function registers (SFR)

The 100H to 17FH range at the top of RAM is designated as special function registers (SFR). This includes the CPU registers, peripheral device control registers, and other registers.

Work RAM (VTRBF)

When connected to the Dreamcast, the system BIOS uses these 512 bytes as communication buffer. During standalone operation, the memory is available to the application as RAM. Access is possible only through SFR in 1-byte units.

XRAM

This is the RAM for the liquid-crystal display. It corresponds to the video RAM in a conventional computer. XRAM consists of three banks. Banks 0 and 1 can be used by the application to drive the dot-matrix display. Bank 2 serves for the VMS mode icons and cannot be accessed by the application.

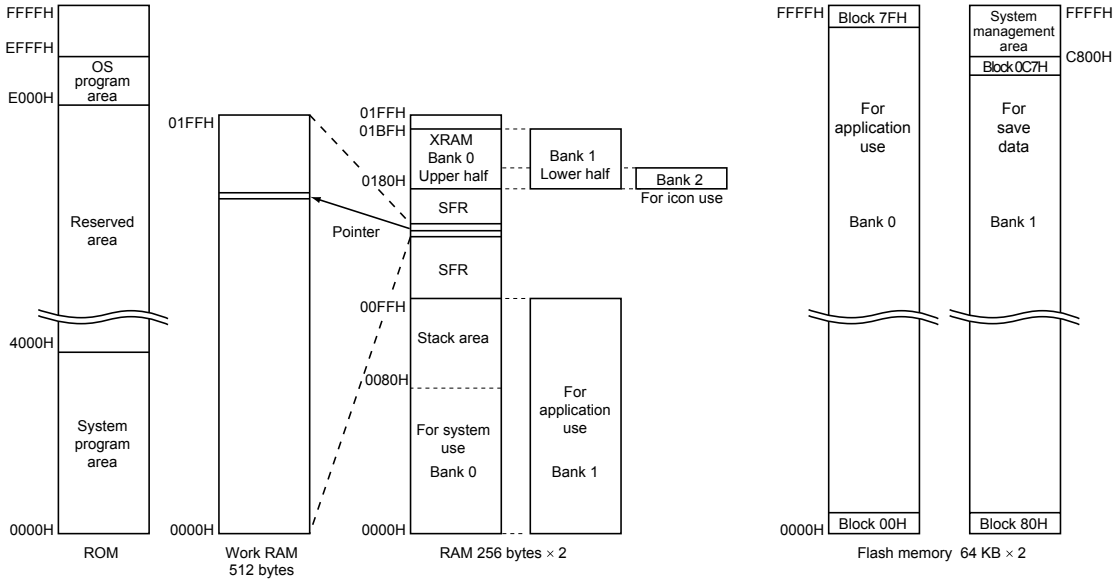


Fig. 7-1 VMS Memory Map

Chapter 8 System BIOS Functions

Applications can call subroutines that are part of OS programs making up the system BIOS. The system BIOS has the following functions.

System initialization

This is performed when the VMS is reset.

Execution mode selection

This includes game data and application management, editing, application startup and shutdown, time display and adjustment.

Mode selection is performed with the MODE button and the A button.

Subroutines

Subroutines can be used by applications. The subroutines allow flash memory access and readout of internal clock data.

1. Flash memory write
2. Flash memory read
3. Flash memory verify
4. Clock count-up timer

Chapter 9 Subroutine Call Procedure

The illustration below shows the operation flow for an application calling a subroutine (part of an OS program), until the return to the application.

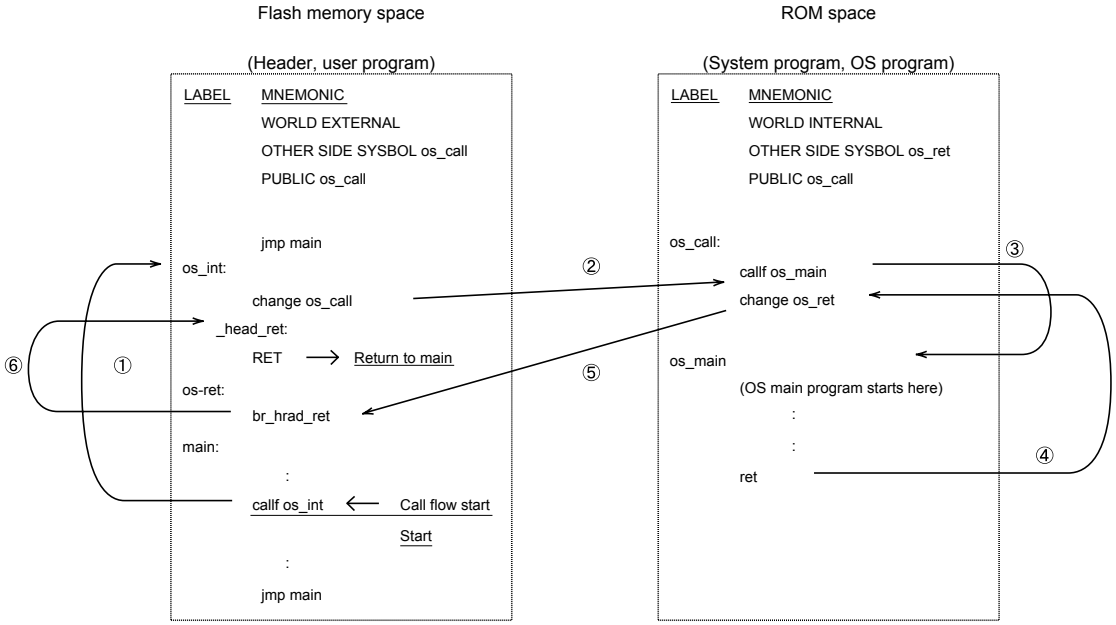


Fig. 9-1 Program Call Flow

9.1 Processing Contents of Labels

The purpose and function of labels in subroutines are explained below.

Reference	When reading this section, you should also refer to “GHEAD.ASM” supplied with the VMS SDK.

Caution	In the explanation below, labels are referred to using preliminary names.

Application (flash memory)

main

Application main program

os_int

Subroutine for switching to ROM space processing.

In the example, calling this subroutine will cause the program to move to ROM space processing. When returning from ROM space, the main program resumes.

This subroutine is provided in the header.

os_ret

Subroutine for returning to the flash memory space.

While processing occurs in ROM space, executing a CHANGE instruction with this label as argument causes a return to the flash memory. After the return, processing jumps to the interrupt return routine provided in the header.

System BIOS (ROM)

os_CALL

This is a return routine for calling an OS program and returning to flash memory. It calls a subroutine in an OS program and returns processing to the flash memory space after completion of the subroutine.

os_main

This is the main OS program. It performs processing for the provided subroutines.

9.2 Interaction Between System BIOS and Application

Assuming that an application in the flash memory is running, the processing flow from calling an OS program until return is described below. Refer also to the sample flow chart.

1. At the point where the running application wants to use an OS program, it calls the os_int subroutine.
   Interrupt processing routines which need to jump to an OS program must contain the os_int subroutine.
2. The CHANGE instruction in the os_int subroutine jumps to the OS program call routine in ROM (os_CALL).
3. The OS program call routine calls the subroutine in the OS main program (os_main).
   OS program processing begins at this point.
4. When OS program processing ends, the RET instruction jumps to the next address of the CALL instruction in the OS program call routine. The OS program call routine always must contain a CHANGE instruction for returning to the flash memory after the OS program CALL instruction.
5. After returning from the OS program subroutine, the CHANGE instruction moves processing to the flash memory. The application provides a subroutine (os_ret) to be called when returning from ROM.
6. This subroutine is called a header. It is supplied as part of the library provided to developers, and must be placed at a fixed location in the application.
7. (In the current example, the headers os_int and os_ret are used.)
8. From the above return routine, processing returns to the os_int subroutine and then to the main program (main) through the RET instruction.

Note	**CHANGE instruction**
	The CHANGE instruction is used to move from the flash memory space to the ROM space and vice versa. Executing the CHANGE instruction causes processing of a program currently running in ROM (or flash memory) to change to flash memory (or ROM). The program counter is set to the specified label (or address).

Chapter 10 Application Shutdown Procedure When MODE Button is Pressed

If the MODE button is pressed while an application is running, processing must be interrupted and the system application mode management screen must be restored immediately.

This section explains the procedure for handing over control from the game application to the system application.

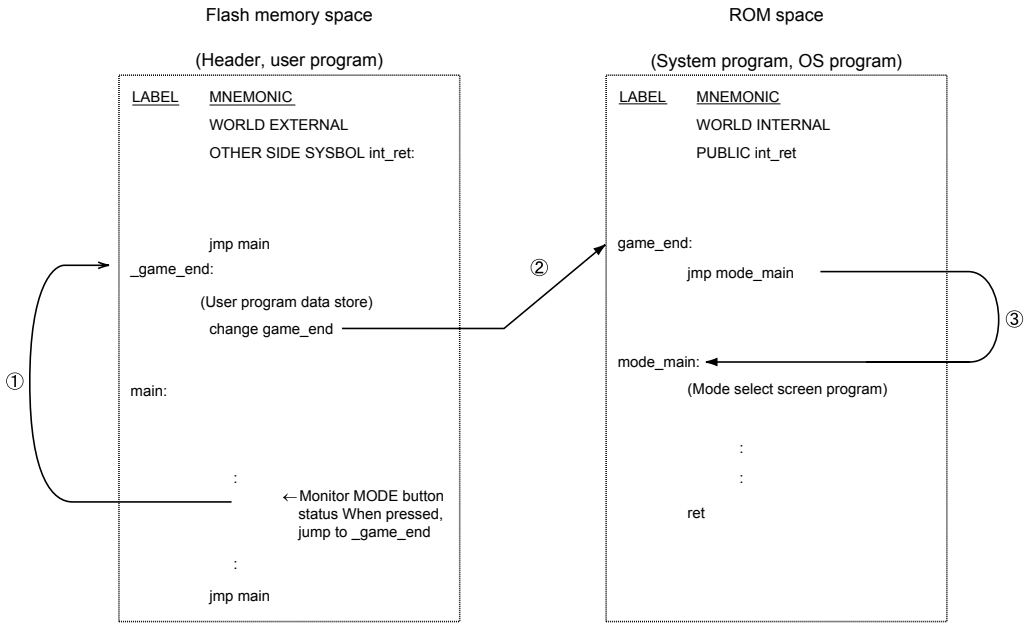


Fig. 10-1 Mode Select Screen Restore Flow

10.1 Processing Contents of Labels

The purpose and function of labels in subroutines are explained below.

Reference	When reading this section, you should also refer to "GHEAD.ASM" supplied with the VMS SDK.

Caution	In the explanation below, labels are referred to using preliminary names, except for game_end.

Application (flash memory)

main

Application main program

The application must be programmed to jump to the OS program return routine listed below when the MODE button is pressed.

_game_end

Subroutine for terminating the application and moving processing to the OS program. If data for the application are to be saved, the code for saving data must be included before jumping to this subroutine.

Caution	The OS program does not save data.

System BIOS (ROM)

game_end

This subroutine serves as a window for returning to the system BIOS after the application ends. The subroutine starts the mode selection program.

Caution	All applications must be designed to speedily CHANGE to game_end when the MODE button is pressed. Any data required for returning to the game at a later point must be saved by the application in flash memory. This must be performed before the CHANGE to game_end. The restore procedure for saved data must also be handled by the application.

mode_main

This is the mode selection program.

Reference	For details on mode selection, refer to chapter 19 "VMS Mode Selection" in the appendix.

10.2 Interaction Between System BIOS and Application

Assuming that an application in the flash memory is running, the processing flow for returning to the mode selection screen is described below. Refer also to the sample flow chart.

1. At the point where the MODE button is pressed while the application is running, processing jumps to the _game_end subroutine.
   The CHANGE instruction in the _game_end subroutine hands processing over to the program in ROM. If data for the application are to be saved, the code for saving data must be included before executing the CHANGE instruction in the _game_end subroutine.

Caution	Do not use the port 3 interrupt for detecting a MODE button press. If the port 3 interrupt processing routine contains a _game_end subroutine, the BIOS does not operate normally.

2. When processing jumps from the application to the _game_end subroutine, the CHANGE instruction in the _game_end subroutine moves processing to the game_end subroutine in the ROM program.
3. After processing has changed from the flash memory to the game_end subroutine, the mode selection program is started.

Chapter 11 VMS Initialization

The VMS is automatically initialized in the following cases.

1. Unit is connected to Dreamcast, and Dreamcast is turned ON.
2. Reset switch on VMS is pressed.
3. Batteries are inserted.

The initialization routine includes the following steps.

Clear RAM

The entire contents of RAM (banks 0 and 1) are set to 00H. The contents of XRAM are not changed.

A hardware reset is applied to all registers, and then software initialization is carried out. For information on initial register values after a hardware reset, refer to section 5.4.2 “Reset” in the “Hardware” part of this manual.

Set system clock and cycle time

The system clock is set to the quartz oscillator. The cycle time is set to 1/6 of the system clock.

Set base timer

The 14-bit base timer mode is selected, and the base timer clock is set to the quartz oscillator.

The base timer interrupt is enabled and counting starts.

Set master interrupt

The master interrupt is enabled.

Set LCD driver

The LCD controller is activated, and the LCD clock is set to 1/2 of the clock signal input to the LCD driver.

The display start address is set to 00H in XRAM, and the character count register and time division register are set.

Then the LCD is set to ON.

Set port 1

All bits of port 1 are set to input. Bit 7 of port 1 is set to audio output.

Caution	After initialization, bit 7 of port is in input mode. Therefore it must be again set to output mode by the application.

Bits 5 to 0 of port 1 (VMS serial interface) are set to function as synchronous serial interface.

Set port 3

All bits of port 3 are pulled up and set to input mode. Port 3 interrupt source generation is enabled, and interrupt request is enabled.

Initialize Maple bus interface circuit

The Maple bus interface circuit is initialized.

Set work RAM

The work RAM is set to be available to applications.

Chapter 12 Subroutine Reference

This section explains the subroutines contained in the system BIOS.

12.1 Flash Memory Access Functions

The following subroutines are provided for flash memory access.

Flash memory page data read

Reads 128 bytes of data from the flash memory space.

Flash memory write

Writes 128 bytes of data to the flash memory space.

Flash memory verify

Verifies data written to the flash memory space.

Caution	When performing flash memory access, the application must switch the system clock to the RC oscillator. Do not switch the clock within GHEAD.ASM.

12.2 Subroutine Use Precautions

When accessing the flash memory space, observe the following precautions.

The VMS incorporates three system clock types that can be used to synchronize instruction execution cycles.

In standalone operation, the quartz oscillator is used, but for flash memory write access, the clock must be switched to the RC oscillator with the 1/6 division ratio setting before calling the flash memory access subroutine. For flash read or verify access, the RC oscillator with any division ratio setting can be used.

When switching to the RC oscillator, inhibit all interrupts including the base timer.

After the subroutine is completed, enable all interrupts and switch back to the previously used clock. The proper timing for clock switching is shown below.

System clock oscillator source	Frequency	Instruction cycle time
RC oscillator	879.236 kHz	6.284 μs
Quartz oscillator	32.768 kHz	183.105 μs

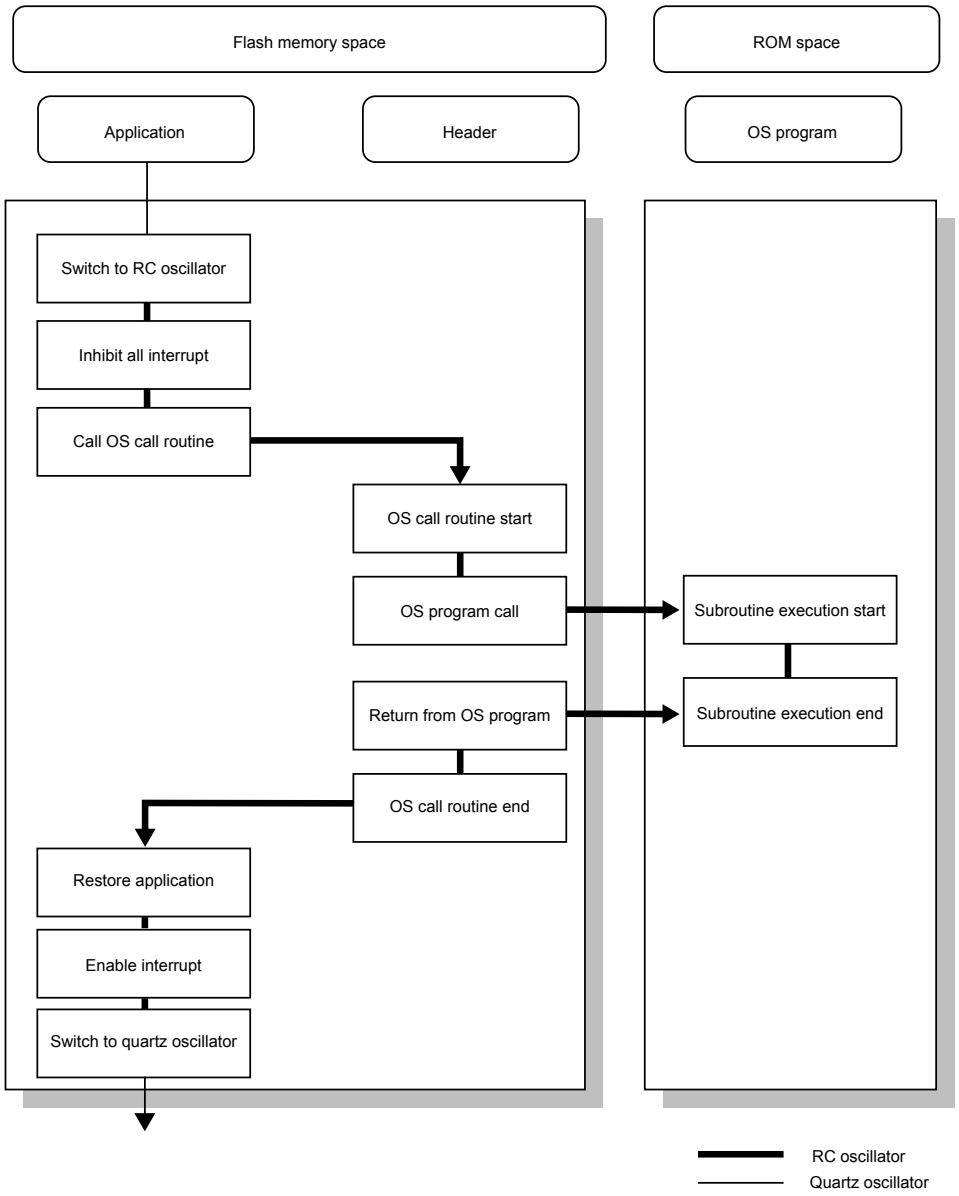


Fig. 12-1 Clock Switching Flow for Flash Memory Access

Flash memory routines

fm_prd_ex(ORG 0120H)

Flash memory page data read

Arguments

Flash memory read start upper address: fmadd_h (RAM bank 1 07EH)

Flash memory read start lower address: fmadd_l (RAM bank 1 07FH)

Flash memory read bank address: fmbank (RAM bank 1 07DH)

Return values

Read data (128 bytes): RAM bank 1 080H to 0FFH

Broken registers

When this subroutine is called, the following registers are broken.

ACC, TRL, TRH, r0

Function

Reads one continuous page of data (128 bytes) starting at the specified address in flash memory.

Description

By calling this subroutine, one continuous page of data (128 bytes) can be read from the flash memory. For using the subroutine, the following settings must be made beforehand.

Caution	This subroutine does not return error information. Make sure that arguments are specified correctly. Call this subroutine only when STAD is set to 00H. If called while STAD is set to a value other than 00H, a part of the screen will be rewritten.

RAM bank settings

1. Set RAM bank to "1" (set bit 1 of PSW to "1")

Note	For information on the PSW register, refer to section 3.8 "Program Status Word (PSW)" in the "Hardware" part of this manual.

Set flash memory read start address

2. Set upper address (8 bits): fmadd_h (RAM bank 1 07EH)
3. Set lower address (8 bits): fmadd_l (RAM bank 1 07FH)

Set flash memory read bank

4. Set read flash memory bank to bank 0.
   Set RAM bank 1 07DH to 00H.

Caution	If another value than the above is set, normal operation is not assured.

The read data are written to RAM bank 1 080H to 0FFH.

Caution	When making the read settings, observe the following points. • Data spanning two pages cannot be read. The read start address must be set to the start of a page. The start address of a page can be determined as follows. Start address value (2 bytes) = 080H × page number (0 to 511) Because data are read in units of one page, bits 0 - 6 of the lower address must be set to "0". If set to an address different from the start address, normal operation is not assured. • The read data overwrite the original location in RAM.

Note	**About pages** The flash memory space is divided into pages of 128 bytes each. The flash memory is managed using these page units. Because the size of one bank in the flash memory is 64 KB, a bank contains 512 pages.

The operation when fm_prd_ex executes is shown below.

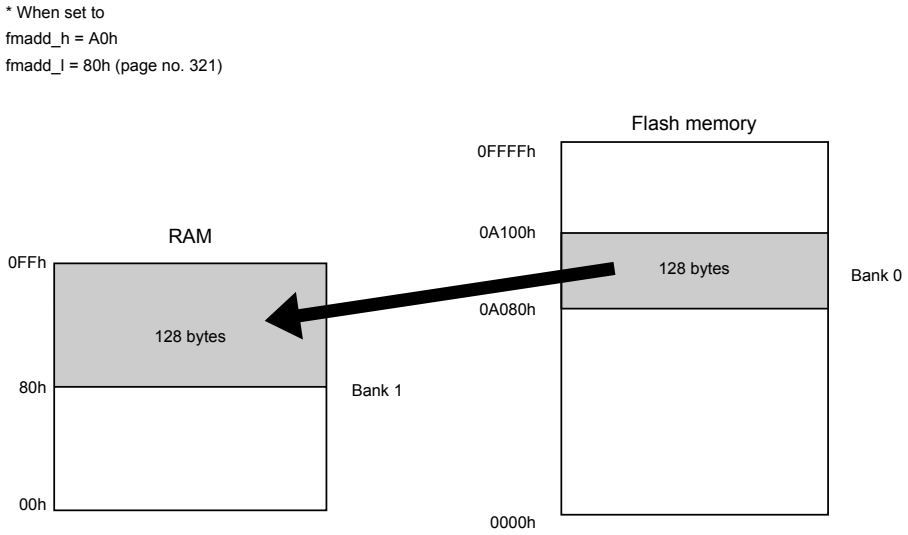


Fig. 12-2 Data Transfer With fm_prd_ex

fm_wrt_ex(ORG 0100H)

Flash memory data write

Arguments

Flash memory write start upper address: fmadd_h (RAM bank 1 07EH)

Flash memory write start lower address: fmadd_l (RAM bank 1 07FH)

Flash memory write bank address: fmbank (RAM bank 1 07DH)

Flash memory write data (128 bytes): RAM bank 1 080H to 0FFH

Return values

ACC At normal end, 00H is set in the accumulator. At abnormal end, 0FFH is set in the accumulator.

Broken registers

When this subroutine is called, the following registers are broken.

ACC, B, C, TRL, TRH, r0

Function

Writes one continuous page of data (128 bytes) starting at the specified address in flash memory.

Description

By calling this subroutine, one continuous page of data (128 bytes) can be written to the flash memory. For using the subroutine, the following settings must be made beforehand.

1. Set bit 1 of PSW to "1", to select RAM bank 1.
2. Store data to write to flash memory in RAM bank 1 080H - 0FFH.
3. Set 07DH of RAM bank 1 to 00H, to set write flash memory to bank 0.

Caution	Flash memory bank 1 may not be accessed by applications. Do not write any data to this bank.

4. Write flash memory upper address (8 bits) to 07EH of RAM bank 1, and write lower address (8 bits) to 07FH of RAM bank 1.

Caution	When making the write settings, observe the following points. • Data spanning two pages cannot be written. The write start address must be set to the start of a page. The start address of a page can be determined as follows. Start address value (2 bytes) = 080H × page number (0 to 511) Because data are written in units of one page, bits 0 - 6 of the lower address must be set to "0". If set to an address different from the start address, normal operation is not assured. • Call only when STAD is set to 00H. If called while STAD is set to a value other than 00H, a part of the screen will be rewritten. • Switch the system clock to RC oscillator with the 1/6 division ratio setting.

The operation when fm_wrt_ex executes is shown below.

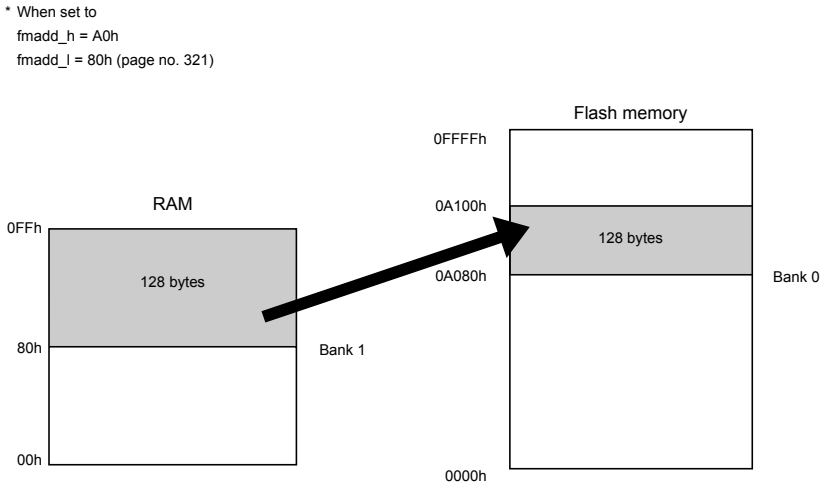


Fig. 12-3 Data Transfer With fm_wrt_ex

fm_vrf_ex(ORG 0110H)

Flash memory page data verify

Arguments

Flash memory verify start upper address: fmadd_h (RAM bank 1 07EH)

Flash memory verify start lower address: fmadd_l (RAM bank 1 07FH)

Flash memory verify bank address: fmbank (RAM bank 1 07DH)

Flash memory verify data (128 bytes): RAM bank 1 080H - 0FFH

Return values

The verify result is set in the accumulator. If there was no mismatch, 00H is set. If there was a mismatch, a value other than 00H is set.

Broken registers

When this subroutine is called, the following registers are broken.

ACC, TRL, TRH, r0

Function

After writing data to flash memory, this function checks whether the data were written correctly. Use the function after using fm_wrt_ex to write tot flash memory.

Description

This subroutine compares the 128 bytes of data specified when calling fm_wrt_ex to the data actually written to the flash memory.

Caution	Call this subroutine only when STAD is set to 00H. If called while STAD is set to a value other than 00H, a part of the screen will be rewritten.

This subroutine may therefore only be called immediately after calling the fm_wrt_ex subroutine.

When calling the subroutine, the same arguments as for the fm_wrt_ex subroutine must be supplied. If different arguments are supplied, data verify will not yield correct results.

After calling the subroutine, 00H is set in the accumulator if all 128 bytes of data were matched. If there was a mismatch, a value other than 00H is set.

The operation when fm_vrf_ex executes is shown below.

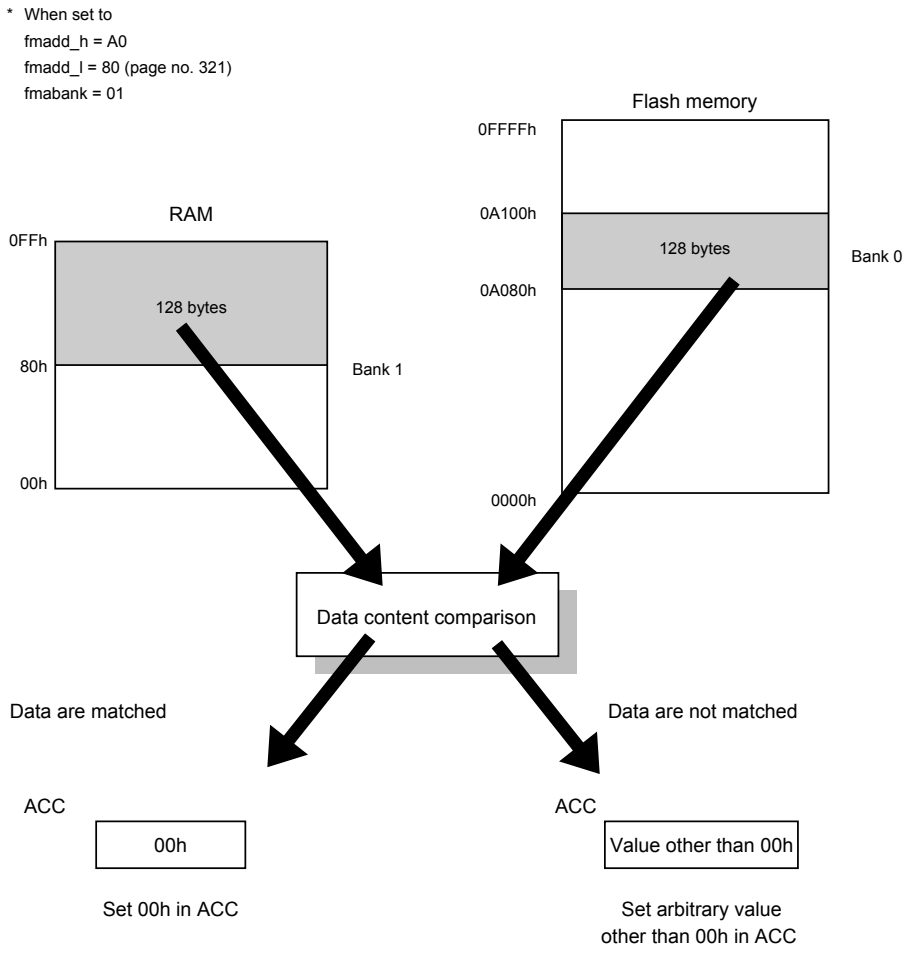


Fig. 12-4 Execution of fm_vrf_ex

Clock Function

timer_ex

Clock count-up timer

Arguments

None

Return values

Year :	year_h	(RAM bank 0 017H_18H)
Month :	mon_h	(RAM bank 0 019H)
Day :	day_h	(RAM bank 0 01AH)
Hour :	hour_h	(RAM bank 0 01BH)
Minute :	min_h	(RAM bank 0 01CH)
Second :	sec_h	(RAM bank 0 01DH)

The year data use 2 bytes. The upper byte is stored in 17H and the lower byte in 18H. RAM bank 0 017H is assigned to year_h. When accessing address 018H, the address for year_h + 1 must be accessed.

Caution	The time data obtained by this subroutine are all in hexadecimal format. They must be converted to decimal format by the application. The work area comprises a BCD date area, but this area is not updated by timer_ex.

Function

Gets current date and time data and places them in the specified area of RAM bank 0.

Description

This subroutine is a timer/counter using the base timer interrupt.

Caution	The base timer interrupt uses timer_ex. For using the base timer interrupt from the application, call the user-side handler immediately after the label timer_ex_exit in GHEAD.ASM. Call this subroutine after generating a base timer interrupt source and jumping to the interrupt vector. At this time, be sure to reset the base timer interrupt source (BTCR1 = 0). If the interrupt source is not reset, the clock function will not work properly.

Chapter 13 Low Battery Voltage Auto Detection

An automatic low battery voltage detection function which displays a warning message on the LCD is incorporated in the system BIOS.

Actions which cause high power consumption such as flash memory data write or data transfer to another VMS may falsely trigger the warning. Therefore the detection function should be disabled before carrying out such actions.

Low battery voltage auto detection flag

This flag specifies whether low battery voltage auto detection is performed or not. Applications can manipulate this flag.

Caution	Be sure to set the low battery voltage auto detection flag to 0FFH (off) before having an application perform one of the following functions. Otherwise the high current consumption caused by these functions may falsely trigger the low battery voltage auto detection. • Communication with other VMS via serial interface • Writing to flash memory

Address

06EH (RAM bank 0) Low battery voltage auto detection flag

When set to 00H, low battery voltage auto detection is carried out. When set to 0FFH, low battery voltage auto detection is not carried out.

Caution	Do not set the flag to values other than 00H or 0FFH.

Operation

The low battery voltage auto detection function monitors the battery voltage. When it falls below a certain threshold, the function interrupts the currently running program and displays a warning message for 3 seconds on the LCD.

Description

The low battery voltage auto detection function comprises code for both voltage detection and message display. When the low battery voltage auto detection flag is set to 00H, these functions are carried out automatically, regardless of the operation status of the VMS. When the low battery voltage auto detection flag is set to 0FFH, all functions related to automatic low battery voltage detection are turned off.

Caution	Programs to save data in flash memory when low voltage is detected should monitor the low voltage detection flag (bit 1 of port 7) rather than using the low voltage interrupt. If the low voltage interrupt is used, triggering may occur while writing to flash memory or during serial communication, although the battery voltage is still sufficient.

Note	For information on the low voltage detection flag, refer to section 4.1.3 “Port 7” in the “Hardware” part of this manual.

Chapter 14 List of Defined Variables

The following variables are required for using the OS program BIOS.

Time data variables

Symbol	Address (RAM bank)	Contents	Comment
year	010H (Bank 0)	Year (BCD 4 digits)	Not updated by timer_ex
mon	012H (Bank 0)	Month (BCD 2 digits)	Not updated by timer_ex
day	013H (Bank 0)	Day (BCD 2 digits)	Not updated by timer_ex
hour	014H (Bank 0)	Hours (BCD 2 digits)	Not updated by timer_ex
min	015H (Bank 0)	Minutes (BCD 2 digits)	Not updated by timer_ex
sec	016H (Bank 0)	Seconds (BCD 2 digits)	Not updated by timer_ex
year_h	017H (Bank 0)	Year (HEX 4 digits)	
mon_h	019H (Bank 0)	Month (HEX 2 digits)	
day_h	01AH (Bank 0)	Day (HEX 2 digits)	
hour_h	01BH (Bank 0)	Hours (HEX 2 digits)	
min_h	01CH (Bank 0)	Minutes (HEX 2 digits)	
sec_h	01DH (Bank 0)	Seconds (HEX 2 digits)	
sec_f	01EH (Bank 0)	Work area	Use prohibited
leaf_f	01FH (Bank 0)	Work area	Use prohibited

Caution	The BCD data fields year, mon, day, hour, min, sec are a work area for applications accessing the clock function. Because timer_ex does not perform BCD conversion, this work area is not updated.

Low battery voltage detection variables

Symbol	Address (RAM bank)	Content
None	06EH (Bank 0)	Low battery auto detect flag 00: Auto detect on FFH: Auto detect off

To perform low-battery checking from the application without using auto detect, check bit 1 of port 7.

Flash memory variables

Symbol	Address (RAM bank)	Content
Fmbank	07DH (bank 1)	Specify flash memory bank
fmadd_h	07EH (bank 1)	Flash memory address (upper 8 bits)
fmadd_l	07FH (bank 1)	Flash memory address (lower 8 bit)

PART 4

PWM Sound Source

The VMS can use the internal timer (timer 1) to produce sound. This can be done either using the 8-bit pulse generator or variable bit length pulse generator (9 to 16 bits). For both of these cases, timer 1 is used. Normally, the 8-bit pulse generator is used.

To use the PWM sound source, switch the system clock to the quartz oscillator..

Chapter 15 Sound Output Method

This section describes the VMS sound output method. The sound output uses timer 1.

15.1 Timer 1 Outline

This section describes the timer 1 used for VMS sound output. The timer 1 incorporated in the VMS is a 16-bit timer with the following four functions.

Mode 0: 8 bit reload timer × 2 channels

Mode 1: 8 bit reload timer + 8 bit pulse generator

Mode 2: 16 bit reload timer

Mode 3: Variable bit length pulse generator (9 to 16 bits)

VMS uses mode 1 for producing sound. For information on use of the other modes, refer to section 4.3 “Timer 1 (T1)” in the “Hardware” part of this manual.

15.1.1 Timer 1 Block Configuration

The timer 1 used for VMS sound output has the following block configuration.

Timer 1 low (T1L) ... ❷

This is an 8 bit reload timer which uses the cycle clock or 1/2 the cycle clock as clock.

At T1L overflow, the T1LR data are reloaded, and sent to T1L if T1LRUN (T1CNT bit 6) is set to “0”.

Timer 1 low comparator (T1LC) ... ➧

This comparator consists of the 8 bit timer 1 low comparison data register (T1LC) and an 8 bit data comparison circuit. It serves to compare the T1L and T1LC data.

Timer 1 high (T1H) ... ➥

This is an 8 bit reload timer which uses the cycle clock or the T1L overflow as clock.

At T1H overflow, T1HR data are reloaded, regardless of whether T1HRUN (T1CNT bit 7) is reset.

Timer 1 high comparator (T1HC) ... ❸

This comparator consists of the 8 bit timer 1 high comparison data register (T1HC) and an 8 bit data comparison circuit. It serves to compare the T1H and T1HC data.

Timer 1 control register (T1CNT) ... ❹

Controls timer 1 mode setting and interrupt control.

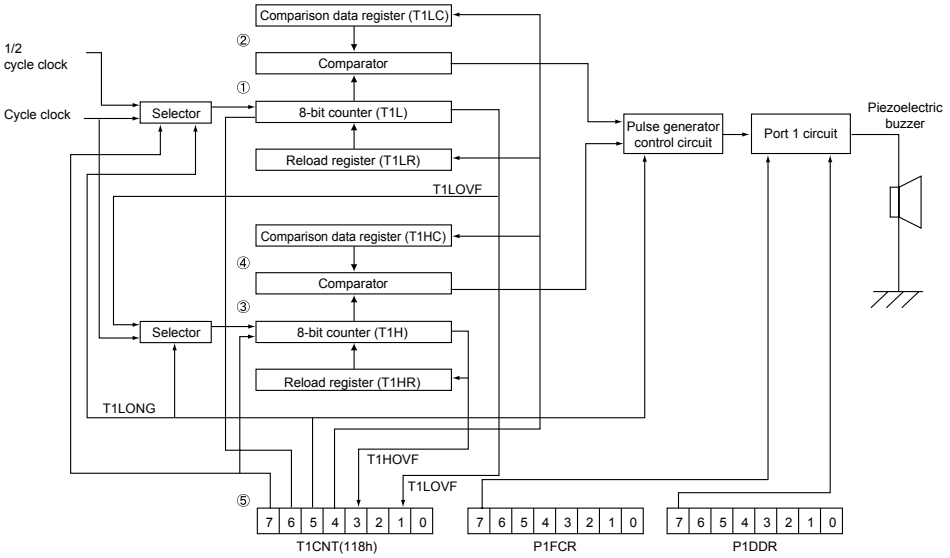


Fig. 15-1 Timer 1 Block Diagram

15.1.2 Related Registers

To control timer 1, the following registers must be controlled.

Symbol	Address	Function
T1L	11BH	Timer 1 lower counter register
T1LR	11BH	Timer 1 lower reload register
T1LC	11AH	Timer 1 lower comparison data register
T1CNT	118H	Timer 1 control register
P1	114H	Port 1 latch register
P1DDR	145H	Port 1 data direction register
P1FCR	146H	Port 1 control register
OCR	10EH	Oscillation control register

For details on the above registers, refer to the “Hardware” part of this manual.

15.1.3 Mode Setting

This section explains how to set timer 1 to the mode required for sound output (mode 1). The following four registers are required for the setting.

T1CNT (bit 5: T1LONG)

P1 (bit7: P17)

P1DDR (bit 7: P17DDR)

P1FCR (bit 7: P17FCR)

The register values for each mode are shown below. The available cycle clock setting for each mode is also shown.

Mode	Clock frequency	T1LONG	P17FCR	P17DDR	P17
1	Tcyc	0	1	1	0

Tcyc in the table indicates the clock cycle.

To use the VMS sound output function, be sure to set the system clock to the quartz oscillator. At other settings, correct sound output may not be obtained.

The cycle clock is as follows.

System clock 32.768 kHz (Tcyc = 183.105 μs)

For information on setting the system clock, refer to the “Hardware” part of this manual.

Caution	**Problems when using other system clock settings**
	If the sound output function is used while the system clock is set to the RC oscillator, the tolerances of the RC oscillator will adversely affect the sound output. Be sure to use the quartz oscillator.

15.2 8 Bit Counter Mode

This section explains VMS sound output using the 8 bit counter mode. For information on basic operation, refer to the “Hardware” part of this manual.

15.2.1 Output Waveform and Parameter Setting

This section describes the waveform of the signal that can be output in 8 bit counter mode and the available parameters.

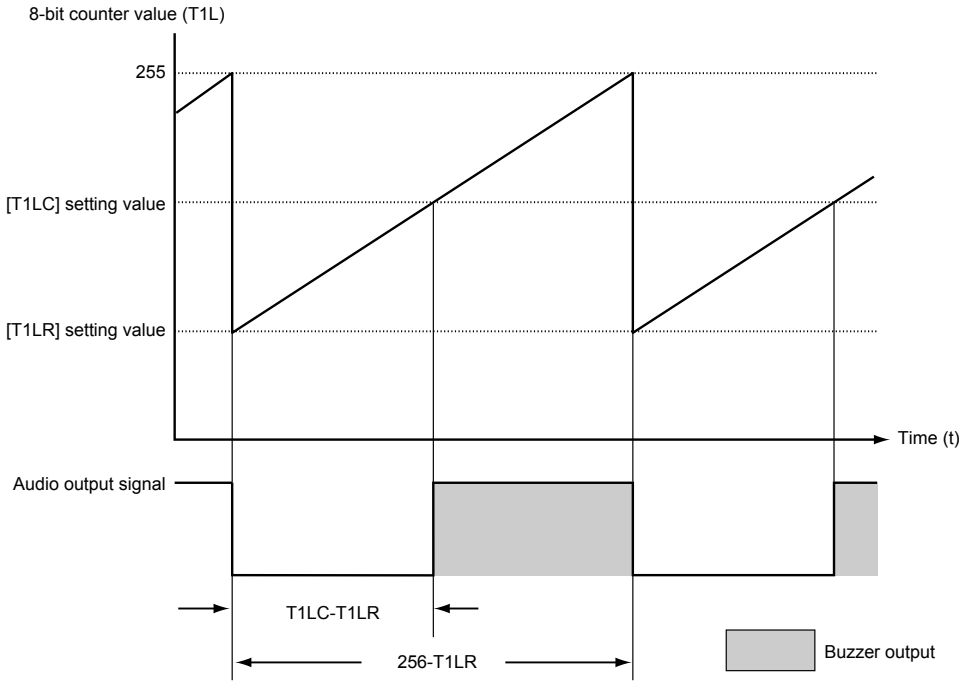


Fig. 15-2 Output Waveform

15.2.2 8 Bit Counter Mode Setting

This section describes sound output in 8 bit counter mode. To use sound output in 8 bit counter mode, make the following settings.

1 Output waveform setting

Set the parameters (T1LR, T1LC) to obtain the desired waveform. Use equations (1) and (2) shown below to determine the waveform.

Audio output signal “Low” level pulse width (decimal) = (T1LC setting value ~ T1LR setting value) × Tcyc ... (1)

Output signal frequency (decimal) = (256 - T1LR setting value) × Tcyc ... (2)

Tcyc: cycle clock

For details in output waveform parameter settings, refer to section 15.2.1 “Output Waveform and Parameter Setting”.

Timer 1 mode setting

Set timer 1 to mode 1. The following four registers are required for mode setting.

T1CNT (bit 5: T1LONG)

P1 (bit 7: P17)

P1DDR (bit 7: P17DDR)

P1FCR (bit 7: P17FCR)

The register values for mode 1 are shown below.

Mode	T1LONG	P17FCR	P17DDR	P17
1	0	1	1	0

Sound start

Timer 1 (lower 8 bit) starts to count, and sound is output. To control timer 1 count start/stop, make the following setting.

1. Waveform parameter update
   Set T1CNT bit 4 (ELDT1C)to “1”. If this setting is not made, the waveform parameter set with T1LR, T1LC does not become effective.
   If the waveform parameter is changed while T1CNT bit4 is “1”, the new parameter setting becomes effective immediately.
2. Timer 1 count start
   Set T1CNT bit 6 (T1LRUN) to “1”.

Sound stop

To stop sound output in 8 bit counter mode, make the following settings.

Set timer 1 (T1L) count stop flag (T1CNT bit 6) to “0”.

The waveform parameter can be changed also during sound output (while timer 1 operates). To continuously output a different frequency, change the output waveform parameter without stopping timer 1. T1CNT bit 4 (ELDT1C) should always be “1” in this case.

15.2.3 Frequency Characteristics

The frequency characteristics of VMS sound output are shown below.

The T1LR value indicates the setting value for the available frequency range. For details on the correlation between the value of T1LR and the output frequency, refer to section 15.2.4 “Output Frequency Table”

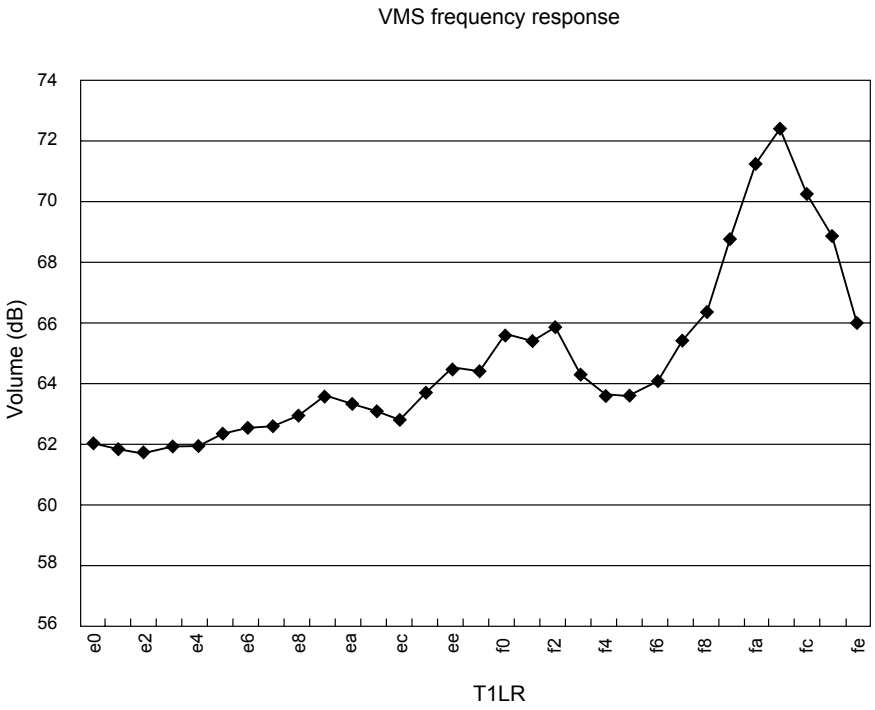


Fig. 15-3 Frequency Response Characteristics

15.2.4 Output Frequency Table

The following table shows the frequencies (theoretical values) available with the 32.768 kHz system clock.

Due to buzzer characteristics, not all frequencies can actually be output. Use the recommended frequencies indicated in the table.

The sound output signal “Low” level pulse width is set to 1/2 (duty cycle 50%) of the output signal cycle.

T1LR(hex)	T1LC(hex)	Frequency (Hz)	T1LR(hex)	T1LC(hex)	Frequency (Hz)	T1LR(hex)	T1LC(hex)	Frequency (Hz)	T1LR(hex)	T1LC(hex)	Frequency (Hz)
00	80	21.346	40	94	28.461	80	A8	42.691	C0	E0	85.383
01	80	21.429	41	A0	28.610	81	C0	43.027	C1	E0	86.738
02	81	21.514	42	A1	28.760	82	C1	43.369	C2	E1	88.137
03	81	21.599	43	A1	28.913	83	C1	43.716	C3	E1	89.582
04	82	21.684	44	A2	29.066	84	C2	44.068	C4	E2	91.075
05	82	21.771	45	A2	29.222	85	C2	44.427	C5	E2	92.618
06	83	21.858	46	A3	29.379	86	C3	44.791	C6	E3	94.215
07	83	21.946	47	A3	29.538	87	C3	45.161	C7	E3	95.868
08	84	22.034	48	A4	29.698	88	C4	45.537	C8	E4	97.580
09	84	22.123	49	A4	29.861	89	C4	45.920	C9	E4	99.354
0A	85	22.213	4A	A5	30.025	8A	C5	46.309	CA	E5	101.194
0B	85	22.304	4B	A5	30.191	8B	C5	46.705	CB	E5	103.103
0C	86	22.395	4C	A6	30.358	8C	C6	47.108	CC	E6	105.086
0D	86	22.488	4D	A6	30.528	8D	C6	47.517	CD	E6	107.147
0E	87	22.580	4E	A7	30.699	8E	C7	47.934	CE	E7	109.290
0F	87	22.674	4F	A7	30.873	8F	C7	48.358	CF	E7	111.520
10	88	22.769	50	A8	31.048	90	C8	48.790	D0	E8	113.843
11	88	22.864	51	A8	31.226	91	C8	49.230	D1	E8	116.266
12	89	22.960	52	A9	31.405	92	C9	49.677	D2	E9	118.793
13	89	23.057	53	A9	31.587	93	C9	50.133	D3	E9	121.433
14	8A	23.155	54	AA	31.770	94	CA	50.597	D4	EA	124.193
15	8A	23.253	55	AA	31.956	95	CA	51.070	D5	EA	127.081
16	8B	23.352	56	AB	32.144	96	CB	51.552	D6	EB	130.107
17	8B	23.453	57	AB	32.334	97	CB	52.043	D7	EB	133.280
18	8C	23.554	58	AC	32.527	98	CC	52.543	D8	EC	136.612
19	8C	23.656	59	AC	32.721	99	CC	53.053	D9	EC	140.115
1A	8D	23.759	5A	AD	32.919	9A	CD	53.573	DA	ED	143.802
1B	8D	23.862	5B	AD	33.118	9B	CD	54.104	DB	ED	147.689
1C	8E	23.967	5C	AE	33.320	9C	CE	54.645	DC	EE	151.791
1D	8E	24.073	5D	AE	33.524	9D	CE	55.197	DD	EE	156.128
1E	8F	24.179	5E	AF	33.731	9E	CF	55.760	DE	EF	160.720
1F	8F	24.287	5F	AF	33.941	9F	CF	56.335	DF	EF	165.590
20	90	24.395	60	B0	34.153	A0	D0	56.922	E0	F0	170.765
21	90	24.504	61	B0	34.368	A1	D0	57.521	E1	F0	176.274
22	91	24.615	62	B1	34.585	A2	D1	58.133	E2	F1	182.149
23	91	24.726	63	B1	34.806	A3	D1	58.758	E3	F1	188.430
24	92	24.839	64	B2	35.029	A4	D2	59.397	E4	F2	195.160
25	92	24.952	65	B2	35.255	A5	D2	60.049	E5	F2	202.388
26	93	25.066	66	B3	35.484	A6	D3	60.716	E6	F3	210.172
27	93	25.182	67	B3	35.716	A7	D3	61.399	E7	F3	218.579
28	94	25.299	68	B4	35.951	A8	D4	62.096	E8	F4	227.687
29	94	25.416	69	B4	36.189	A9	D4	62.810	E9	F4	237.586
2A	95	25.535	6A	B5	36.430	AA	D5	63.540	EA	F5	248.385
2B	95	25.655	6B	B5	36.674	AB	D5	64.288	EB	F5	260.213
2C	96	25.776	6C	B6	36.922	AC	D6	65.053	EC	F6	273.224
2D	96	25.898	6D	B6	37.173	AD	D6	65.837	ED	F6	287.604
2E	97	26.021	6E	B7	37.428	AE	D7	66.640	EE	F7	303.582
2F	97	26.146	6F	B7	37.686	AF	D7	67.463	EF	F7	321.440
30	98	26.272	70	B8	37.948	B0	D8	68.306	F0	F8	341.530
31	98	26.398	71	B8	38.213	B1	D8	69.171	F1	F8	364.299
32	99	26.527	72	B9	38.482	B2	D9	70.057	F2	F9	390.320
33	99	26.656	73	B9	38.755	B3	D9	70.967	F3	F9	420.345
34	9A	26.787	74	BA	39.032	B4	DA	71.901	F4	FA	455.373
35	9A	26.919	75	BA	39.313	B5	DA	72.860	F5	FA	496.771
36	9B	27.052	76	BB	39.598	B6	DB	73.844	F6	FB	546.448
37	9B	27.186	77	BB	39.887	B7	DB	74.856	F7	FB	607.165
38	9C	27.322	78	BC	40.180	B8	DC	75.896	F8	FC	683.060
39	9C	27.460	79	BC	40.478	B9	DC	76.965	F9	FC	780.640
3A	9D	27.598	7A	BD	40.780	BA	DD	78.064	FA	FD	910.747
3B	9D	27.738	7B	BD	41.086	BB	DD	79.195	FB	FD	1092.896
3C	9E	27.880	7C	BE	41.398	BC	DE	80.360	FC	FE	1366.120
3D	9E	28.023	7D	BE	41.714	BD	DE	81.559	FD	FE	1821.494
3E	9F	28.167	7E	BF	42.034	BE	DF	82.795	FE	FF	2732.240
3F	9F	28.313	7F	BF	42.360	BF	DF	84.069	FF	FF	5464.481

:Recommended settings

Fig. 15-4 Waveform Parameters and Output Frequencies

Chapter 16 Sample Program

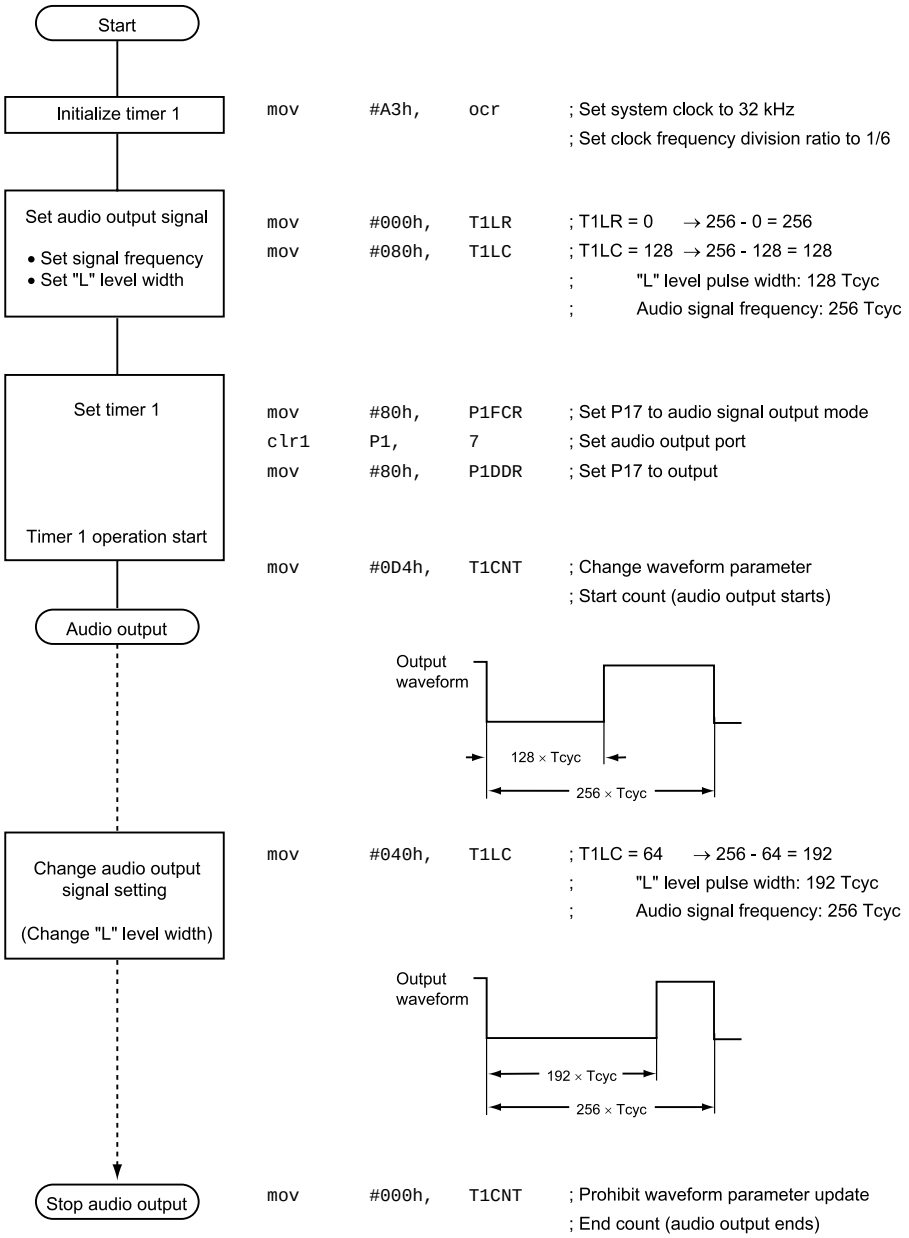


Fig. 16-1 Flow Chart and Program

PART 5

Appendix

Chapter 17 Variable Bit Length Pulse Generator

This section provides additional information about the equation shown in "Mode 3: Variable bit length pulse generator (9 to 16 bits)" of section 4.3 "Timer 1 (T1)" in the "Hardware" part of this manual.

- Large interval P cycle Tp
  $Tp = 2^{[BIT]} \times Ttc$
- Total "L" level pulse width <sigma>TL of large interval P
  $\Sigma TL = (2^{[BIT]} \times T1LC]/256 + [T1HC]) \times Ttc$

- T1HC, T1LC are decimal.
- [T1HC] is the effective number of bits.

[BIT] is the bit length to be set. The number of small intervals T in the large interval P is determined by the bit length. It is set by the timer 1 high reload register (T1HR) and timer 1 low reload register (T1LR). Set the T1LR to 00H. For a 9-bit pulse generator, the setting is [BIT] = 9, large interval P cycle Tp = 29 × Ttc = 512 Ttc. Since the small interval T is 256 Ttc (Ttc: pulse signal clock cycle), T is repeated 2 times in the large interval P.

Table 17-1

Bit length	Small interval T repeat count	Pulse generator bit length setting (binary)		"L" level pulse width setting (binary)	
		T1HR value	T1LR value	T1HR value (upper bits)	T1LR value (lower bits)
16	256	0000 0000	0000 0000	XXXX XXXX	XXXX XXXX
15	238	1000 0000	0000 0000	XXXX XXXX	XXXX XXX0
14	64	1100 0000	0000 0000	XXXX XXXX	XXXX XX00
13	32	1110 0000	0000 0000	XXXX XXXX	XXXX X000
12	16	1111 0000	0000 0000	XXXX XXXX	XXXX 0000
11	8	1111 1000	0000 0000	XXXX XXXX	XXX0 0000
10	4	1111 1100	0000 0000	XXXX XXXX	XX00 0000
9	2	1111 1110	0000 0000	XXXX XXXX	X000 0000

(X: 0 or 1) indicates effective bits

[T1LR] specifies the timer 1 low comparison data register (T1LC) value. It sets the "Low" level pulse width common to all small intervals. If [T1LC] = 3, a "Low" level pulse of 3 Ttc is applied to all small intervals, and the "Low" level pulse width of the large interval is 6 Ttc (if [T1HC] = 0).

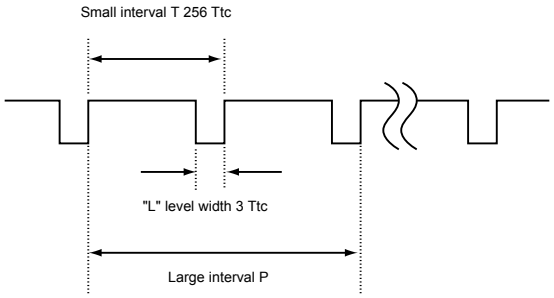


Fig. 17-1 "L" Level Width for T1LC = 3

[T1HC] specifies the effective value of the timer 1 high comparison data register (T1HC), determining the number of "Low" level pulses to be added in the large interval P.

The position of the small intervals to which "Low" level pulses are added by [T1HC] is shown in the illustration below. If [T1HC] = 1, a "Low" level pulse of 1 Ttc is added to the intervals marked "1". If [T1HC] = 6, a "Low" level pulse of 1 Ttc is added to the intervals marked "2" and "4".

| | | |

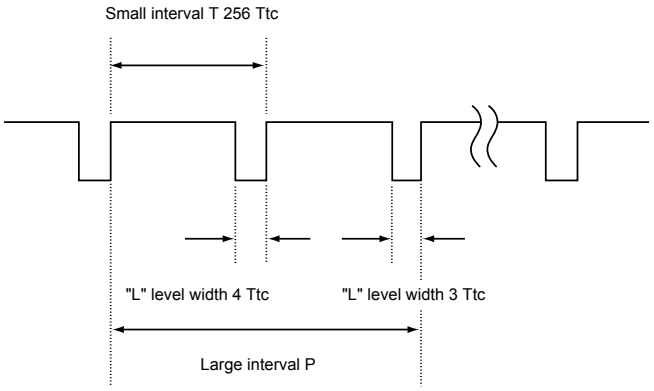


Fig. 17-2 "L" Level Width (7 Ttc) for T1HC = 1

Next, consider the 11-bit pulse generator.

[BIT] = 11, large interval P cycle $T^P = 2^{11} \times$ Ttc = 2048 Ttc. Since the small interval T is 256 Ttc (Ttc: pulse signal clock cycle), T is repeated 8 times in the large interval P.

Fig. 17-3 shows the change in output waveform caused by T1HC when T1LC is constant.

For [T1LC] = 10 (0AH), [T1HC] = 0, a "Low" level pulse of 10 Ttc is output for all small intervals.

For [T1HC] = 3, three small intervals with a "Low" level pulse width of 11 Ttc will be generated in the large interval, as shown by (B). For [T1HC] = 4, four small intervals with a "Low" level pulse width of 11 Ttc will be generated in the large interval, as shown by (C). For [T1HC] = 5, five small intervals with a "Low" level pulse width of 11 Ttc will be generated in the large interval, as shown by (D).

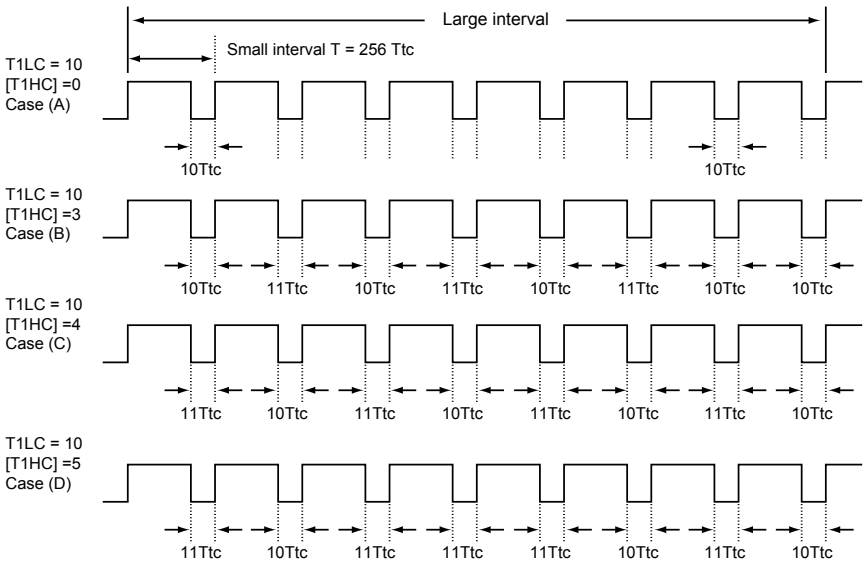


Fig. 17-3 T1HC Value and 11 Ttc "L" Level Pulse Count

Chapter 18 Symbol Table

Caution	The initial values are the values established by the BIOS after a reset.

Symbol	Address	R/W	Designation	Default value	See page
RAM (bank 0)	000H-0FFH	R/W	Data memory	XXXXXXXX (stored at reset)	43
RAM (bank 1)	000H-0FFH	R/W	Data memory	XXXXXXXX (stored at reset)	43
ACC	100H	R/W	Accumulator	00000000	50
PSW	101H	R/W	Program status word	00H00000	52
B	102H	R/W	B register	00000000	51
C	103H	R/W	C register	00000000	51
TRL	104H	R/W	Table reference register lower byte	00000000	54
TRH	105H	R/W	Table reference register upper byte	00000000	54
SP	106H	R/W	Stack pointer	XXXXXXXX	53
PCON	107H	R/W	Power control register	HHHHHH00	158
IE	108H	R/W	Master interrupt enable control register	0HHHHH00	138
IP	109H	R/W	Interrupt priority control register	00000000	151
EXT	10DH	R/W	External memory control register	HHHH0000	_
OCR	10EH	R/W	Oscillation control register	0H00HH00	156
T0CNT	110H	R/W	Timer 0 control register	00000000	67
T0PRR	111H	R/W	Timer 0 prescaler data	00000000	71
T0L	112H	R	Timer 0 low	00000000	71
T0LR	113H	R/W	Timer 0 low reload data	00000000	71
T0H	114H	R	Timer 0 high	00000000	72
T0HR	115H	R/W	Timer 0 high reload data	00000000	72
T1CNT	118H	R/W	Timer 1 control register	00000000	83
T1LC	11AH	R/W	Timer 1 low comparison data	00000000	86
T1L	11BH	R	Timer 1 low	00000000	85
T1LR		W	Timer 1 low reload data	00000000	85
T1HC	11CH	R/W	Timer 1 high comparison data	00000000	87
T1H	11DH	R	Timer 1 high	00000000	86
T1HR		W	Timer 1 high reload data	00000000	86
MCR	120H	W	Mode control register	00000000	127
STAD	122H	R/W	Start address register	00000000	129
CNR	123H	W	Character count register	H0000000	130
TDR	124H	W	Time division register	HH000000	130
XBNK	125H	R/W	Bank address register	HHHHHH00	130
VCCR	127H	W	LCD contrast control register	00000000	131

Symbol	Address	R/W	Designation	Default value	See page
SCON0	130H	R/W	SIO0 control register	00H00000	108
SBUF0	131H	R/W	SIO0 buffer	00000000	113
SBR	132H	R/W	SIO0 baud rate generator	00000000	113
SCON1	134H	R/W	SIO1 control register	00000000	111
SBUF1	135H	R/W	SIO1 buffer	00000000	113
P1	144H	R/W	Port 1 latch	00000000	58
P1DDR	145H	W	Port 1 data direction register	00000000	58
P1FCR	146H	W	Port 1 function control register	10111111	59
P3DDR	14DH	W	Port 3 data direction register	00000000	62
P3INT	14EH	R/W	Port 3 interrupt function control register	11111101	62
P7	15CH	R	Port 7 latch	HHHHXXXX	64
I01CR	15DH	R/W	External interrupt 0, 1 control	00000000	135
I23CR	15EH	R/W	External interrupt 2, 3 control	00000000	137
ISL	15FH	R/W	Input signal select	11000000	138
VSEL	163H	R/W	Control register	11111100	143
VRMAD1	164H	R/W	System address register 1	00000000	144
VRMAD2	165H	R/W	System address register 2	HHHHHHH0	144
VTRBF	166H	R/W	Send/receive buffer	XXXXXXXX	144
BTCR	17FH	R/W	Base timer control	01000001	101
RAM (XRAM) (Bank 0)	180H-1FBH	R/W	LCD memory	XXXXXXXX (stored at reset)	126
RAM (XRAM) (Bank 1)	180H-1FBH	R/W			
RAM (XRAM) (Bank 2)	180H-185H	R/W			

Chapter 19 VMS Mode Selection

The operation modes available for the VMS and the selection principles are shown in the illustration below.

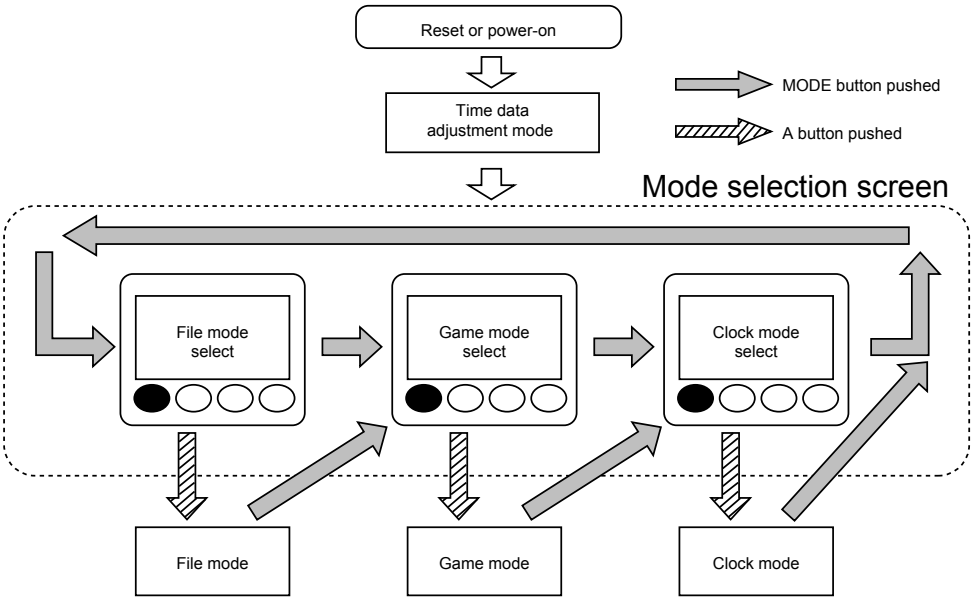


Fig. 19-1 Mode Select Transition Diagram

Mode selection screen

The mode selection screen serves to select and execute one of the three modes of the VMS.

With each push of the MODE button, the mode is switched. During selection, the corresponding LCD icon flashes. Pressing the A button then executes the selected mode.

File mode

This mode serves for managing and editing game data and applications stored on the VMS.

While the file mode is active, pressing the MODE button switches to the mode selection screen.

Game mode

This mode serves for running an application transferred to the VMS.

The application must be programmed to restore the mode selection screen when the MODE button is pressed.

Caution	The return to the mode selection screen from the game mode is not supported by the BIOS. It must be incorporated in the application.

Clock mode

In this mode, the current date and time are shown.

While the clock mode is active, pressing the MODE button switches to the mode selection screen. Keeping the A button depressed while pressing a direction key activates the time set mode.

Chapter 20 Calculation of Battery Life

Because the VMS incorporates two system clocks with different current consumption, battery life will differ with different applications. Another important factor that influences battery life is whether a program is designed only for standalone operation or for use in conjunction with another VMS.

The instruction manual of the VMS therefore contains only the specification "With new lithium batteries, the built- in clock will operate continuously for about 130 days if only the OS is used."

This section contains information about how to calculate expected battery life based on the source code of an application. Developers should use these data to determine expected battery life, and this information should be conveyed to the user. Since the price of a lithium battery of the type used in the VMS is approximately 280 Japanese yen (as of November 1998 in Japan), each replacement will cost the user about 560 yen for the two batteries. Programs should therefore be designed so as to consume as little power as possible.

20.1 Methods for Enhancing Battery Life

The VMS incorporates two system clocks, an RC oscillator (879.236 kHz; tolerance range 600 to 1200 kHz), and a quartz oscillator (32.768 kHz).

The RC oscillator increases processing speed compared to the quartz oscillator, but it also consumes more power. An important consideration when programming an application is therefore how to use the RC oscillator as little as possible without impairing playability.

Writing to XRAM or flash memory always requires use of the RC oscillator, which will increase the RC oscillator load. When two VMS units are connected for serial transfer, the load will also increase. Depending on the circuit configuration, not only the transfer but also the connection process itself can consume considerable power.

With regard to power consumption, take the following points into account when coding an application.

- Use the RC oscillator as little as possible.
- Avoid frequent writing accesses to flash memory.
- Keep communication sessions short.
- Clearly specify the connect/disconnect timing for communication applications.
- Do not redraw the LCD frequently.

20.2 Oscillator Circuit and Current Consumption

The following table shows the current consumption of the two oscillator circuits.

Oscillator circuit	Clock frequency	Current consumption
RC oscillator	879.236 kHz	2.600 mA
Quartz oscillator	32.768 kHz	0.610 mA

Caution	An RC oscillator inherently is subject to frequency tolerances. The oscillator used in the VMS has a tolerance range of 600 - 1200 kHz. Consequently, there will also be differences in current consumption. The battery life calculations in this section assume a frequency of 1000 kHz. The tolerance of the quartz oscillator is 50 ppm - 30 ppm from the center frequency of 32.768 kHz.

When using the RC oscillator, select a cycle time of 1/12 the system clock, except when writing to the flash memory.

20.3 Oscillation Control Register

The oscillation control register (OCR) serves for selecting the oscillator circuit, start/stop control, and setting the system clock division ratio. By effectively managing these settings, battery life can be extended.

20.3.1 System Clock Division Ratio Setting

The OCR7 bit serves for setting the cycle time to 1/12 or 1/6 of the system clock.

When set to “1”, the cycle time is 1/12 of the system clock. For drawing the LCD image, use the RC oscillator with the 1/12 setting.

When reset to “0”, the cycle time is 1/6 of the system clock. At this setting, current consumption is about 1.2 times higher than at the 1/12 setting. For writing to the flash memory, use the RC oscillator with the 1/6 setting.

Division ratio	Quartz oscillator cycle time	RC oscillator cycle time
1/12 (OCR7 = 0)	366.210 µs	12.568 µs
1/6 (OCR7 = 1)	183.105 µs	6.284 µs

20.3.2 Oscillator Circuit Selection

The OCR5 and OCR4 bits serve for selecting the oscillator circuit.

When set to OCR5 = 0, OCR4 = 0, the RC oscillator is used for the system clock.

When set to OCR5 = 1, OCR4 = 0, the quartz oscillator is used for the system clock.

Caution	OCR5 and OCR4 only select the oscillator circuit to be used for the system clock. To reduce power consumption, it is also necessary to perform start/stop control of the RC oscillator. When switching the system clock to the stopped oscillator circuit, insert a wait of at least 300 microseconds.

20.3.3 Oscillator Circuit Start/Stop

The OCR1 bit serves for starting and stopping the RC oscillator.

When set to “1”, the RC oscillator is stopped.

When reset to “0”, the RC oscillator starts or continues to operate.

The RC oscillator should be stopped while using the quartz oscillator. When the RC oscillator operates, current consumption is about 1.1 times higher.

Caution	The quartz oscillator is used by the clock and therefore should not be stopped.

20.4 Calculating Battery Life

Use the “Battery Life Calculation Chart” at the end of this section to calculate expected battery life, as follows.

The lithium battery (CR2032) used in the VMS has a capacity of 210 mAh. 82% (174 mAh) of this can be used by the VMS.

Before performing the calculation, the source code of the application under development must be available. Main processing parts should be extracted as model programs.

Calculate total number of instruction cycles where quartz oscillator is operating

Enter this value at position A in the chart.

Calculate total number of instruction cycles where RC oscillator is operating

Enter this value at position B in the chart.

Reference	For information on instruction cycles used by the various instructions, refer to the “VMS Programmer's Guide”.

20.4.1 Calculating Continuous Operating Time

Calculate division ratio

Calculate RC oscillator operation time at 1/12 ratio setting (OCR7 = 0) and 1/6 setting (OCR7 = 1)

Enter these values at positions C and D in the chart.

Calculate total operation time of quartz oscillator

There are two C positions in the chart. Add up the two values, multiply by 30.5 µs, and enter the result at position E in the chart.

Calculate total operation time of RC oscillator

There are two D positions in the chart. Add up the two values, multiply by 1 µs, and enter the result at position F in the chart.

Calculate average current consumption of quartz oscillator

Multiply the value in E by 0.610 mA, and enter the result at position G in the chart.

Calculate average current consumption of RC oscillator

Multiply the value in F by 2.600 mA, and enter the result at position H in the chart.

Calculate combined average current consumption of quartz oscillator and RC oscillator

Calculate (G + H) ÷ (E + F), and enter the result at position I in the chart.

Calculate battery life (hours) from effective battery capacity (190 mA)

174 mAh ÷ I yields battery life J (hours).

Provide a 10% safety margin.

Possible factors influencing the actual battery life are sub-processing cycles, flash memory write access, load during data transfer, temperature influences, etc.

J × 0.9 = K (battery life in hours, with margin)

This value can be included in product documentation. It should be defined as the expected life of one set of batteries for continuous operation of the software.

20.4.2 Calculating Battery Life in Days

The following steps show how to calculate expected battery life in days, assuming a certain number of hours of use every day, and assuming that the unit is in sleep mode (0.060 mA) at other times.

Calculate current consumption during play hours

Calculate I × number of play hours per day L (hours), and enter the result at position M in the chart.

Calculate daily current consumption in sleep mode

Calculate 0.060 mA × (24 hours - L), and enter the result at position N in the chart.

Calculate daily average current consumption

Calculate (M + N) ÷ 24, and enter the result at position O in the chart.

Calculate effective battery life in days

Calculate (174 ÷ 24) ÷ O, and enter the result at position O in the chart.

Provide a 10% safety margin.

P × 0.9 = Q (battery life in days, with margin)

This value can be included in product documentation. It should be defined as the expected life of one set of batteries when using the software for n hours per day.

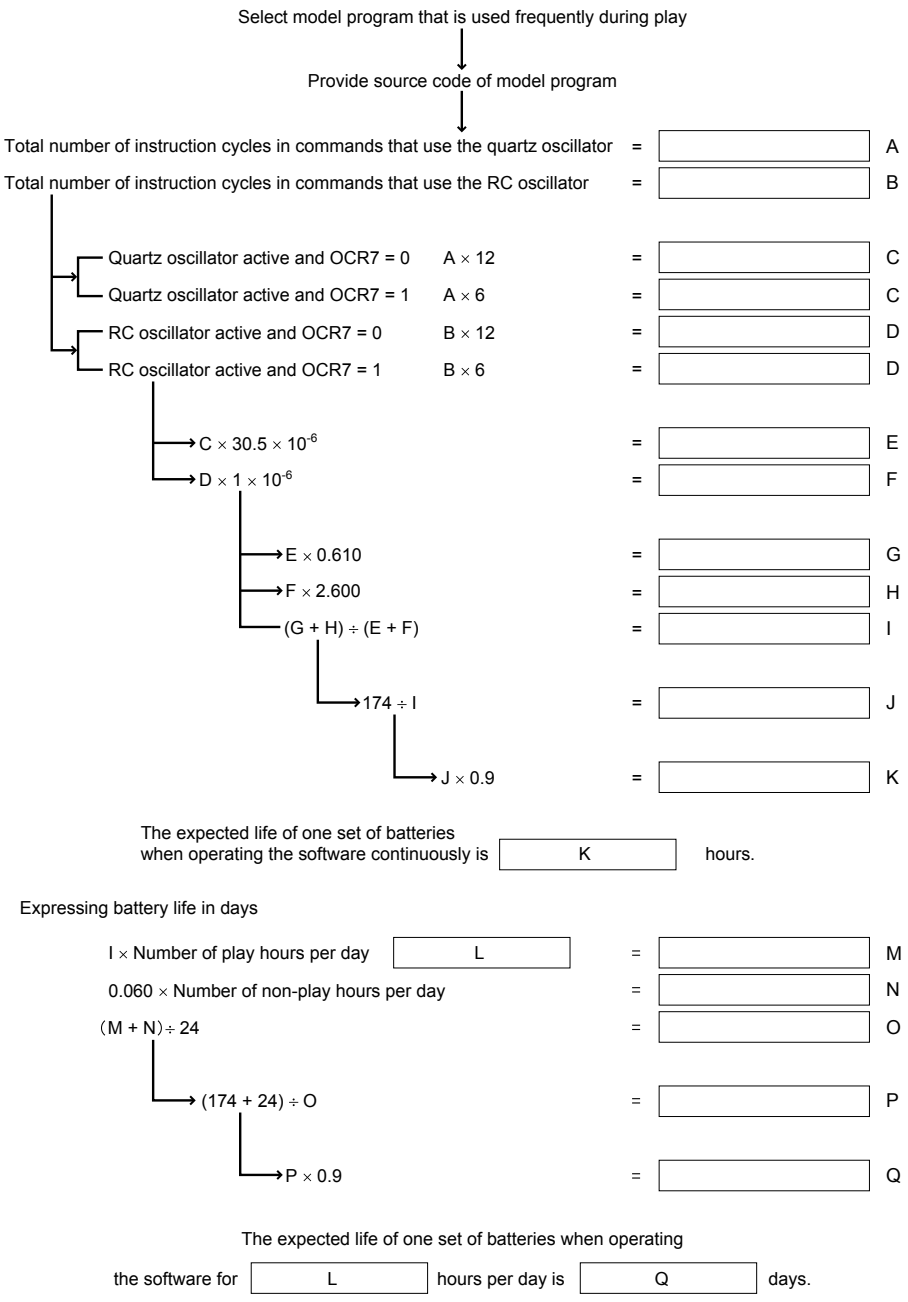

Select model program that is used frequently during play
Provide source code of model program
Total number of instruction cycles in commands that use the quartz oscillator = A
Total number of instruction cycles in commands that use the RC oscillator = B
Quartz oscillator active and OCR7 = 0 A × 12 = C
Quartz oscillator active and OCR7 = 1 A × 6 = C
RC oscillator active and OCR7 = 0 B × 12 = D
RC oscillator active and OCR7 = 1 B × 6 = D
C × 30.5 × 10-6 = E
D × 1 × 10-6 = F
E × 0.610 = G
F × 2.600 = H
(G + H) ÷ (E + F) = I
174 ÷ I = J
J × 0.9 = K
The expected life of one set of batteries
when operating the software continuously is K hours.
Expressing battery life in days
I × Number of play hours per day L = M
0.060 × Number of non-play hours per day = N
(M + N) ÷ 24 = O
(174 + 24) ÷ O = P
P × 0.9 = Q
The expected life of one set of batteries when operating
the software for L hours per day is Q days.

Fig. 20-1

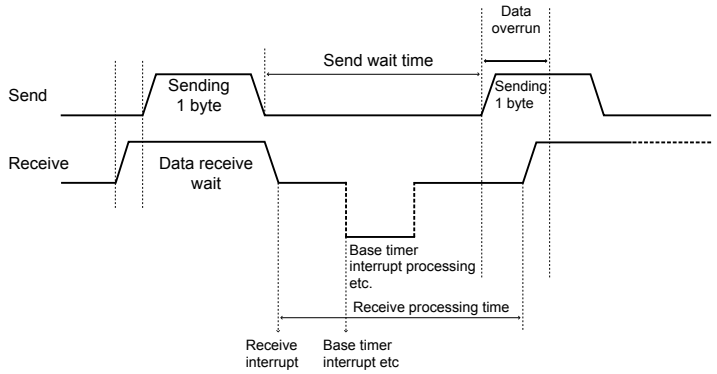
Chapter 21 Serial Communication Precautions

This chapter describes points to be observed regarding serial communication between two connected VMS units. Techniques for ensuring proper communication are also explained.

21.1 Serial Communication Timing Chart

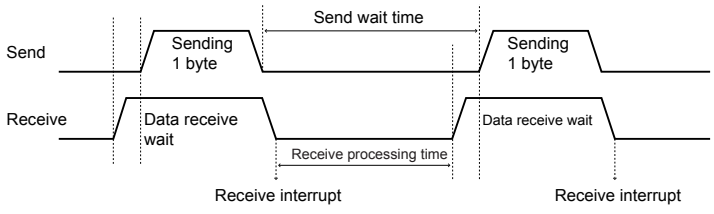
A timing chart for connecting two VMS units and performing serial communication is shown below.

Correct sequence diagram



When data are sent while the receive side is in the receive wait condition, the data will be received correctly. Otherwise, data overrun may occur, as shown in the next illustration.

Data overrun diagram



When the send wait time is longer than the receive processing time, data transfer will be carried out correctly. However, the base timer interrupt is generated also during receive processing, and the processing time will be longer than the actual time required for all receive processing handler steps.

21.2 Measures to Ensure Problem-Free Serial Transfer

The following measures for ensuring smooth serial transfer are possible.

(1) Mask all interrupts except those needed for serial transfer.

(2) Give highest priority to receive interrupt.

(3) Make send wait time longer than sum of receive wait time and time required for other processing steps.

Method (1) will result in a slow-down of the built-in clock of the VMS, because the clock uses the base timer interrupt, and interrupts cannot be counted while masking is active.

If method (2) were adopted, the receive interrupt handler located in the flash memory space would not be called, because the base timer interrupt handler (clock processing routine) is located in the ROM space.

When a base timer interrupt is generated, the CHANGE instruction causes the CPU to start program fetch from ROM. When a receive interrupt occurs while the base timer interrupt of the ROM BIOS is being processed, the CPU references the interrupt table (equivalent address) in the ROM space and jumps to that address. Because the CPU is carrying out ROM fetch, jumping to the appropriate receive interrupt handler is not possible. Therefore method (2) is not suitable and cannot be used.

For method (3), the processing time required for all interrupts with higher priority than the receive interrupt is calculated and added to the receive processing time, and the result is taken as the send wait time. This ensures correct data transfer but increases the time required for sending.

The decision of whether to adopt method (1) or (3) must be made by the application designer, while taking into account the advantages and disadvantages of either approach. These are described in more detail below.

21.2.1 Mask All Interrupts

When all interrupts except those needed for serial transfer are masked, the clock which uses the base timer interrupt will be slowed down. But compared to method (3), method (1) will result in higher transfer speed and simplified programming logic.

For transfer of only a few bytes, the clock slow-down will be within the tolerance specifications. But for frequent data transfer or transfer of a larger amount of data, method (3) should be adopted.

The send wait time is calculated as follows.

(1) Count the total number of steps required for receive processing.
In addition to the number of steps in the interrupt processing handler, also include the steps immediately before and immediately after the interrupt.

(2) Check the system clock and division ratio used during execution of these steps.

(3) Calculate the send wait time from the total number of steps, system clock, and division ratio.

(4) Incorporate the wait time calculated in (3) into the NOP or similar for send side processing.

21.2.2 Set Maximum Send Wait Time

Compared to masking all interrupts, this method results in a considerable decrease of transfer speed. When coding an application using this method, the send wait time calculation and priority assignment must be carried out with care. The advantage of this method is that there will be no clock slow-down.

For calculating the send wait time, it is assumed that all interrupts with higher priority than the receive interrupt have been generated. The processing time required for these is added to the receive processing time, and the result is taken as the send wait time.

Note	Because the port 3 interrupt is a level interrupt, it will be generated continuously for as long as the user presses a button. If this happens during a transfer, data overrun may occur regardless of how long the send wait time is set. For checking the button press status during a serial transfer, inhibit the port 3 interrupt and use timer 0 or other means for monitoring port 3 latch data.

(1) Check whether there is an interrupt with equal priority to the receive interrupt. If there is such an interrupt, set the interrupt to a higher or lower priority than the receive interrupt.

(2) Count the total number of steps required for receive processing.
In addition to the number of steps in the interrupt processing handler, also include the steps immediately before and immediately after the interrupt.

(3) Check the system clock and division ratio used during execution of these steps.

(4) Calculate the processing time from the total number of steps, the system clock and division ratio.

(5) Pick up interrupts with higher priority than the receive interrupt, except the base timer interrupt.

(6) Count the total number of steps for the interrupt handlers of these interrupts.

(7) Calculate the processing time for each handler, according to the method of steps (2) to (4).

(8) Calculate the base timer interrupt processing time.
Because the base timer interrupt handler is processed within the BIOS, the number of steps can be assumed as shown below. These figures apply only if GHEAD.ASM was not changed. If GHEAD.ASM was changed, count the actual number of steps.

Shortest case

Second count processing only

	5 steps (until CHANGE instruction in GHEAD.ASM)
+	48 steps (in BIOS)
+	3 steps (until return to user program in GHEAD.ASM)
Total	56 steps

Longest case

Year increment processing necessary

	5 steps (until CHANGE instruction in GHEAD.ASM)
+	145 steps (in BIOS)
+	3 steps (until return to user program in GHEAD.ASM)
Total	153 steps

Caution	Because the longest case shown above will occur only once per year, it is not necessary to always provide for this number of steps. The system clock and division ratio are derived from the settings established by the application.

(9) Calculate the base timer interrupt processing time from the system clock and division ratio.

(10) Add up the receive interrupt handler processing time and the time required for all calculated interrupts.

(11) Incorporate the wait time calculated in (10) into the NOP or similar for send side processing.

index

Symbol

@R0 48
@R1 48
@R2 48
@R3 48
14-bit binary up counter 101
16-bit reload counter 67, 81
16-bit reload timer....67, 80, 83, 94
2-channel synchronous serial interface...108
8-bit pulse generator 83, 91
8-bit reload timer
+ 8-bit pulse generator 83, 91
8-bit reload timer
+ 8-bit reload counter 67, 78
8-bit reload timer x 2 channels....77
8-bit reload timer × 2 channels 67, 83, 89

A

AC 54
ACC....52
Accumulator....45, 52
Alarm Function....34
Application development....30
Application Shutdown Procedure
When MODE Button is Pressed....183
Arithmetic Register Contents 53
ASEL 146
Auto power-off....33
Auto power-off function 24
auxiliary carry flag....54

B

B register....53
Bank address register 134
Base Timer....101
base timer clock select....72
Base timer control register 103
base timer interrupt 0 cycle control....103
base timer interrupt 0 request
enable control 105
base timer interrupt 0 source flag 104
base timer interrupt 1 cycle control....104
base timer interrupt 1 request
enable control....104
Base timer interrupt1 source flag....104
base timer operation control 104
Batteries....35
Battery exhaustion 35
Battery life....35
Battery life calculation 225
Battery life calculation
for n hours of play per day....229
Battery life, methods for enhancing....225
Battery replacement....35
Battery status monitoring 35
Baud rate generator register....116
BCD time and date data....197
Block number....28
BTCR 103
BTCR0 105
BTCR1 104
BTCR2 104
BTCR3 104
BTCR5 - BTCR4....104
BTCR6 104
BTCR7 103
Bus cycle time 40
buSaveExecFile 149
Buttons....34

C

C register 53
Calculating Continuous
Operating Time....228
carry flag 54
CHANGE instruction 45
CHANGE Instruction 57
Character count register 133
character font....25
Clock count-up timer 197
Clock mode....26
Clock switching timing....189
Clock timer....101
CNR 133
Counter 0 67
CPU Features....39

current page 47
CY 54
cycle clock 68

D

Data area 28
day 201
day_h 201
Dedicated Dreamcast interface circuit 116
Defined Variables List 201
Detectable signals 136
Differences to Conventional CPU 39
Direct addressing mode 49
Direction buttons, simultaneous key press prevention 64
display mode select 131
Display RAM 129
display RAM bank address control 134
Display start address control register 132
Dot Matrix LCD Controller 129
Dreamcast application linking 32
Dreamcast connection check port 66
Dreamcast Connection Status and Operation Mode 25
ELDT1C 86
Executable file size 30
Executable file start 30
Executable file transfer 30
expansion device 22
Expansion device connector 22
External clock 120
External interrupt 2, 3 control register 74
External Interrupt Function 136

F

Fast-forward mode 101
File Management 27
File mode 26
Flash memory 40
Flash Memory 52, 149
Flash memory allocation 30
Flash memory management area 28
Flash Memory Memory Map 27
Flash memory page 192
Flash memory page data read 191
Flash memory space 45
Flash memory specifications 43
Flash memory total capacity 27
Flash memory verify 195
Flash memory write 193
fm_prd_ex 191
fm_vrf_ex 195
fm_wrt_ex 193
fmadd_h 202
fmadd_l 202
Fmbank 202
Full-duplex serial transfer 33

G

Game mode 26
game_end 184
general-purpose RAM 45
GHEAD.ASM 174
Graphics display 129

H

HALT mode 43
HALT Mode 165
HALT mode cancel 166
HALT mode control 162
HALT mode setting 165
Header 22, 174
High-speed arithmetic instructions 42
hour 201
hour_h 201

I

I01CR0 140
I01CR1 139
I01CR3, I01CR2 139
I01CR4 139
I01CR5 139
I01CR7, I01CR6 138
I23CR 74
I23CR0 76
I23CR1 76
I23CR2 76
I23CR3 76
I23CR4 76
I23CR5 75
I23CR6 75
I23CR7 75
Icons 23
IE 141
IE1, IE0 142

IE7 142
INCE 146
Indirect Address Register 48
indirect address register bank flag 48
Indirect addressing mode 49
Individual interrupt vector 174
Input signal select register 71
I/O Ports 59
instruction cycle time 40
INT0 detection level/edge select 139
INT0 interrupt enable control 140
INT0 interrupt source 139
INT0\, INT1 interrupt priority control 142
INT1 detection level/edge select 138
INT1 interrupt enable control 139
INT1 interrupt source 139
INT2 falling edge detection control 76
INT2 interrupt request enable 76
INT2 interrupt source 76
INT2 rising edge detection control 76
INT2/T0L interrupt priority level setting ... 156
INT3 falling edge detection control 75
INT3 interrupt request enable 76
INT3 interrupt source 75
INT3 rising edge detection control 75
INT3/base timer interrupt priority level setting 156
Internal clock 119
internal memory space 45
Interrupt 42
Interrupt Functions 151
Interrupt Function Operation 153
Interrupt high priority level 153, 157
Interrupt highest priority level 153, 156
Interrupt low priority level 153, 157
Interrupt nesting 153
Interrupt Priority Control Register 155
Interrupt priority sequence 156
Interrupt Types 152
Interrupt vectors 47
IP 155
IP0 156
IP1 156
IP2 156
IP3 156
IP4 155
IP5 155
IP7 155
IRBKx 54
ISL 71
ISL0 72
ISL2, ISL1 72
ISL5, ISL4 72

L

LCD 28
LCD clock 1/2 division select 131
LCD clock division ratio 131
LCD contrast control register 134
LCD Controller 129
LCD controller control 131
LCD display control 135
LCD display RAM access control 136
LCD refresh rate 29
LDC instruction 52
leaf_f 201
Low-battery detection enable flag 199
Low-battery detection flag 199
Low-battery detection function 199
LSB/MSB start select 112, 115
LSB/MSB switchable start sequence 108
LSB/MSB Switchable Start Sequence 122

M

Maple bus 22
Maple bus mode 117
Maple bus protocol 33
master interrupt enable control 142
Master interrupt enable control register .. 144
Maximum transfer rate 33
MCR 130
MCR0 131
MCR3 131
MCR4 131
MCR7, MCR5 131
Memory Map 22
memory-mapped I/O 45
min 201
min_h 201
MODE button 24
Mode control register 130
Mode Setting 25
mon 201
mon_h 201
Noise filter time constant select 72
Normal mode 117

Number of simultaneous sounds 34
OCR 160
OCR1 161
OCR5, OCR4 161
OCR7 161
Operation mode switching 24
OS program 22, 31
OS Program 174
Oscillation frequencies 34
oscillator circuit 43
Oscillator circuit and
current consumption 226
Oscillator circuit frequency 158
Oscillator control register 160
Oscillator operation stop and start 227
Output Waveform and
Parameter Setting 208
OV 54
overflow flag 54
Overrun detection 108
Overrun Detection 124
overrun flag 112, 114

P

P 55
P1 60
P1 port use select control 146
Select P10 function 62
P10FCR 62
Select P11 function 62
P11FCR 62
Select P12 function 62
P12FCR 62
Select P13 function 62
P13FCR 62
Select P14 function 62
P14FCR 62
Select P15 function 62
P15FCR 62
Select P16 function 62
P16FCR 62
P17, P10 I/O control 61
Select P17 function 62
P17FCR 62
P1DDR 60
P1FCR 61
P1nDDR 61
P3 64
P30INT 65
P31INT 65
P32INT 64
P3DDR 64
P3INT 64
P7 66
parity flag 55
PC 46
PCON 162
PCON0 162
Piezoelectric buzzer 24
Port 41
Port 1 60
Port 1 Block Diagram 63
Port 1 data direction register 60
Port 1 function control register 61
Port 1 latch 60
Port 3 64
Port 3 data direction register 64
Port 3 interrupt control register 64
port 3 interrupt generation selector flag 64
port 3 interrupt request enable control 65
port 3 interrupt source flag 65
Port 3 latch 64
Port 7 66
Port 7 Block Diagram 66
port data 59
Port Interrupt Function 142
port latched data 59
Power control register 162
Prescaler 68
Program Counter 46
Program status word 45
Program Status Word 54
PSW 45, 54
Pulse generator 24
Pulse generator data update
enable control 86
PWM 24
PWM sound source 203
PWM sound source frequency
characteristics 210
PWM sound source frequency table 210

Q

Quartz oscillator 163
Quartz oscillator circuit
oscillation frequency 163

R

RAM 31, 40
RAM bank flag 55
RAM bank0 45
RAM Memory Map.....31
RAM Space.....48
RAM used by system BIOS 175
RAMBK0 55
RC oscillator.....162
RC oscillator circuit control.....161
RC oscillator frequency 162
registers 45
Reset hardware status 167
RETI instruction 153
Return to system BIOS.....184
ROM.....40
ROM space.....47
ROM table reference instruction.....52

S

SBR.....116
SBUF0.....116
SBUF1.....116
SCK0 polarity control.....112
SCON0..... 111
SCON00.....113
SCON01.....113
SCON02.....112
SCON03.....112
SCON04.....112
SCON06.....112
SCON07.....112
SCON1.....114
SCON10.....115
SCON11.....115
SCON12.....115
SCON13.....115
SCON14.....114
SCON16.....114
sec.....201
sec_f.....201
sec_h.....201
Serial buffer 0.....116
Serial buffer 1.....116
Serial interface.....41
Serial Interface.....107
Serial Interface Operation 117
Serial interface.....33, 109
Serial transfer 0 clock control 62
Serial transfer 0 data control.....62
Serial transfer 1 clock control 62
Serial transfer 1 data control.....62
Serial transfer clock..... 119
Serial transfer timing 117
Serial Transfer Timing 122
SFR.....39, 49
SIO0 control register 111
SIO0 interrupt priority level setting 155
SIO0 interrupt request enable control 113
SIO0 operation control 112
SIO0 transfer clock polarity
switching function..... 108
SIO0 transfer end flag 113
SIO1 control register 114
SIO1 interrupt priority level setting 155
SIO1 interrupt request enable control 115
SIO1 operation control 115
SIO1 transfer end flag 115
SIOSEL..... 146
Sleep Function 34
Sleep state.....24, 33
Sleep state cancel.....34
Software clock 121
Sound Output Method 205
Sound start 209
Sound stop..... 209
SP 55
Special function register.....39, 49
Special Function Register
Initial Values 168
Specifications 20
Stack area.....42
Stack Pointer 55
STAD..... 132
STAD7\, STAD0..... 133
Standalone operation
(without using SIO).....60
Standby function.....43
Standby Operation Status..... 165
Subroutine Call Procedure..... 179
Subroutine precautions 189
Subroutine Reference 189
Symbol Table..... 221
Synchronous serial transfer 33
System BIOS 22, 173

System BIOS Functions 177
System clock generator circuit control..... 161
System clock generator function 158
System clock select..... 161
System mode..... 26
System program 22

T

T0 67
T0 clock input pin select..... 72
T0CNT 69
T0H 74
T0H count control 69
T0H interrupt..... 67
T0H interrupt priority level setting..... 156
T0H interrupt request enable control..... 70
T0H overflow flag 70
T0HIE..... 70
T0HOVF..... 70
T0HR..... 74
T0HRUN 69
T0L 73
T0L count control..... 69
T0L input clock select..... 70
T0L interrupt 67
T0L interrupt request enable control 71
T0L overflow flag 70
T0LEXT..... 70
T0LIE..... 71
T0LONG..... 70
T0LOVF 70
T0LR 73
T0LRUN..... 69
T0PRR 73
T1 83
T1 interrupt priority level setting 156
T1CNT 84
T1H 88
T1H count control 85
T1H interrupt request enable control..... 86
T1H overflow flag 86
T1HC..... 89
T1HIE..... 86
T1HOVF..... 86
T1HR..... 88
T1HRUN 85
T1L 87
T1L count control..... 85
T1L interrupt request enable control 87
T1L overflow flag..... 86
T1LC 88
T1LIE 87
T1LONG 86
T1LOVF 86
T1LR 87
T1LRUN..... 85
Table Reference Register..... 56
Table reference register (lower byte)..... 56
Table reference register (upper byte) 56
TDR 133
Time data..... 197
Time division register..... 133
Time measurement..... 33
Timer..... 41
Timer 0 67
Timer 0 high register..... 74
Timer 0 high reload register..... 74
Timer 0 low register 73
Timer 0 low reload register..... 73
Timer 0 prescaler 67
Timer 0 prescaler data register 73
Timer 1 83
Timer 1 bit length select..... 86
Timer 1 control register..... 85
Timer 1 high comparison data register 89
Timer 1 high register..... 88
Timer 1 high reload register..... 88
Timer 1 low comparison data register..... 88
Timer 1 low register 87
Timer 1 low reload register..... 87
Timer/counter 0 bit length select 70
Timer/counter 0 control register 69
timer_ex..... 197
transfer bit length control 112, 114
Transfer bit length control function..108, 125
Transfer clock select function..... 108
TRH..... 56
TRL 56
TRR..... 56
Port 3 interrupt priority level setting..... 155

V

Variable bit length pulse generator 83, 95
Variable Bit Length Pulse Generator..... 217
VCCR..... 134
VCCR5, VCCR0..... 136

VCCR6 136
VCCR7 135
VMS application transfer 149
VMS control register 146
VMS Initialization 187
VMS low-voltage check port 66
VMS mode select 223
VMS operation modes 23
VMS simulator 30
VMS System Block Diagram 44
VMS Work RAM 146
VMS-to-VMS connection detect 137
VRMAD1, VRMAD2 147
VSEL 146
VTRBF 48
VTRBF address counter auto increment . 146
VTRBF address input select control 146

W

work RAM 40, 48
Work RAM 31, 146, 147
Work RAM access 147
Work RAM access address 147
Work RAM address register precautions 148
XBNK 134
XBNK1, XBNK0 134
XRAM 28, 129
XRAM display start address setting 133
year 201
year_h 201